Multi-modal and Inertial Sensor Solutions for Navigation-type Factor Graphs

by

Dehann Fourie

Submitted to the Dept. of Electrical Engineering and Computer Science and the Joint Program in Applied Ocean Science and Engineering

in partial fulfillment of the requirements for the degree of Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

September 2017

Author .....

Joint Program in Applied Ocean Science and Engineering

August 31, 2017

Certified by .....

John Leonard

Samuel C. Collins Professor of Mechanical and Ocean Engineering

Thesis Supervisor

Accepted by .....

Henrik Schmidt

Professor of Mechanical and Ocean Engineering

Chair, Joint Committee for Applied Ocean Science and Engineering, WHOI

Accepted by .....

Leslie A. Kolodziejski

Professor of Electrical Engineering and Computer Science

Chair, Committee on Graduate Students, MIT

Multi-modal and Inertial Sensor Solutions for Navigation-type Factor Graphs

by

Dehann Fourie

Submitted to the Dept. of Electrical Engineering and Computer Science and the
Joint Program in Applied Ocean Science and Engineering
on August 31, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis presents a sum-product inference algorithm for platform navigation called Multi-modal iSAM (incremental smoothing and mapping). Common Gaussian-only likelihoods are restrictive and require a complex front-end processes to deal with non-Gaussian measurements. Instead, our approach allows the front-end to defer ambiguities with non-Gaussian measurement models. We retain the acyclic Bayes tree (and incremental update strategy) from the predecessor iSAM2 max-product algorithm [Kaess et al., IJRR 2012]. The approach propagates continuous beliefs on the Bayes (Junction) tree, which is an efficient symbolic refactorization of the nonparametric factor graph, and asymptotically approximates the underlying Chapman-Kolmogorov equations. Our method tracks dominant modes in the marginal posteriors of all variables with minimal approximation error, while suppressing almost all low likelihood modes (in a non-permanent manner). Keeping with existing inertial navigation, we present a novel, continuous-time, retroactively calibrating inertial odometry residual function, using preintegration to seamlessly incorporate pure inertial sensor measurements into a factor graph. We centralize around a factor graph (with starved graph databases) to separate elements of the navigation into an ecosystem of processes. Practical examples are included, such as how to infer multi-modal marginal posterior belief estimates for ambiguous loop closures; raw beam-formed acoustic measurements; or conventional parametric likelihoods, and others.

Thesis Supervisor: John Leonard
Title: Samuel C. Collins Professor of Mechanical and Ocean Engineering

Acknowledgments

I would like to thank my mother, Linda, and my late father, Chris, for their love, sacrifice, care, support, trust, motivation, and commitment throughout the years. My fiance, Kristen, for her love and support, and the countless hours spent on thesis work. My family and friends during this journey, and a special thanks to my friend Sam Claassens who has always been a great help and sounding board throughout the thesis period. Thanks to David Rosen for discussions, reading, and deeper investigation into many aspects of this thesis. Thanks Mark, Johan, Pedro, Sudeep, Nick, Thomas, and Ted. I would also like to thank all the staff at MIT and WHOI for their help in completing the many required tasks along the way, and a special thank you to Judy. I would like to thank my advisors, Professor John Leonard for the opportunity, and Professor Michael Kaess for all discussions and brain storming. My committee, Dr. John Fisher III, Prof. Russ Tedrake, and Dr. James Kinsey for your guidance. Thanks to late Professors Seth Teller and Andre Booysen for their contributions and support.

This work was partially supported by the National Science Foundation under grant IIS-1318392 and by the Office of Naval Research under grant N00014-16-1-2628.

Contents

List of Figures

List of Tables

List of Acronyms

PG pose graph

RPG reduced pose graph

FG factor graph

GPS global positioning system

DVL Doppler velocity log

DIDSON dual frequency identification sonar

GPU graphical processing unit

LBL long-baseline acoustic navigation

RLG ring laser gyro

FOG fiber optic gyro

AUV autonomous underwater vehicle

HAUV hovering autonomous underwater vehicle

ESEIF exactly sparse extended information filter

SLAM simultaneous localization and mapping

EKF extended Kalman filter

EIF extended information filter

SEIF sparse extended information filter

ESDF exactly sparse delayed-state filter

SAM smoothing and mapping

iSAM incremental smoothing and mapping

IMU inertial measurement unit

LORAN-C long range navigation

USBL ultra short base line

UKF unscented Kalman filter

NDT normal distribution transform

ROV remotely operated vehicle

ML Maximum Likelihood

MRF Markov random field

DBMS database management system

SQL standard query language

VO visual odometry

LM LevenbergâĂŞMarquardt

SAD sum of absolute differences

OOI ocean observatories initiative

LCM lightweight communication and marshaling

WO wheel odometry

Chapter 1

Introduction

This thesis is focused on developing simultaneous localization & mapping, together with high-bandwidth state estimation, following a tendency of convergence among techniques towards a common state estimation, localization, mapping, and trajectory planning navigation system.

A key problem faced in navigation systems today is how to perform robust and efficient inference over many sensor measurements simultaneously, and how to guide a system towards some user desired goal. Sooner or later a bad sensor measurement or algorithmic decision will disrupt the consistency of a navigation and control system's estimates, resulting in unpredictable behavior and effectively voids reliable use of such a system all together. A robotic navigation system should create an artificial *location and dynamic awareness* of a vehicle over widely varying timescales, starting in the sub-millisecond through to sessions lasting multiple hours.

Gyroscope and accelerometer measurements for inertial navigation are well established for high-bandwidth navigation, but require additional aiding measurements to overcome systemic inertial sensor errors, such as measurement bias, see Farrell [56] Titterton et al. [229] or Groves [79]. In turn, all navigation sensory information should be equally assembled into a consensus state estimate. Information from multiple robots or sessions should also be aggregated into a larger common world state estimate.

Existing data fusion approaches generally assume that all sensing modalities have a Gaussian error distribution, however, in reality many measurements and associations are ambiguous and require much more capable measurement likelihood models. The community have generally suggested techniques to identify and remove so-called ”outlier” measurements, but we would rather allow the naviga-

tion system designer freedom to propagate uncertainty—by means of more complicated non-Gaussian measurement models—to the common inference system and let consensus be found there in aggregate.

1.1 Motivation

Fig. 1-1 shows four different robotic platforms which had a direct influence on the development of this thesis. Each enables development into more advanced autonomy and robotics methods for helping people solve greater societal problems. Robotic platforms in Fig. 1-1 were developed for data gathering in environments which are too hostile or prohibitively expensive for humans to go in person, and intended to advance the science, national security, or the monitoring of valuable assets.

The Atlas robot can manipulate objects and was part of the DARPA robotics challenge—which was officially motivated by the Fukushima nuclear disaster and Deep Water Horizon explosion and oil spill events. The humanoid robot, in some way, represents the perfect motivation for pursuing robotic systems: offloading dangerous and difficult tasks, rather than risking human life, to replaceable hardware which can interact with a world built for humans – climbing stairs, ladders, opening doors and valves, pulling levers, using common power tools, etc.

We note among the platforms shown in Fig. 1-1, WHOI's Sentry vehicle is the most active. Simultaneously, Sentry's underwater operations are some of the most challenging, given the hostile environment at $6000\ m$ depth in a hot to cold, corrosive and conductive environment; with limited access prior information since most operations are in new unexplored areas; notable funding limitations being considerably less than the comparable space industry, although the vehicle costs more than twice that of the original BDi Atlas humanoids, and about five times one of the SPHERES satellites; compounded by the major risk of vehicle loss; and severe physical limitations such as having only acoustic signal propagation available for communication and finding an on-board and mothership based navigation solution. The point being, that the more difficult the operating environment becomes, the more feasible a robotic solution is.

Our motivation is well aligned with the views of Mindell [151] who illustrates the *perfect-five* concept: Consider some task, such as robotic construction in space, drilling for oil in the deep ocean, or displacing dull and dirty tasks such as cleaning solar panels in the hot sun. Each of these tasks have levels of abstraction, starting with low-level motor and actuator control up to general system guidance and deci-

Figure 1-1: Robotic systems which gave direct motivation for this thesis. Top left shows a long endurance solar power unmanned aerial vehicle using a GPS and magnetometer aided inertial navigation system in 2010 [61, 241]. Top right shows the SPHERES/VERTIGO experiment aboard the International Space Station during early testing in 2013 of the visual relative object tracking and inertial navigation experiments along with directional, synchronized time-of-flight acoustic beacons for absolute ranging and reference [66, 67]. Bottom left shows the Team MIT's BDi Atlas humanoid robot (later 2013) for the DARPA robotics challenge, using a kinematic and lidar aided inertial navigation system, with the intention of remote user manipulation of objects [54, 223]. Bottom right shows the Woods Hole Oceanographic Sentry autonomous under water vehicle before deployment at Juan de Fuca ridge in 2014, using multiple inertial sensors and Doppler velocity log for navigation; as well as independent acoustic and optical localization of equipment deployed on the sea floor [85, 189]. Photo credit, top-right: NASA.

sion making. We describe level one autonomy as the equivalent of joystick control of the lowest level actuators, while level ten autonomy can be a hypothetical case where we push a button and walk away while a robot does some value-adding task.

Practical systems should enable human-in-the-loop decision making and intervention while computational and manipulative autonomy deals with low-level, mundane tasks that logically and defensibly follow from the user's stipulated intentions, see Mindell's discussion on the Apollo guidance system [150]. For example, a self driving car might take control in consistent traffic flows, or park itself at slow speed in dedicated parking lots. Another example would be a hovering robot in space, air or under water for inspection operations and might be moved in increments with automated positional control from sparsely human defined way points.

1.2 Applications of Interest

We list a few examples applications for which we would like a navigation solution. While all the applications will require different front-end processes to interact the different sensing modalities used, we would like to generalize as much of the solution as possible. Our focus in this thesis is on the largest common denominator, the back-end inference solution, and again note that these decisions are guided by experience from the platforms shown in Fig. 1-1.

1.2.1 Kinematic Robots

Humanoids or quadruped kinematic robots are well motivated as generic robotic platforms interacting with objects in a man-made environment. Such kinematic robots require both a high-bandwidth vehicle state estimate and a long-term stable and accurate navigation solution. Robot trajectory planning an manipulation processes also require a map estimate of the local surroundings. To manipulate objects, the automation also requires some virtual representation of objects that a machine can understand how to interact with, which we shall call an affordance.

Consider a robot moving towards a local object in preparation for a manipulation task, as illustrated in Fig. 1-2, where a the robot has been instructed to walk up to, reach out, open, and then walk through the doorway. The navigation and control needs to plan forward in time and then, through control, guide the robot along the planned trajectory towards the user specified goal. The relative state estimate between the robot and the door should improve as the robot approaches the

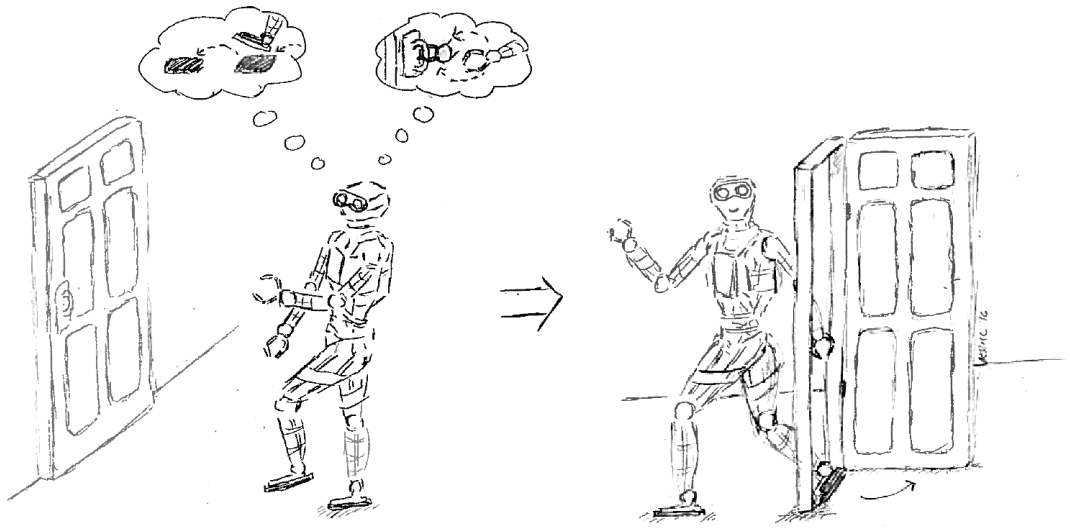


Figure 1-2: Vision of what a centralized navigation system should be able to achieve in terms of a kinematic robot manipulating objects in its surroundings. This humanoid robot example is used throughout the thesis as a hypothetical use-case, along with a secondary example in Fig. 1-3.

intended manipulation target. Increased accuracy indicates some kind of feedback between the localization and mapping estimation processes.

Focusing on the individual state estimation elements: Consider estimating the position of the robot's hands or feet relative to the robot body, and then also relative to the objects in the world surrounding the robot. The guidance and control systems need to prevent object and self collisions while the robot kinematics are being actuated and trajectories planned. From these and similar arguments, we can see a strong common framework is required for expressing spatial relationships between all robot kinematic components, physical objects in the world, and context of the surrounding map.

A kinematic robot nominally has measurements from gyroscopes and accelerometer triads, joint encoder kinematics, motion model predictions, cameras, GPS satellite ranging, and possibly laser scanners and other body mounted ranging equipment which should be combined into a consensus navigation state estimate. The question becomes how do we best combine these different sensing modalities while achieving robustness in a high-bandwidth, real-time state estimate.

1.2.2 Wheeled Robots

Wheeled vehicles, such as industrial factory and packing robots, self driving cars, and other domestic robots are popular platforms which require a robust navigation solution. Wheeled robots commonly employ wheel odometry and steering angle measurements, inertial measurement units, and can easily be fitted with multiple cameras and Doppler radar sensors. More complicated sensing modalities such as two dimensional and three dimensional laser scanners can also be added. Road incline, engine throttle position, and steering angle can be monitored for predictive modeling of future actions under multiple control policies. This thesis is intended to contribute in how to combine each of the navigation measurement and trajectory planning elements.

1.2.3 Flying Robots

Unmanned aerial vehicles (UAVs) are a popular reference for autonomous systems, also known as *drones*. Automated flying vehicles date back to the first World War, using mechanical gyroscopes, pendulums, and propeller turn counts, vehicle trajectories could be planned and executed in crude fashion. State-of-the-art Kalman filtering based inertial and GPS navigation systems are the foundation for intelligent fly-by-wire control systems.

Many applications and operational requirements for aerial vehicles often dictate that aerial navigation systems depend on on-board measurements only. Measurements including GPS ranging, inertial measurements, passive camera, active laser scanners, ground Doppler, range, and imaging radar are common sensing modalities. Active ranging signals likely suffer adverse multi-path affects.

1.2.4 Marine, Ocean, and Space Robotics

Marine vehicles would greatly benefit from automation across vast oceans and given the given the relative safety of automated operations during the early adoption and development phases. A system failure during testing would in most situations result in the vehicle slowly drifting allowing hours or days to intervene, opposed to aircraft or cars where severe damage swiftly follows from a system failure. Lastly, consider the relative payloads that can be carried by marine vehicles – even a modest size vehicle can carry an enormous amount of stored energy and computational resources.

Automation in marine and ocean engineering can allow missions where vehicles go to sea for months at a time, and current news media has plentiful examples

of increasing reliance on unmanned marine technology by both military and commercial entities. The Woods Hole Oceanographic Institutions' (WHOI) Jetyak [122] vehicle is a low-cost vehicle which in the scientific community has already been replacing manned operations for dull and dangerous data collection activities. Another example is a long history of bathymetric data collection, with recent practical demonstrations of behavioral control strategies of the Sentry AUV searching for underwater oil plumes in during the Deep Water Horizon oil spill [28], or transceiver localization and pursuit for optical data muling operations [85, 189].

Further opportunities for marine autonomy include security patrols for piracy or narcotics; data collection and surveying for science and construction; autonomous convoy shipping, rigging, and tugging operations; and rapid emergency response to a vehicle in distress. Furthermore, underwater robotics equipment is arguably some of the most advanced – such as deep sea oil and gas infrastructure, exclusively assembly by remotely operated vehicles (ROVs).

Fig. 1-3 shows a hypothetical robot in a space or underwater setting, approaching a known object over topographically known terrain, and illustrates how a robot should be able to arrive at some user-defined position relative to existing equipment on the surface. What is not emphasized in the figure is the level of unpredictability in the environment or user intentions.

Looking at the sensing modalities common to marine systems, GPS measurements are used for location at the surface as available, and sightings to remote landmarks in coastal regions. Magnetic and gyrocompass systems are common, along with a velocity log for dead reckoning navigation and bottom depth measurements. Throttle and rudder positions may also be recorded for model based dead reckoning. Boats and ships often have radar and possibly sonar imaging capabilities. When operating near structures or in rivers, we can expect above water cameras and laser scanners.

Underwater vehicles generally only have access to inertial measurement and magnetic measurements from passive sensors, and a motion model prediction. Active acoustic ranging, such as fixed long and ultra-short base line (LBL & USBL), and Doppler velocity logs (DVL) for sea floor or under ice operations are common. Imaging sonars or powerful laser scanners are also available but generally at high cost. A camera might be present for near bottom or near structure operations but cloudy water or any form of remote operation prevents use of visual data as primary source of navigation. We note the enormous value of camera data as a secondary source of navigational information. Lastly, consider moving over topographically interesting terrain from where we can back out likely locations using a prior map of the area, altitude above ground and depth below the surface.

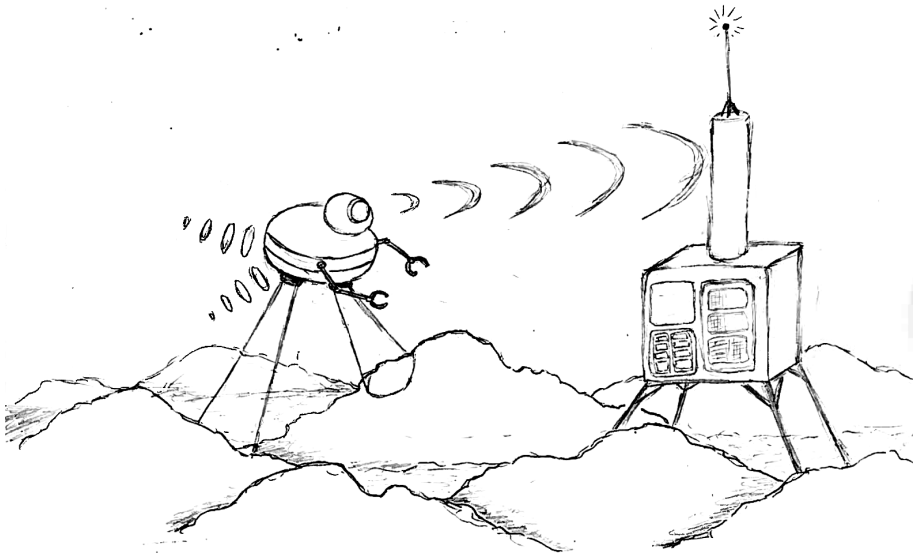


Figure 1-3: Vision of what a centralized navigation system should be able to achieve in terms of marine or space robotics, here showing a hypothetical inspection and light intervention case.

Many of these sensing modalities have highly non-Gaussian error distributions, and are similar to what space robotic systems will use and considered part of the discussion here.

1.2.5 Free-moving Navigational Devices

Many applications require estimating the position of some device or person in a working volume. Consider estimating the position of a robotic end-effector relative to its mounting or work piece being manipulated. An nearly independent navigational device that could be attached to a robot hand may be more cost effective than requiring high quality joint encoders and mechanics. Going back to the kinematic robots discussed in Section 1.2.1, we could imagine a more extreme situation where the hands and feet of a robot are jointly navigated with low-cost inertial and ranging measurements, avoiding more complicated kinematic structures and joint angle measurements.

A slightly different example, such as in car manufacturing, an automated assembly line assumes that each car will move through the line at exactly the same location. Now consider mounting a temporary navigation device to each of the objects being manipulated, as well as the robotic arms working in the volumes.

If such a navigation system were able to track each device accurately, then the tolerances along the assembly line can be relaxed and the robots be given more autonomy to react to passing of each individual item.

Consider hand-held mobile devices to be used as personal navigation systems, such as the Google Tango device [2], which will allow software applications to interact with the physical world. People could use a the technology to better navigate in a foreign train station, or be navigated away out of buildings during a fire emergency, or be tracked in a dangerous environment such as a battlefield. Free-moving devices must conform to strict size, power, weight, and computational restrictions. Again, low-cost inertial sensors, magnetometers, and cameras can be included in such devices. Acoustic or radio ranging, or even directional, beacons could be incorporated into a working volume. Alternatively, the device could be tracked by statically mounted sensors.

The navigation modalities discussed in this section represent measurements that may be highly unpredictable. Lower quality joint encoder measurements would likely have backlash and poor transfer of angles due to hysteresis or otherwise. Camera scenes might be very dynamic, or views often obstructed by the robot, people, or moving objects in the scene; or could subject to adverse glare and reflection problems. Acoustic and radio ranging measurements will likely suffer from plentiful multi-path problems in confined spaces, or around many flat reflective surfaces. Magnetic sensors would likely be corrupted with metal structures or electrical currents.

1.3 Problem Statement

General autonomous mobile (robotic) platforms require a computational inference and sensor modeling strategy for robust in-situ navigation solutions covering timescales of milliseconds through multiple hours.

1.3.1 Premise

This research is directly focused on platform state estimation, taking lessons from past navigation systems and projecting towards improved robustness and enhanced capabilities. The state estimate and multi-sensor robustness concerns are treated separately and recombined into a common navigation system. Fig. 1-4 shows a conceptual illustration where a real-time state estimation is obtained from a short dead reckoning segment which is ”grounded” relative to some robust multi-sensor fusion algorithm which infers consensus navigation state from

sensory data over some computational period. The duration of computation and dead reckoning navigation are dictated by each application and performance requirements.

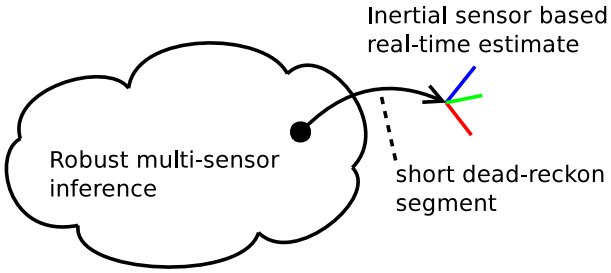


Figure 1-4: Conceptual illustration of some robust multi-sensor data fusion operation, discussed in Chapter 5, which supports a short dead reckoning trajectory segment from some odometry, where we will put more emphasis on using and dynamically calibrating inertial measurements, discussed in Chapter 4. The combination of state estimation and multi-sensor systems is discussed in Chapter 3.

Chapter 2 discusses existing solutions and their limitations. Previous experience suggests that, when operating in unknown environments, factor graphs can be used as a unifying abstraction language for building simultaneous localization and mapping algorithms. Furthermore, factor graph and Bayes tree frameworks — which are in used with the iSAM2[1] algorithm — have been very successful at large scale multi-sensor data fusion; the iSAM2 algorithm, however, requires the navigation system developer to only use normal distributed uncertainty (Gaussian) measurement models.

The robustness of the iSAM algorithm is known to be a problem when "bad", inconsistent, or non-Gaussian measurements are added to the factor graph model [181, 219]. This thesis is charged as a remake of the iSAM2 algorithm, by using the Bayes tree approach, but allowing non-Gaussian uncertainty models into the factor graph – called the *Multi-modal iSAM* algorithm.

The intention is to expand modeling and inference, using factor graphs, to encapsulate multi-hypothesis tracking within the multi-modal framework by generalizing the iSAM2 algorithm with multi-modal belief propagation on the Bayes tree. Generalizing to belief propagation allows inference with more complicated likelihood and prior uncertainty models. The ability to defer uncertain associations to the back-end inference, will allow the inference technique to consider

[1] incremental Smoothing and Mapping, Kaess and Dellaert et al. [115, 116].

the aggregate of all uncertainties and stand a much better chance of resolving the correct associations. This new freedom allows us to more accurately describe the measurement uncertainty models as the factor graph is constructed.

The premise of this work is to address the underlying crux problem known as the Chapman-Kolmogorov transit integral, and is specifically discussed in Section 5.4. Our approach to the Chapman-Kolmogorov integral is divided into convolutions and products of infinite functions, and was inspired by methods employed by Tanner and Wong [221], Gelfand et al. [71], and Sudderth & Ihler et al. [218].

1.3.2 Research Scope

This thesis focuses on developing a navigation solution to establish a sufficient level of location awareness for human supervised operations of an autonomous systems mentioned in Section 1.2. Given a working list of robotic applications, we re-factor the problem and transition our focus away from a particular robot and instead investigate the dominant sensing modalities, combined inference, and state estimation in detail. This work includes computational intelligence — built into the software and hardware computing elements — using the available sensing modalities to arrive at navigation state estimate.

Our approach should allow the navigation system designer the maximum freedom to express measurement uncertainty models which may be encountered by the variety of sensing modalities. In particular, we note that automated navigation systems involve a degree of computational decision making which may introduce greater uncertainty in measurements, such as data association type problems where valid measurements are made to ambiguous references in the world. Chapter 3 details how a front-end robot navigation process can combine four groups of factors for probabilistic modeling of the navigation problem with factor graphs.

1.3.3 The Multi-modal Aspect (Robustness)

By multi-modal we mean that almost all of the low likelihood hypotheses are not important and will be discarded to reduce the computational burden. One contributions of this thesis will be to discover how to develop an algorithm that focuses computation on a couple of the most likely modes without sacrificing accuracy on the dominant modes, and managing computational complexity with some hypothetical dial where the user can select how many modes and/or fidelity the solution should try recover. Our's is a full sum-product inference computation, rather than max-product or loopy belief simplifications which are generally used.

We decided to follow a nonparametric route, which encapsulates the notion of multi-modal inference in *belief*, and bars us from individually tracking each mode, and allows us to track highly non-Gaussian posterior probability densities. A multi-parametric mixture model of say Gaussians, will allow multi-modal tracking, but not quickly returns to heuristic selection techniques. This choice would further allow us to capitalize on existing theories and methods already developed in other disciplines.

For example, work on Random Operator Theory shows the deep theoretical study of generalized statistical inference, see Prokhorov [190] and Skorohod [209] is an entry point into literature outside modern robotics. That work concentrated on fixed point theorems and contraction mapping arguments for developing general algorithmic approaches. Yet more work is looking into higher dimensional Hilbert feature space embeddings to improve linearity, or trying to construct probability theory as groups as a powerful mechanism to analyze and develop algorithms.

1.3.4 The Inertial Aspect (Bandwidth)

Inertial sensors are a corner-stone of modern high-bandwidth and real-time navigation systems — see Farrell [56], Groves [79], or Titterton et al. [229] —, with a difficult but successful history, see Mackenzie [144] and Mindell [150]. Inertial measurements offer a means to "black box" dead reckoning, but suffer from measurement bias offsets and nonlinearity errors. For example, large errors in the position estimate result from integrating a gyroscope bias offset, which produces an orientation error and in turn results in incorrect gravity compensation being double integrated.

We purposefully include inertial navigation as the gateway to real-time capable and robust navigation. Besides the ability to track fast vehicle dynamics, inertial sensors boast major statistical advantages, including highly repeatable behavior of unimodal measurements (strongly Gaussian). The unimodal nature of inertial navigation (odometry) will play a key role in making practical multi-modal systems viable, by helping to isolate dominant modes and discarding low likelihood modes as the vehicle moves through the world – consider only a few possible hypotheses can exist from a collection of measurements with approximately known odometry. The navigation systems for robots illustrated in figs. 1-2 and 1-3 will require proper inertial odometry type measurement models, which and is developed as a core contribution of this thesis.

1.3.5 The Modeling & Prediction Aspect (Trajectories)

Knowing that state estimation, localization and mapping are generally used for controlling a vehicle towards some user desired set points, we also discuss how inference over past measurements blurs the line to future trajectory planning. Existing guidance and control systems assume a single time instant snapshot of vehicle state and user intention, in order to simplify the connection. This simplification, however, gives rise to methods like model predictive control and trajectory optimization as mechanisms to cater for situations not covered by the single time snapshot approach.

Predictive modeling of vehicle dynamics is the major connection between state estimation and control of a dynamic platform. We know from trajectory optimization and model predictive control that vehicle models can be used to enforce feasible and under-actuated solutions in the vehicle state. In addition, model predictions can be used as a measurement to aid in the navigation state estimation process with the expected dynamics of the robotic vehicle.

A unified state estimation and model predictive trajectory planning framework should allow for tight integration between sensory information over some history window of measurements the desired future state. This abstraction implies some kind of joint inference between past and across real-time boundary to some hypothetical future state.

Our investigation will combining navigational sensor measurements and model predictions into the navigation system, and leave the actual actuator control and feedback control loop structure as out of scope. This thesis should is intended to support current development work towards a more common and centralized navigation system, i.e. an automated *location and dynamic awareness*, and will show how and where model predictive trajectories should be incorporated into the simultaneous localization and mapping framework. This thesis will not study how model predictive trajectory planning performs, but will be part of our future work. The far reaching extension is to capture multi-policy trajectory optimization with a multi-modal belief representation.

1.3.6 Out of Scope

While the algorithmic development is focused on efficient computations, this is not a one-size-fits-all, plug-and-play solution. The dominant focus of the thesis is how to develop a back-end inference solution which is common to multiple robotic platforms, as the largest common denominator in any navigation system. Front-end processes, particular to each robot or mobile platform, will be required to construct

the factor graph (see Chapter 3) which is then consumed and used for inference computations by the back-end process.

In terms of computations, we are not attempting to place all computations as a real-time system on a single small processing unit, and willing to explore larger computational methods to advance the science of navigation. The thesis will explore how real-time architectures can be assembled, but will not validate real-time performance on any particular platform. Multi-modal factor graph approaches will likely require larger compute capability, and this thesis will investigate the merits of such an approach while looking to identify ways to reduce the computational load. Finding reasonable system level architectures which would allow robots to offload large computations, while maintaining statistical exactness and achieving good robustness is more in line with this thesis.

We acknowledge that high quality results can be obtained with existing methods with large engineering teams, and are not competing with a highly engineered system such as the Google Tango device [2], or high class inertial and GPS based navigation systems. This study focuses on the architecture that would be feasible with technology that is available today, rather than spend excessive time on engineering solutions to one particular problem which has limited transferability to other robotic platforms.

The outcome of this thesis is to find a viable theoretical avenue in robust navigation, combined with high-bandwidth and real-time inertial navigation, and through examples show that each of the specific elements discussed do in fact work. The intention of this research is to explore navigation systems which, when engineered, can achieve desired computation time and performance criteria in operational environments which are not accessible with the solutions in use today. This thesis marks a new direction for addressing current robustness problems, rather than incrementally refining the prevalent unimodal, Gaussian, and null-hypothesis assumptions found in literature today.

Regarding future model (trajectory) prediction and control: We are not working to implement feedback control architectures for affecting system actuators, but do spend some effort in advancing the integration of trajectory planning and model predictive control strategies into a common navigation inference framework.

1.4 Thesis Roadmap

Chapter 1 identifies inference in robotic navigation as an important and open area for research, and defines the scope of work covered in this thesis.

Chapter 2 investigates previous work in state estimation and mapping, which problems still need to be addressed, and how different principles of existing methods have inspired our work.

Chapter 3 describes how to assemble a consistent joint probability factor graph representation over navigation variables of interest using existing — and introducing several new — parametric and nonparametric sensor likelihood models based on measurements from real-world events.

Chapter 4 derives and details one of the measurement models in particular, namely the new continuous time and second order inertial odometry measurement factor for capturing high-bandwidth motion, and achieving real-time navigation performance.

Chapter 5 relaxes parametric optimization to a more general nonparametric inference over joint probability factor graph models described in Chapter 3, and presents our inference algorithm called *Mutli-modal iSAM* (incremental Smoothing and Mapping).

Chapter 6 is dedicated to canonical multi-modal examples to help understand the nature of nonparametric solutions.

Chapter 7 discusses centralizing the factor graph around a navigation server rather than using a central front-end robot process to pull different parts of the solution together.

Chapter 8 presents results from real navigation use cases, including hand held monocular visual-inertial navigation, the BDi Atlas humanoid robot, and underwater beam forming acoustic robot, and a larger scale wheeled robot.

Chapter 9 draws conclusions about the work in this thesis and discusses future directions that may be taken.

1.5 Critical Analysis

A navigation system can be analyzed with three fundamental questions: Is the state estimate improving when more time is spent in the same area; does the relative state estimate between vehicle and objects improve as the robot moves closer to the objects for the first time; and does the navigation system consider future trajectories or only existing measurements. In this thesis we propose a solution that answers affirmative to each of these questions.

Literature suggests a strong synergy between belief interpretations and consensus among multiple hypotheses. Most existing methods try find ”outlier” measurements and remove them as incorrect hypothesis, but such approaches are

prone to parameter tuning heuristics, almost zero guarantees, and result in discarding valid measurements which suffer from bad associations by the front-end processing. The risk of not dealing with bad associations usually results in unpredictable and catastrophic failure of the navigation state estimate, given that such measurements are incorrectly treated as having Gaussian uncertainty.

A robust, multi-sensor data fusion system has long been sought after by the inertial navigation community, and is one of the core contributions of this thesis. Filtering solutions have long dominated inertial navigation, but we expect them to transition to factor graph approaches in the future. This thesis intends to combine high-bandwidth and real-time inertial navigation capabilities with robust multi-sensor data fusion.

In general, common sensors perform well under controlled circumstances, but only measure partial navigation state information through some uncertain and nonlinear model. A generalized navigation scheme that can combine large amounts of varying measurements and extract the most likely consensus modes from sensor data, assuming that measurements from one reality will eventually result in only a single plausible dominant mode estimate.

A core contribution of this thesis is to show that larger scale non-Gaussian, multi-modal type uncertainty models can be successfully used in a factor graph, and perform efficient inference over the ensuing factor graph models. Our approach will leverage the benefits of the Bayes tree from iSAM2 [115] to allow more powerforl factor graph representations for robust navigation systems of practical sized problems.

Biological systems show that a generalized navigation system should be possible: Humans can take the controls of a wide variety of platforms, such as cars, planes, helicopters, space ships, or submarines, and are able to resolve a dynamic navigation state estimate from available data. Dynamic navigation in this context means to have intuition about vehicle dynamics in the near future.

Autonomous systems are operated by people who need an intermediary interface, which ultimately involves visualization of robot state and expression of user intentions. User driven automation require a common reference frame so that users can understand how the robot perceives the world, and the navigation system is the key component that establishes the common reference frame.

Lastly, a navigation system is developed by a group of people using the tools at their disposal. To this end, our development will focus on supporting designers of navigation systems for robotic applications by simplifying sensor modeling and data interaction tasks. Chapter 7 promotes a centralized factor graph in a client-server model as the common platform to bring together all the pieces of a

navigation system.

1.6 Conclusion

This thesis is about robot navigation and state estimation. This chapter motivates why a reliable, high-bandwidth navigation solution is vital for increased autonomy in robotic systems – to let robots *understand* their surroundings with some level of *location and dynamic awareness*. A navigation system must allow users and algorithms to freely extract spatial relationships from a synthetic reconstruction of the world, to support human instructed inspection or manipulation tasks.

Robust navigation will inevitably encounter non-Gaussian measurements and will have to draw on a broader class of measurement uncertainties such that a consistent navigation inference problem can be propagated to the back-end solving process. Current unimodal belief propagation on the Bayes tree should be extended with non-Gaussian models to accommodate, for the first time, multi-modal belief propagation, without breaking the incremental update mechanics achieved with the original iSAM2 algorithm. The transition to nonparametric methods is most heavily inspired by ground breaking work from Kaess & Dellaert et al. [115], Tanner & Wong [221], Gelfand et al [71], and Sudderth & Ihler et al. [218].

All the robotic platforms shown in Fig. 1-1 rely on Kalman filtering based data fusion solutions as primary means of navigation. There are some secondary use cases and developments with factor graphs on those platforms, but nothing to suggest that factor graphs will take over as the primary means of doing state estimation. This thesis is focused on promoting a multi-modal and non-Gaussian approach based on factor graphs as the primary means of navigation in automated (robotic) platforms.

Chapter 2

Literature Survey

Chapter 1 isolates an area of research relevant to robotic navigation. In this chapter, we consider previous work in state estimation and mapping. While our approach makes a significant break for current approaches in SLAM[1], the principles from inspiring works remain relevant. Chapters 3 and 4 present new ways to construct probabilistic likelihood models for a variety of sensor measurements. Chapter 5 discusses how inference with the newly developed parametric and nonparametric sensor measurement likelihoods.

2.1 Introduction

Improvements in technology have fueled the development of many approaches to state estimate of localization and mapping. A looming question is whether the variety of state estimation techniques are leading towards towards a centralized localization, mapping, and planning approach. Literature on state estimation, localization, mapping, signal processing and computer vision are all pertinent to research on navigation systems.

Simultaneous Localization and Mapping (SLAM) research has been at the forefront of jointly estimating vehicle location and opportunistic features state in the surroundings. The big question in state estimation, following the advent of the Kalman filter in the early 1960's, is which data fusion process or architecture will replace existing filtering approaches.

The classic localization only problem is far simpler, with lower computational cost and algorithm complexity. Localization only systems assume stable and

[1]Simultaneous Localization and Mapping

unique long term references, for example GPS satellite signals, will be available during vehicle operations. SLAM systems, however, face the key problem of simultaneously inferring state of the vehicle and unknown surrounding using many partial and uncertain sensor measurements, while being robust to ambiguity in the measurements and the navigation problem.

Navigation solution failures generally stem from inconsistent measurements when data association to known references or features fail. Data association failure in ambiguous sensor measurements is the process whereby the navigation software is forced to assume Gaussian only measurement models under incorrect association between predicted and true references in the world.

The process of reducing all ambiguous measurements to Gaussian uncertainty models inevitably result in inconsistent measurement models entering the navigation solution. Inconsistency here has the formal meaning, Groves [79], that the user models a measurement's as a highly unlikely event – such that all measurements do not adhere to the frequency of occurrence specified in the user model.

A strong synergy exists between probabilistic belief interpretations and consensus among multiple ambiguous measurements, but thorough literature on using multi-modal belief uncertainty for the navigation task is less prevalent. This chapter reviews literature from state estimation, mapping, and statistical inference disciplines in preparation for our contributions towards multi-modal navigation.

Combining robust localization and mapping with high-bandwidth state estimation will require computation over widely varying timescales. This combination produces a systems engineering question of how to assemble different sensing and processing modalities into a common framework.

This chapter discusses different navigation sensing modalities, and look at how inertial navigation systems came to play a central role in present day autonomous systems. Followed by how the Kalman filter data fusion concepts develop deeper understanding into more common multi-sensor inference inference, specifically the cyclic factor graph graphical representation for navigation. Re-factoring the cyclic factor graph to an acyclic tree have become a critical staple of parametric optimization. Existing stochastic inference over graphical models, developed by the statistics community, may well shed light on more general inference for non-Gaussian type belief. The influence of technology on assembly of navigation solutions is also briefly reviewed.

2.2 Sensing Modalities

Broadly speaking, three major sensing types are of interest to navigation: (i.) active reference from external sources such as radio or acoustic ranging; (ii.) passive, self-contained measurements such as inertial measurements or camera; and (iii.) active, self-contained measurements such as lidars or Doppler velocity logs. The different sensing modalities lead to several trade-offs in how to fuse, into a coherent estimate, information about the vehicle and world state.

2.2.1 Triangulation and Ranging

Triangulation and ranging remain the primary means of navigation in many localization systems used today, see Fig. 2-1 for ranging type navigation example. For example, common GPS navigation combines simultaneous ranging measurements from several global navigation satellites [79] into a Kalman filter position and velocity state estimate. Satellite based ranging measurements replaced previous hyperbolic radio ranging systems such as LOng RAnge Navigation (LORAN) and Decca.

Key problems with ranging mechanisms [144] (although they can provide near absolute navigation measurements) is that ranging to external reference systems are comparatively slow to vehicle dynamics, and easily disrupted by natural or intentional means. Additionally, time-of-flight distance measurements require additional information such as global timing or location of reference beacons, which

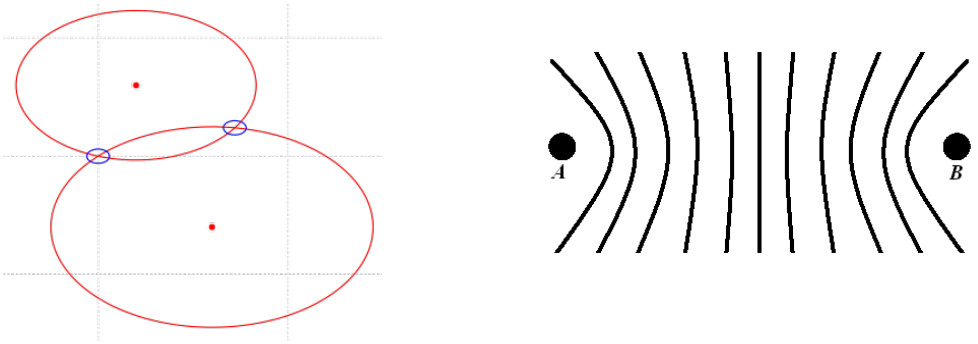


Figure 2-1: Illustration of ambiguous ranging navigation. *Left*, shows how absolute timing allows absolute ranges to resolve two possible positions. *Right*, shows relative timing between two concurrent time-of-flight signals for hyperbolic arcs of possible vehicle location [239].

in turn increases the cost of these systems. For example, hyperbolic ranging systems do not have access to an absolute time reference and measure the difference between expected signals instead. As a result, combinations with other sensing modalities, most notably inertial navigation, has became a popular trade-off.

2.2.2 Platform Inertial Navigation

The aerospace community have long contributed to in-situ navigation with work on early aircraft instrumentation through missile guidance. Missile guidance led much of the early development work on platform type navigation systems — greatly influenced by Draper et al. [46] — , leading to patents such as the famous Polaris ballistic missile system, Hall et al. [82]. Wonderful historical discussions on the socio-economics precipitating from the space/arms race between the global super powers were studied by Mackenzie [144], leading to the Apollo Lunar missions navigation technology studied by Mindell [150].

In turn, platform based inertial navigation systems (INS) developed from earlier gyrocompassing systems, which are commonly used for independent, self-contained north finding and latitude estimation at sea. Gyrocompassing follows from the discovery of the Foucault pendulum and early attempts by Van den Bos in the 1880's to develop prototypes had proved difficult to master.

Mechanical gyrocompassing platforms float level to gravity and use a control system to steer measured earth rotation rate to zero [106], thereby isolating East-West orientation. A further capability for estimating latitude involves comparing vertical and North aligned rotation rates which should sum to $15^o/hr$ rotation rate.

Metal hull ships, during the first World War, obfuscated magnetometer readings and kick started detailed technological and scientific development in inertial guidance. The ferrous hull problem was exacerbated in submarines, leading to rival gyrocompassing technologies between the earlier Antshutz-Kaempfe systems from Germany and copied Sperry systems from the United States, calling on Albert Einstein as expert witness for courtroom patent disputes[2].

Work on gyrocompassing led to great scientific works, including that of Schuler [243], who derived the de-facto process-noise parameter selection criteria for inertial navigation system, still in use today. The Schuler pendulum is specially chosen to not be perturbed from the local vertical from any vehicle accelerations by choosing the oscillation frequency at $84.4\ mins$ – that is a pendulum having the whose length is the radius of the earth.

[2]Historic article in Newsletter from the Institute of Navigation.

Starting in the 1940's, missile inertial guidance platforms used the gyroscopic effect to establish a rotationally fixed reference frame in three dimensional inertial space, see Britting [25]. The vehicle orientation angles could then be measured relative to the stable platform at any time. By mounting and integrating the measurements from accelerometers on the platform, we can estimate changes in velocity and even doubly integrated for position in the stable platform reference frame.

Over time, position estimates would be adversely affected from integration of errors from gravity coupling due to platform misalignment[3] and accelerometer measurement bias offsets.

The best solution was to combine long term stable navigational references as a calibration reference to correct the platform predicted navigation solution and implicitly estimate sensor bias offsets. Celestial star tracking was an early popular choice, which worked at high altitude for missile and aircraft guidance, but was not ubiquitously available in all weather or covered operation regimes.

The Apollo lunar program of the 1960's provided a great backdrop for showing what a celestially aided inertial navigation was capable of doing: Rocket launch, lunar landings, and lunar orbital rendezvous, inter-celestial travel, and re-entry were all critically dependent on inertial navigation based fly-by-wire systems [150].

With the Apollo lunar program, the Command and Lunar module's onboard inertial navigation system was aided through a Kalman filtering process, using observations from a sextant star sighting mechanism [144]. Early uncertainty and the unexpected complexity of the Apollo onboard inertial navigation system had resulted a project level decision to switch the primary navigation to stereographic ranging measurements from radio stations on earth [150]. The onboard inertial navigation did play a critical role in lunar landings, where the system was also combined with a ground referenced Doppler radar velocity system. A further interesting aspect is that the space traveling Module navigation state parameters could be updated or reset by controllers from ground based computers.

After the Apollo program, inertial navigation went through a difficult development period, and only became popular for commercial use with the requirement of a bespoke system to be used on the Boeing 747 intercontinental airliner. From this point, security in the field of inertial navigation was established, but development in data fusion techniques slowed down and most development was done at sensor computational hardware level [229].

Digital technology and more computational resources has had a major influence on inertial navigation: towards reducing the mechanical complexity and errors, size, power consumption, extending maintenance intervals, and reducing

[3]Problem of the vertical

cost. Strapdown inertial navigation became a mainstream commercial technology by the mid 1980's [203]. Strapdown systems keep track of the rotationally fixed reference frame through electronic propagation of the rotational estimates [30]. Gyroscope technology favored closed-loop—or nulling—measurement strategies, with dynamically tuned, ring laser, tuning fork, hemispherical, and fiber-optical gyroscopes offering great performance/cost trade-offs [229].

A common theme from previous work in inertial navigation is that sensor errors can be well modeled and dynamically estimated in real-time, resulting in a major improvement in velocity and position estimation which in turn support real-time tracking of opportunistic references.

Industrialization of INS/GPS type navigation systems became the priority. The next big break would follow the commercial introduction of micro machined electro-mechanical (MEMS) gyroscopes and accelerometers, and how to combine them with alternative localization and mapping algorithms developed by the robotics community. We identify this as an area this thesis can contribute to, and direct the reader to Chapters 3 and 4 for more details on *inertial odometry*. We also note our contribution is not limited to low-cost inertial sensing, but actually focused on high quality computation that can easily be replicated between high and low cost sensor technologies.

2.2.3 Towards Digital Multi-sensor Fusion (INS/GPS)

Aided inertial navigation systems have become the workhorse for nearly all autonomous vehicles today. Inertial navigation solutions are usually aided by independent measurements beacon range or bearing, or Doppler velocity (body or global relative) measurements. Chapters 3 and 4 presents our understanding of modern inertial odometry in concert with other navigation measurements.

Widespread public access to the American GPS satellite navigation system, starting in the 1990's, allowed self contained and high-bandwidth inertial navigation systems to be combined with slower but stable absolute position updates from GPS satellite ranging systems [57]. The dominant data fusion method that emerged from this work is an extended Kalman filter (EKF) design, estimating either the total or error states of the system, in feed-forward or feed-back configurations. Industrial applications commonly use Kalman filters to dynamically estimate a single time-slice of many IMU and GPS sensor calibration parameters; orders as high as 90 system error states have been reported. Inertial navigation systems are described in great detail by Titterton et al. [229], Groves [79] and Farrell [56].

A common approach is to use inertial sensors for capturing high bandwidth dynamics and gravity measurements while radio based ranging measurements, such as a global navigation satellite system (GNSS), preserve the long term accuracy of the navigation solution. GNSS aiding is either done by cascading the INS/GNSS filter behind the GNSS solution filter (loosely coupled), or directly using pseudo range satellite measurements in the INS/GNSS measurement model (tightly coupled) [49]. In the extreme, inertial information is used to also aid satellite signal tracking (deeply coupled) [229]. The Kalman filter as data fusion technique is discussed in Section 2.4 following the discussion on relevant navigation sensing modalities. Recent work by Barfoot [14] gives a survey of current data fusion strategies used by the robotics community.

2.2.4 Emergence of New Sensing Modalities

The quest for ever more autonomy and improved low-cost navigation systems has fueled research into alternative navigation methods. Notably, efforts by the robotics community to navigate autonomous vehicles in in-door or sub-sea (GPS denied) environments, and have demonstrated a vast array of new localization and strategies. Many technological improvements, such as computational power resources, lower cost sensors, and better sensor performance have allowed significant progress in the navigation data fusion discipline.

Terrain Relative Navigation

Many situations such as submarine guidance, flying out of range — or in radio silence — prohibit access to active external navigation beacons. Instead, passive referencing mechanisms were developed whereby a computational system can help resolve a navigation solution using features of interest in the world surrounding the vehicle. Early aircraft systems could be pre-loaded with reconnaissance imagery which are then re-observed by cameras onboard the vehicle. Feature detection and recognition systems would then produce prior estimates of where the vehicle was in relation to the previously surveyed area [79]. These techniques are known as terrain relative (referenced) navigation, or TRN.

Terrain relative and celestial navigation aides were complimentary given the operation altitude and can be integrated with the inertial navigation system through some feedback correction or data fusion process. However, the size, power, computation and cost constraints of celestial or TRN aiding would often limit a system to only include a smaller subset of these aiding measurements. Ter-

rain relative or celestial navigation are both still limited by weather conditions. As a result of high availability requirements on cruise missiles and other stand-off type systems, topographic and altitude type terrain referencing systems were developed.

Topography based navigation, as is common on Tomahawk missiles today, use a histogram filter to correlate a short history of measured altitude over ground measurements with a pre-loaded topographic map of the area. The combination of local odometry estimates with the varying ground altitude quickly constrains the navigation solution position uncertainty to a small area over the known topography map. Histogram filters are a type of recursive filtering technique (discussed in Section 2.4) using a fixed grid of possible world or body frame positions.

Velocity over Ground (Doppler Measurements)

Maritime and underwater navigation generally do not have access to clear imagery or topograhical information of the sea floor terrain, see Kinsey et al. [123] or Paull et al. [186] for in depth surveys on state-of-the-art underwater navigation techniques. Sonar measurements, and specifically Doppler acoustics, can provide velocity over ground estimates.

Underwater autonomous vehicle navigation has further developed aided inertial integration strategies using a Doppler velocity log (DVL). INS/DVL systems combine accurate body relative acoustic Doppler velocity measurements, near the sea floor, with rotation rate and acceleration measurements in a similar Kalman filter data fusion architecture [56].

Body relative ground velocity estimates can be integrated into position estimates relative to the starting position, often using an attitude only estimate from and IMU to de-rotate vehicle dynamics into a stable vehicle centered reference frame. Similarly, there are many situations where aerial vehicles may be flying over unknown terrain, but have access to Doppler radar type measurements See Farrell [56] for a similar loosely coupled Kalman filtering architecture to combine body relative Doppler measurements with inertial navigation type dead reckoning state predictions.

Ultra Short Baseline (USBL) Receivers (Nonparametric)

The relative geometry of received radio or acoustic wavefronts can be determined through beam-forming calculations from a transducer array to measure varying arrival times of incoming signals. Beam forming techniques, Van Trees [233], can

recover the orientation angle from where a signal originated. Time-of-flight ranging techniques are discussed separately in Section 2.2.1.

Beam forming techniques allow the array size to be reduced to the order of signal wavelength. As a result, higher frequencies allow receivers to be small and portable. USBL systems have become popular for underwater robotic operations for both tethered and untethered vehicles. However, automated fusion of USBL data with solutions from vehicle inertial navigation systems have been limited. The predominant use case of ship mounted USBL systems in the underwater robotics case, seems to involve human in the loop operations for all measurement interpretation and integration into the operational system at hand. Automated inclusion of USBL type measurements in robotic platforms seems to be limited.

The most recent technological development is a chip scale atomic clock [44], at relatively low-cost. The combination is leading to several new developments in autonomous vehicle navigation sensors, such as a one way travel time vehicle mounted inverted (iUSBL) system [163]. Both acoustic and radio based beam forming measurements provide an excellent source of navigational information, but easily corrupted by noise or multi-path interference in crowded and confined spaces.

As a result, the signal processing and understanding of these type of sensors become critical to their industrial application. It is critical to have have a data fusion approach which is able to interpret highly non-Gaussian measurement uncertainties and fuse the information with other sensing modalities for a coherent state estimate. A core contribution of this thesis is the ability to combine nonparametric type measurement likelihood beliefs into a common factor graph description and be able to perform meaningful *sum-product* type inference on system state variables.

Laser Scanners and Depth Fields (Lidar)

Two dimensional and three dimensional depth fields type sensors have had a major influence on navigation and mapping type algorithms. These sensors, when combined with feature detection and tracking processing, provide strong range and bearing constraints for any navigation solution. Bry et al. [27] show how an inertial navigation solution can be combined with a planar laser scanner and a prior known map to successfully guide a small aerial vehicle indoors on a prescribed trajectory amongst many objects.

The Carnegie Mellon *Multisense* stereo camera and spinning lidar used by BDi Atlas during the DARPA Robotic Challenge [54] provides great three dimensional

position constraints and object detection [55], however, this system separates mapping and localization tasks into separate stages and avoids the desirable, fully fluid mapping and localization framework discussed in Chapter 1

Self driving cars, starting with the DARPA self driving car challenges, has made considerable use of spinning lidars for mapping the surroundings slaved to a separate INS/GPS localization solution. MIT's approach to the DARPA urban challenges relied on autonomous decision making from a locally computed map of the surroundings [134]. Instead the winning Stanford approach followed (with GPS) a preplanned route more closely and avoided locally observed obstacles that would interfere with the intended trajectory path [155].

Tracking and Mapping

Methods such as Parallel Tracking and Mapping, Klein et al. [126], use laser scanner, lidar, or camera measurements for separated localization and mapping tasks. Parallel tracking and mapping approaches are a preamble to simultaneous localization and mapping approaches.

Autonomous underwater inspection operations, much like self driving cars, are another area seeing significant development, where vehicles are now able to successfully use lidar measurements to map underwater structures. Similar to the earlier self driving car technologies, and INS/DVL type navigation solution is used for localization, while lidar measurements are slaved to the location solution to produce a secondary map estimate, see Lockheed Martin's approach [148] is another example where maps are generated after localization has been solved.

Red-green-blue and depth (RGB-D) sensors, such as the Microsoft Kinect, have greatly influenced development in 3D mapping, see Han et al. [83] for a general review of recent work. Three dimensional lidar measurements (such as RGB-D) give extremely dense information, sufficient to compute the localization and mapping solutions in parallel. For example, Newcombe et al. [173] and [171] demonstrated dense tracking and mapping (DTAM), which was able to resolve both location and mapping using only a RGB-D sensor. Later work by Whelan et al. [236] proposed a method to combine sparse and dense methods using RGB-D, and deformable map objects in DynamicFusion by Newcombe et al. [172]. These approaches compare consecutive three dimensional point clouds to find an odometry solution [235], and the points are reprocessed into mesh map of the local surrounding objects.

Earlier work from Klein et al. [126] uses multiple passive cameras for parallel tracking an mapping (PTAM). It is assumed that a PTAM derivative, using a loosely coupled inertial navigation solution through a Kalman filter framework is

used in the Microsoft Hololens project[4].

These and similar approaches produce visually appealing reconstructions of the surroundings, while solving the localization problem of the sensor platform. These approaches are, however, totally dependent on the lidar sensed information. Localization and mapping fails when the scene is obscured by sunlight, fog, dust or silt. Each of these approaches stand to benefit a great deal from data fused results where other sensor modalities can improve the resolution and robustness of the estimation process.

Lidar based approaches are also information intensive, often imposing bandwidth constraints on both the motion of the sensor, as well and the electronic communication and processing. Long term maps produce large volumes of data, intricately tied to the map itself, see Whelan [237, 238]. Johannsson [105] suggests a long term solution by combing a sparse index of location to which starves the surrounding map information to a separate data store.

A last example of using planar lidar information is one state estimation and mapping solution used for the BDi Atlas humanoid robot. The state estimate system described in [55] uses an iterated mapping and localization solution. The robot starts at some initial condition, standing based on position, velocity and orientation estimates from a inertial navigation state estimate. A spinning lidar from a Carnegie *Multisense* device is used to generate a three dimensional map of the local surroundings, based on the current position state estimate.

The mapping process is then paused and a localization service then takes over, using the map as staring environment to produce position updates, in a loosely coupled fashion, to the inertial navigation solution, using the same code developed by Bry et al. [27]. The data fusion is done with a common 9 state loosely coupled Kalman filter framework: position, velocity and orientation. Inertial sensor bias terms are not estimated. Once the robot approaches the end of the current mapped region, the localization process is paused and the mapping process is again used to develop a new map local to the current robot location. Robot leg kinematics, using joint angle measurements, are also used for navigation position updates and footstep location estimation.

Kinematic Linkages

The advent of kinematic biped and quadruped robots, requiring great balancing finesse, have fused local inertial navigation solutions with leg kinematics. Joint encoder angles are used together with kinematic models to estimate the position of

[4]Speculated.

footsteps and the body relative to some starting position, see for example Koolen et al. [128]. Similar solutions, including Bloesch, Alacantarilla, Rotella et al. [5,24,199] respectively, use Kalman filtering to infer the state as a single snapshot estimate, using only a single snap transform to place new footstep estimates in the world.

From experience, two major problems complicate the use of joint encoder angles. Uncertainty in mechanical tolerance and backlash of actuators result in large kinematic uncertainty. Second, contact estimation with the floor or objects such as stairs or platforms can be difficult, and results in a delicate parameter tuning exercise of balancing foot contact accuracy with tolerance to bumping and dragging actuators against surfaces and obstacles.

Experience with this type of system has helped motivate the suggested approach discussed in Chapter 3, where trajectory planning and factor graph type inference are combined for a more resilient robot navigation framework. See Fig. 3-1 and related discussion for more details.

Camera only (Monocular or Stereo)

Cameras are a self contained and passive sensing modality, see Harley and Zisserman [88] for an in depth look at imaging geometry. The Apollo program started using cameras for remote sensing, using imagery to build topographically accurate surface maps. Bundle Adjustment, Triggs et al. [231], and Structure from Motion (SfM), Horn [93], techniques were developed to combine multiple images into a simultaneous localization and mapping solution. Later methods would extract distinct features of interest, such as Harris corners [87], and SIFT descriptors, Lowe [139], which can be corresponded across images. Multiple feature detections are then used to assemble a least squares objective, which is minimized to find a mean parameter estimate (maximum likelihood or otherwise). The solution for monocular only solutions have the additional problem of unconstrained scale, since no depth or odometry information is available.

The relative transform between nearby images can be measured using Nister's eight point [175] or Horn's relative orientation [94] techniques. These techniques amount to visual odometry methods. Kanade-Lucas-Tomasi (KLT) feature tracking [230] is an in-situ method to track how feature points move through an image sequence, and used as a common base for several of the methods discussed above. As a side note, our discussion returns to these relative transformation techniques in Section 3.4.6, where a new compound factor, or multiple feature factors, is assembled.

Continuing with visual odometry techniques, Davidson [40] suggested adding

a constant velocity model to visual feature tracking, using a Kalman filtering framework, to fuse the camera only measurements into a real-time SLAM solution. Monocular camera LSD-SLAM, Engel et al. [51], avoids regularizing assumptions such as constant velocity but resorts to explicitly parameterizing a map scale parameter, thereby isolating the non-observable scale problem. The parameterized solution can the be fit to metric scale using additional information.

In 2001, Qian et al [191] suggested incorporating inertial data with a structure from motion procedure to aid in scale observability; this was later suggested by authors such as [177] as well. This data fusion further enables high-resolution trajectory estimation between camera frames and coasting through featureless image sequences. However, these methods do not model raw inertial sensor error terms, and in effect require high-performance IMUs or ultimately incur little dependence on poor inertial measurements.

Loop Closures (Vital Data Association)

Loop closures are not limited to any particular sensor and can involve any of the sensing modalities. The term loop closure refers to positively identifying a geometric relation to previous objects or pose positions seen by the navigating platform, and has been the subject of many investigations as summarized by Lowry et al. [140]. A loop closure could be as simple as a user input statement, confirming that the navigating platform has been returned exactly to a previous position, using some accurate mechanical alignment technique. On the other hand, loop closures could complicated interactions of suspected observations of previously identified features or objects in the world. Observations of objects could be via radar, sonar, camera, or otherwise.

Loop closures provide a fantastic source of navigation information, since an entire loop and many sensor error parameters can compensated knowing a segment of the trajectory produces a loop to the same location. In a relative mechanization, as suggested by Leonard et al. [135] and others, the loop closure point means two parts of a time separated poses are known with high accuracy to each other, and the midpoint in the trajectory between the loop closure points is in greatest error. The problem with loop closures is to automate a data association framework that does not make mistakes, or miss too many potential loop closures. Incorrectly accepting an erroneous loop closure can have catastrophic consequences to a SLAM type approach, given the inconsistency of the measurement and prevalence of Gaussian error assumptions.

2.3 State-of-the-art Fusion Strategies

The availability of vast amounts of navigational sensory data raises the question on how best to fuse multi-sensory data into a common state estimate. Our discussion follows the a similar chronology as scientific development in the understanding of inference, using aided inertial navigation as a preamble to Kalman filter dominated techniques. By generalizing the Kalman filter to a hidden Markov model, and working our way towards general factor graph models, currently algorithms to infer state using factor graphs is discussed.

2.3.1 Loosely and Tightly Coupled INS/GPS

Starting in the 1990's, the popularity of INS/GPS systems lead to three major categories of Kalman filter based state estimator coupling schemes. Loosely coupled schemes compare the INS position output with that of an independent GPS receiver (which internally also uses a Kalman filtering framework), Farrell and Barth [57]. Since Kalman filters assume that all process inputs and observational measurements are statistically independent, the tightly coupled scheme directly predicts and measures individual satellite time-of-flight range measurements in one centralized Kalman filter [229]. The deeply coupled system, Gustafson et al. [80], establishes bi-directional data fusion by using the inertial solution to predict where GPS correlator tracking loops should find satellite ranging signals.

Attitude heading and reference systems (AHRS) generally combine gyroscope derived orientation with accelerometer measured gravity in a Kalman filtering framework, Farrell [56]. AHRS are not considered inertial navigation systems, since they do not exploit double integration of acceleration measurements. Our focus is on fully fledged inertial navigation solutions.

2.3.2 Loosely Coupled Visual-Inertial Filtering

A strong synergy between visual and inertial sensors [36] has resulted in various visual-inertial navigation systems (VINS) being developed. Loosely coupled visually aided inertial navigation by Veth et al. [145,234] combines position or velocity measurements from visual odometry as updates to a Kalman filter, similar to INS/GPS systems. This arrangement can be considered as loosely coupled, since separate position estimates are compared.

2.3.3 Tightly Coupled Visual-Inertial (MSC-KF)

Major contributions to visual-inertial odometry came in 2007 with the introduction of the Multi-State Constrained Kalman Filter (MSC-KF) by Mourikis et al. [157, 158] and inertial structure from motion by Jones et al. [107, 108]. Both employ tight coupling between inertial measurements and visual feature tracking, in that feature position estimates are directly dependent on the state estimates from the visual-inertial Kalman filter.

The MSC-KF approach has one additional modification from a singular Kalman filter approach, in that landmark position states are separated out in a Rao-Blackwell fashion. That is the previous pose locations are estimated with Kalman filter, assuming the landmark positions are fixed. Once new pose estimates are available, the position of landmarks is estimated using a least squares approach, keeping pose positions fixed. Null projection, using the Schur compliment, is used to marginalize out pose information from landmark estimates before they are again used for updating the pose localizations, Mourikis et al. [158]. The motivation for this approach is to reduce the cubic dimension cost of Kalman filter updates. However, the MSC-KF approach is very similar to EKF-SLAM discussed in Section 2.5.2, and is known to be computationally less efficient than graphical model approaches, Dellaert et al. [41].

Refinements to the MSC-KF have improved filter consistency by restricting the state updates to respect the unobservable nullspace of the system, see Huang, Hesch and Li et al. [90, 98, 138], which enters the state through linearization assumptions and numerical computational issues. Filtering style visual inertial navigation system (VINS) generally estimate a single snapshot estimate of inertial sensor biases. There are several further works on visual inertial localization by filtering but are general variations of these presented above.

One notable issue with visual inertial filtering solutions is that both the prediction and measurement models are non-linear. It has been noted that performance may be improved by improving linearization points through iteration of the updates to the state vector. Since filter update steps are in essence Gauss-Newton updates steps, the work of Bell [16] stands a good middle ground between filtering and smoothing solutions.

2.4 Recursive Hidden Markov Model (Filtering)

The discussion will now focus on existing data fusion approaches, starting with the Kalman and particle filters as recursive implementations of the underlying hidden

Markov model.

2.4.1 The Kalman Filter (HMM)

The Kalman filter [118] was developed in the early 1960's and has become a popular method for data fusion into a common state estimate, and following the Apollo program defined a common blueprint for many state estimation systems in use today. All measurements $\mathbf{y}$ and inputs $\mathbf{u}$ are assumed Gaussian and uncorrelated, resulting in a recursive least squares solution to the state vector $\mathbf{x}$.

The filter operates in two stages, namely prediction steps according to model $\mathbf{f}(\mathbf{x}, \mathbf{u})$ which are driven by Gaussian input $\mathbf{u}$; and various update steps using sensor model $\hat{y} = h(x)$, driven by measurements $\tilde{\mathbf{y}}$. Kalman filter prediction steps are generally done at very high rate, with a few measurement update steps between from different aiding measurements, Groves [79] & Farrell [56]. The prediction of Gaussian state $\mathbf{x}$ and covariance $\mathbf{P}$ is described by:

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{K}(\tilde{\mathbf{y}} - \hat{\mathbf{y}}) \tag{2.1}$$

$$\frac{d}{dt}\mathbf{P} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{Q} - \mathbf{K}\mathbf{R}\mathbf{K}^T. \tag{2.2}$$

with Kalman gain $\mathbf{K} = \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}$, where process noise $\mathbf{Q}$ and measurement noise $\mathbf{R}$ covariances are used. Linearizations of the prediction $\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}^T}$ and measurement $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}^T}$ models are used for estimating the covariance.

Discrete time evolution equations are found by integrating the equations, which produces discrete state and covariance estimates ${x_{(k+1)}}^+$ and ${P_{(k+1)}}^+$. Prediction steps which increase uncertainty and update steps reduce uncertainty, Grewal [78]. At each time step, the posterior belief is marginalized to the current state and used as a prior. The state prior represents all previous information that has been marginalized out, while the current state variables $\mathbf{x}_{(k)}$ and measurement prediction $\hat{\mathbf{y}}_{(k)}$ are kept fluid.

Fig. 2-2 illustrates a graphical model representation, at time (k) of the state ${x_{(k)}}^+$ and covariance estimate ${P_{(k)}}^+$ which are propagated forward in time and updated by measurements $y_{(k)}$, and is commonly referred as a hidden Markov model (HMM). Markov, since each next state is only dependent on the previous state, and hidden since the deterministic instantiation of state vector $\mathbf{x}$ may contain latent variables that are not directly observable through the measurement model producing $\mathbf{y}$. The probabilistic understanding of the hidden Markov model was in

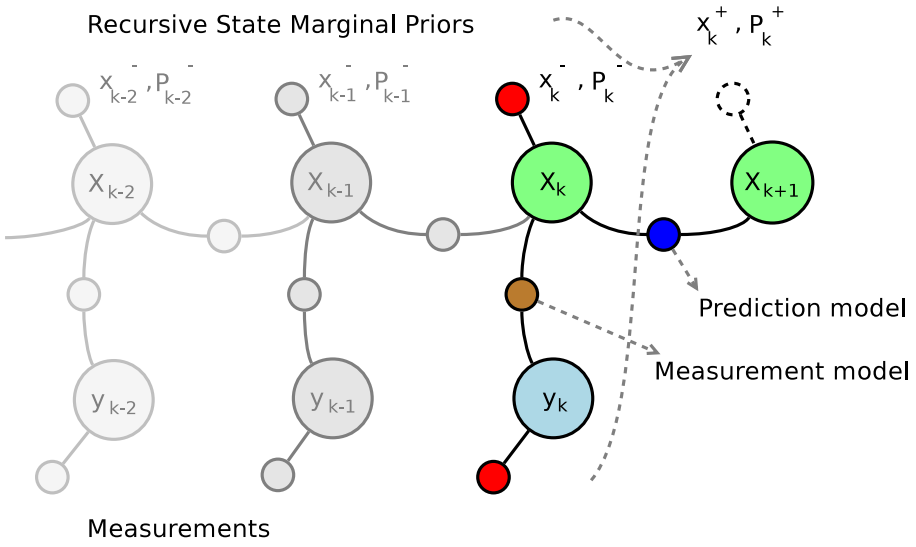


Figure 2-2: Factor graph based hidden Markov model interpretation of a Kalman filtering process. The previous posterior marginal state belief is $\left[{X_k}^+ \,|\, Y_k, Y_{k-1}, \ldots \right]$, and predicted next state is $\left[{X_{k+1}}^- \,|\, Y_k, Y_{k-1}, \ldots \right]$. Measurement observations y_k are related to the state x_k through the measurement model. Under the all Gaussian measurement and prediction model assumption (Kalman filter), only the covariance P_k is needed to fully describe the belief states.

part developed from the understanding of recursive filtering with Gaussian variables, as discovered by Kalman [118].

Zarchan and Musoff [244] present linear, polynomial, continuous, and extended Kalman filtering equations, and relate them to weighted least squares solutions. In particular Zarchan and Musoff (in later chapters) show how an augmented state vector can be updated in iterated Gauss-Newton update steps to simultaneously estimate the least squares solution to all variables of interest.

A major advantage of Kalman filter systems is that predictable computation times can be used to design and develop real-time data fusion or signal tracking system. This predictable nature of Kalman filtering has had a major influence in adoption of the filter, but also prevented designers from adopting a different kind of inference system.

The sigma point—or unscented—Kalman filter proposed by Julier & Uhlmann [110] uses the sigma point transform [109] to better propagate the measurement noise uncertainty $\mathbf{P}_{(k+1)}{}^-$ through a nonlinear transform $\mathbf{f}\left(\mathbf{x}, \mathbf{u}\right)$. The transform propagates several sigma points from the current covariance $\mathbf{P}_{(k)}{}^+$ through the

nonlinear transform $\mathbf{f}(\cdot)$ and refits a new Gaussian belief to the sigma points to estimate the new priori covariance estimate $\mathbf{P}_{(k+1)}{}^{-}$, Van Der Merwe et al. [232]. All states are still assumed to be normally distributed.

Taking the Bayesian view of a forward pass on the HMM, the prediction of stochastic state variables $\mathbf{X}_{(k)}$ to time step $\mathbf{X}_{(k+1)}$ corresponds to a convolution with a conditional proposal distribution $\left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{X}_{(k)}, \mathbf{U}_{(k)} \,\right]$,

$$\left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{U}_{(1,..,k+1)}, \mathbf{z}_{(1,..,k)} \,\right] \propto \int_{\mathcal{X}} \left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{X}_{(k)}, \mathbf{U}_{(k)} \,\right] \times \left[\, \mathbf{X}_{(k)} \,|\, \mathbf{U}_{(1,..,k)}, \mathbf{z}_{(k)} \,\right] \, d\mathbf{x}_{(k)}, \tag{2.3}$$

with model input $\mathbf{U}_{(k)}$ and previous marginal belief $\left[\, \mathbf{X}_{(k)} \,|\, \mathbf{U}_{(1,..,k)}, \mathbf{z}_{(k)} \,\right]$. The conditional belief is the probabilistic likelihood understanding of the state transition model $\mathbf{f}(\cdot)$, which is commonly calculated as the time integral from a dynamics model

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}, \mathbf{U}). \tag{2.4}$$

The previous state $\mathbf{X}_{(k)}$ is marginalized out by integrating over the entire alphabet $\mathbf{x} \in \mathcal{X}$. Note, the measurement $\mathbf{z} = \ominus\hat{\mathbf{y}} \oplus \tilde{\mathbf{y}}$ represents the on-manifold residual between model predicted and sensor measured observation states $\mathbf{y}$. Also note the shorthand, as per Gelfand et al. [71], where probability densities are represented with square braces, $p(\cdot) = [\,\cdot\,]$.

The predicted state $\mathbf{X}_{(k+1)}{}^{-} = \left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{z}_{(1,..,k)} \,\right]$ is combined with observational data through the product between infinite belief functions

$$\begin{aligned} \left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{z}_{(1,..,k+1)}, \cdot \,\right] &= \frac{\left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{z}_{(1,..,k)} \,\right] \left[\, \mathbf{z}_{(k+1)} \,|\, \mathbf{X}_{(k+1)} \,\right]}{\left[\, \mathbf{z}_{(1,..,k+1)} \,|\, \mathbf{z}_{(k)} \,\right]} \\ &\propto \left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{z}_{(1,..,k)} \,\right] \left[\, \mathbf{z}_{(k+1)} \,|\, \mathbf{X}_{(k+1)} \,\right], \end{aligned} \tag{2.5}$$

where consecutive measurements are assumed independent, i.e. $\left[\, \mathbf{z}_{(1,..,k+1)} \,|\, \mathbf{z}_{(k)} \,\right]$ is uniformly distributed. Furthermore, the input vector $\mathbf{u}$ is generalized to a "measurement" to our predictive transition model, and wrap the associated variable in as part of the measurement vector $\mathbf{z}_k$. The measurement likelihood model $\left[\, \mathbf{z}_{(k+1)} \,|\, \mathbf{X}_{(k+1)} \,\right]$ implicitly defines the stochastic variable $\mathbf{Y}_{(k)} = \mathbf{y}_{(k)}$, when given a measurement $\mathbf{z}_{(k)}$.

Work by Dellaert et al. [41] in 2005 shows that the Kalman filter is a computationally poor choice for SLAM style inference, and closer investigation shows that

graphical models present a more efficient approach to describing the navigation inference problem.

2.4.2 Particle Filtering (HMM)

Many different measurements are useful to inferring a navigation state estimate, but may have highly non-Gaussian measurement uncertainties. Data association ambiguities (multi-modal errors), topographic derived position likelihood regions, or correlator outputs from acoustic or radio beam formed processing with USBL systems are clear examples. Many navigation systems, which fundamentally rely on such non-Gaussian measurement likelihoods—such as Rypkema et al. [163]—have opted for particle filtering for approximating belief over the state variables of interest.

Particle filtering implements the Bayesian forward recursive solution to the hidden Markov model, as presented by eqs. (2.3) and (2.5) and shown in Fig. 2-2. As the name suggests, particle filtering uses "particles", or samples, from the estimated marginal belief over state vector variables. The histogram of particles represent the belief estimate over the variable of interest, reduces computational complexity down from infinite belief functions by using a N number of samples. For example, consider the Monte Carlo approximation of the expectation operation given the marginal over some state estimate $\left[\, \mathbf{X}_{(k)} \,|\, \mathbf{z}_{(1,..,k)} \,\right]$:

$$\begin{aligned} \mu_{\mathbf{X}} &= \mathbb{E}_{\mathbf{X}_{(k)}|\mathbf{z}_{(1,..,k)}} \left[h\left(\mathbf{X}\right) \right] \\ &\approx \sum_{i=1}^{N} \mathbf{h}\left(x_{i,(k)}\right), \qquad\qquad x \sim \left[\, \mathbf{X}_{(k)} \,|\, \mathbf{z}_{(1,..,k)} \,\right]. \end{aligned} \tag{2.6}$$

The prediction step or time propagation step, as shown in eq. (2.3), is achieved by propagating forward each particle through the prediction model, given in eq. (2.4). This step approximates the convolution of the current best estimate belief with the prediction model (conditional likelihood), $\left[\, \mathbf{X}_{(k)} \,|\, \mathbf{z}_{(1,..,k)} \,\right] \left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{X}_{(k)} \,\right]$.

Critically, what makes the particle filter, is the product operation between the measurement likelihood belief and state predictions which is done with importance weighting and resampling [224]. Importance sampling is algorithmically simple but suffers several detrimental weakness as a result, Doucet et al. [45]. Taking the measurement model as function $\mathbf{h}\left(\mathbf{X}\right)$, as shown in eq. (2.5), the product

between two infinite functions over the alphabet $\mathcal{X}$ can be taken as

$$\begin{aligned} &\left[\, \mathbf{z}_{(k+1)} \,|\, \mathbf{X}_{(k+1)} \,\right] \times \left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{z}_{(1,..,k)} \,\right] \\ &= \mathbf{h}\left(\mathbf{X}\right) \times \left[\, \mathbf{X}_{(k+1)} \,|\, \mathbf{z}_{(1,..,k)} \,\right]. \end{aligned} \tag{2.7}$$

This concludes convolution and product approximations used for a recursive particle filtering framework that implements the hidden Markov model shown in Fig. 2-2.

Representing the belief over variables of interest allow the user to track uncertainty over variables of interest. The great benefit lies in the ability to track non-Gaussian, possibly multi-modal, belief for each variable. Particle filtering by itself, however, has two significant disadvantages. Firstly, weighted particles with importance weighting assumes that particles will be available in all parts of the domain (alphabet) $\mathcal{X}$ which are of interest. The influence of measurement functions, using the Monte Carlo approximation shown in eq. (2.6), can only be supported if that part of the domain is represented by particles.

Furthermore, when repeatedly weighting many particles by measurement functions, many particles end up have very small weights. The number evenly weighted particles can be recovered through a technique called resampling [224], but the approach drastically reduces the regions of the state space with support. The problem is exponentially exacerbated as the dimension of state vector $\mathbf{X}$ is increased. This leads to the common particle depletion problem for higher dimensions.

The second disadvantage is a consequence of the hidden Markov model approach, and involves complexity associated with inferring the posterior belief over higher dimension state variables. Although the problem is also endemic to Kalman filtering approaches, the complexity is more obviously visible with particle filtering. Consider augmented state systems, such as trying to estimate the previous pose positions and observed landmark positions in a simultaneous localization and mapping setting, see for example FastSLAM [226].

The high dimensional augmented state variable $\mathbf{X}$, when updated in a hidden Markov (filtering) framework, becomes prohibitively difficult to compute. Kaess et al. [115] and others show that exploiting the structure within the problem can significantly reduce computational cost, over a simplified hidden Markov model approach. Classically these structure exploiting approaches have focused on parametric optimization.

2.4.3 Transitioning to Optimization Approaches

The simplicity of filtering approaches also pose some problems, namely the complexity associated with estimating posterior distribution of high dimensional state variables (especially non-Gaussian estimates), and a rigid design philosophy which limits the flexibility in combining various measurements of opportunity. One clear example of the rigidity of filtering solutions is the MSC-KF, visual-inertial navigation algorithm [158]: Introduce wheel odometry or leg kinematic type measurements into the state estimate will require a near total re-engineering to develop a new Kalman filter solution with the required measurement update cycles. Furthermore, Mourikis et al. [159], proposed a dual-layer approach to include loop closures into the MSC-KF, but resulted in a complicated loop whereby old measurement information is reused multiple times and thereby solving an inconsistent problem.

In contrast to an Apollo style filtering state estimate solution, robotic localization and mapping systems developed with heavy emphasis on wheeled odometry, visual odometry [96], and loop closures [105]. Other modalities still have tried to use lidar or underwater sonar [196], along with a motion model of how the robot might be moving, or mapping the RMS Titanic from camera image data [53]. Even more techniques use model priors to aid the mapping and localization process [183]. These highly varied systems, and lack of GPS solutions in in-door settings, has resulted in the development of simultaneous localization and mapping methods.

2.5 Parametric Simultaneous Localization & Mapping

The navigation systems discussed above all use the a hidden Markov model—predominantly the Kalman filter—to combine inertial prediction steps with state measurement updates for aiding information sources. We will now look at methods developed by the robotics community to navigate robots in GPS-denied environments.

2.5.1 Miniaturizing Digital Computers (Least Squares)

Over many centuries, cartographers had developed many localization and mapping techniques but the advent of miniaturized computers had started a new automated direction in navigation. During the 1980's, Smith, Cheeseman [210] and

others help start the discussion on automating a localization and mapping framework. Their starting point — without human involvement — was that a robot would be switched on and have to navigate and map the world from nothing but the available (onboard) sensors, software, and computational resources. Their work [211, 212] was able to identify that either the mapping or localization could be done with some stochastic representation.

Leonard et al. [135] and others — in early the 1990's — transitioned towards simultaneous localization and mapping (SLAM), which could be achieved by writing all sensor measurements in a relative reference frame. For example, instead of writing the navigation solution as the result of absolute measurements from ranging or bearing measurements to know landmarks, all position variables are kept fluid and measurements are only defined between them. Lu and Milios [141], in 1997, introduced a common notation based on Lie Groups and Algebra for a two dimensional simultaneous localization and mapping framework.

The ability to perform both localization and mapping, by analogy, allows the navigation system "to pick itself up by its shoestrings". Similarly in the early 1800's, Gauss had formulated a system of equations in a relative coordinate frame to estimate the positions and distances between objects from bearing only geographic surveys, and is embodied on the ten Deutsche Mark banknote shown in Fig. 2-3. Gauss developed a least squares solution, assuming normally distributed measurement measurement errors, thereby estimating an multivariate normal posterior distribution to represent the uncertainty in position estimates.

Figure 2-3: Deutsche Ten Mark showing a relative coordinate frame optimization problem, bottom right, in honor of Gauss for mapping and localizing positions from bearing only measurements.

During the same period, starting with the Apollo lunar missions, techniques such as bundle adjustment (BA) and structure from motion (SfM) had been developed to build three dimensional reconstructions of from photographic data, as shown by the work of Brown in the mid 1970's [26]. A modern look at Bundle Adjustment and Structure from Motion techniques was published by Triggs et al. [231] in 2000, and heavily focused on parametric optimization techniques to perform inference on large scale photometric data. These techniques use Gaussian parametric representations with a Quasi-Newton iterative optimization process to infer maximum likelihood estimates (MLE) for variable assignments. Section 2.6 discusses parametric optimization in the context of factor graphs in more detail.

2.5.2 EKF-SLAM

The discussion above shows how our understanding develop from using absolution navigation measurements, from a GPS for example, had to evolve into a relative frame representation. With a relative frame representation, we can write down how wheel odometry develops a rigid transform between two pose locations as a robot is traveling through the world. The next question was how to use relative frame mechanizations to develop an online state estimate in the location of the robot, with the additional by product of a local map of the surrounding environment.

Following success from INS/GPS or INS/DVL Kalman filtering solutions for multi-sensor fusion processes, a natural step was to use an extended Kalman filter to infer state estimates for simultaneous localization and mapping, [212]. Many authors, including Leonard et al. [135] , Bailey [10] and Durrant-Whyte [11, 48], resorted to an augmented state vector EKF approach. Previous pose positions and landmark locations in some world reference frame would be stacked into a large augmented state vector $\Theta = \begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2, ..., \mathbf{L}_1, ...\end{bmatrix}$.

A trivial EKF transition model is used as all variables are assumed static in the world frame, but a large measurement function would be used to produce residuals $\mathbf{z}_{(k)}$ between predicted $\hat{\mathbf{y}}_{(k)} = \mathbf{h}\left(\Theta_{(k)}\right)$ and actually observed measurements $\tilde{\mathbf{y}}_{(k)}$

$$\delta_{(k)} = \mathbf{z}_{(k)} = \tilde{\mathbf{y}}_{(k)} - \mathbf{h}\left(\Theta_{(k)}\right). \tag{2.8}$$

Note, the problem time step counter $\cdot_{(k)}$. With measurement residuals in hand, a Kalman gain $\mathbf{K}$ can be used to update the state vector estimate with usual Kalman

filter update equation:

$$\Theta_{(k)}{}^{+} \leftarrow \Theta_{(k)}{}^{-} + \mathbf{K}\,\mathbf{z}_{(k)} \tag{2.9}$$

The major cost of the EKF-SLAM algorithm is in inverting the high dimension state covariance matrix $\mathbf{P}$ for each Kalman measurement update step. Similar to the MSC-KF, the computational cost could be reduced by Rao-Blackwellized landmark states – that is removing landmark variables from the Kalman filter state, and iteratively optimizing landmarks relative to current pose states. Pose states are then resolved, through Kalman filter update models, according to the latest landmark state estimates. Double counting of information can be avoided with a null-projection, as done by Mourikis et al. [158].

2.5.3 Sparse Information Filters

The next major development in SLAM came in the early 2000's with the realization that inverting the state covariance matrix $\mathbf{P}$ would result in a sparse information matrix, $\mathbf{P}^{-1}$. Methods, such as Thrun et al. [225] or Eustice et al. [52], developed SLAM based algorithms directly in the inverted information space, greatly reducing the cubic computational cost of inverting the covariance matrix at each step.

The problem with information state filters is difficulty in tracking which terms of the information matrix should be kept, and which should be discarded since the inherent structure of the matrix was not yet obvious. New developments would soon show that the hidden Markov model assumption was an over simplification, and that far better computational performance could be achieved while still computing the exact solution for a Gaussian distributed system.

2.5.4 Monte Carlo Localization

Another facet to HMM-type localization and mapping approaches was inspired by particle filtering methods. Monte Carlo localization, by Dellaert et al. [43], estimates the posterior belief of the latest pose and landmark states using a particle filter approach. The approach allows non-Gaussian beliefs to be used for measurement and prediction update cycles to the state vector. These approaches suffered from particle depletion in high dimensions, and therefore maintained relatively small augmented state vectors.

To improve on depletion, the FastSLAM [226] approach is conceptually similar by emphasizing a subset of Gaussian parametric solutions. FastSLAM is a

midway between belief methods and multi-hypothesis parametric solutions. In FastSLAM, a set of particles are used to approximate the belief of all possible trajectories, where each particle represents an entire trajectory.

All trajectory poses and landmark positions are stacked in one vector, and each vector represents one particle. Each vector is then augmented with new pose information, using the odometry model, to update the entire system. Multi-hypothesis approaches therefore would require exponentially many particles to explore the entire space of possible solutions. FastSLAM offloads the problem of hypothesis selection to the user, and internally loses hypotheses at the resampling step. Lost modes cannot be recovered, and represents a fundamental difference in the method for multi-modal inference.

Furthermore, the augmented state variables can be Rao-Blackwellized, such that landmarks are separated out and estimated in an iterative fashion alongside the pose state variables. For example, under correct data association, a single particle can be used to recover the full SLAM solution. However, FastSLAM does not exploit structure within the joint probability distribution, unlike the Bayes tree [115]. Critically, the Bayes tree precisely encodes the type of structure needed for multi-hypothesis tracking, but this has not previously been studied in detail.

While the FastSLAM approach is able to track non-Gaussian belief, each variation across the entire trajectory and landmark positions requires a different particle. In turn, exponentially many particles are needed to encompass all the variable dimensions for an accurate posterior estimate – this is prohibitively inefficient. Since belief complexity scales exponentially in the dimension of a particle, one would rather use more particles who each represent a much smaller state vector. Our approach will instead work on a particles per marginal belief basis, where marginal belief dimension is kept small.

2.6 The Factor Graph Representation

Probabilistic inference and high dimensional nonlinear optimization are closely related through Bayes rule by using Gaussian distributed measurement likelihood models. The factor graph interpretation, Kschischang [129], makes this common understanding possible. Chapters 3 and 5 discuss how nonparametric factor graphs can be assembled, and how non-Gaussian, multi-modal state estimates are inferred.

During the early 2000's, along with new understanding of using information rather than covariance representations, graph based description of SLAM became

started becoming more popular, for example Bailey [12]. The pivotal factor graph representation was described by Kschischang et al. [129] in 2001. Dellaert et al. [41,42] connected the square root information matrix (of previous EKF-SLAM systems) to the factor graph representation.

Factor graphs provide a tractable language (unifying perspective) for large scale, non-linear, and belief space interactions of many variables and factors (likelihood models). The interaction of information from various sensors is naturally described by adding the associated measurement likelihood model (algebraic functions) between variable nodes as a graph. Furthermore, factor graphs can then be used to develop the associated inference algorithms to produce the desired state estimates.

Dellaert et al. [41] illustrates how to write measurement likelihoods as a factor graph model, and how to employ known linear systems solvers, such as [6], to recover mean parameter estimates. These tools include pivoted Cholesky or QR factorization for quasi Newton type optimization routines, and we direct the reader to a thorough development by Rosen et al. [198] which describes how to use trust-region methods to overcome many of the numerical problems of quasi-Newton methods.

In particular, a Gaussian factor graph solution when represented as the hidden Markov type model in Fig. 2-2 produces the same parameter estimate result as a Kalman filter. Solving over all variables produces a smoothing result, while solving and marginalizing forward along the HMM produces exactly the same set of equations used for the Kalman filter.

Consider, for example, a robot exploring a building room to room, and that new information from a later room has little impact on the geometry of a previously visited room. Kaess & Dellaert et al. [111] realized that a full batch solution over all variables does not have to be computed at each step, allowing for incremental quasi-Newton routines and leading to the iSAM1 algorithm [116]. The first key insight to incremental updates was to keep a triangular decomposition of the square root information matrix, new variables would add columns and measurements would add rows. The triangular nature of the augmented information matrix could be restored with Givens rotations (or Householder reflections) to modify only a small portion of the matrix and thereby recycling previous computations.

Maintaining the triangular component of the information matrix still had one lingering difficulty: linearization. The components in the matrix represent linearizations from the nonlinear measurement functions and would periodically be updated as state estimates changed. Further research into using graphical structures similar to the elimination tree would result in development of the Bayes tree.

2.6.1 The Bayes Tree

During the late 1980's, before the definition of factor graphs, Pearl's seminal book [187] introduced Bayesian networks for graphical models in statistical inference. A Bayesian network is topographically similar to the factor graph, as it is assembled by eliminating one variable at a time from the graph. The Bayes network encodes the conditional dependence structure and establishes the chordal property through implicit cliques. In turn, cliques can be discovered with the maximum cardinality search algorithm from Tarjan et al. [222], producing an acyclic graphical model description, also known as a Junction tree.

Many methods have been developed to find an acyclic refactoring of variables in a cyclic graphical model. Koller et al. [127] point to cluster trees, rake-and-compress trees and bucket elimination trees. Alongside junction trees, the so called elimination tree from sparse linear algebra all basically represent the same desire of finding the acyclic conditional dependency structure between the variables. Many different tree factorizations are possible, where the best trees have many cliques with small dimension each. While elimination strategies vary for each of the trees mentioned, the variable ordering in which the tree is assembled is a common and vital step to finding a good Bayes network and tree.

The Bayes tree by Kaess et al. [114] is a specific form of the Junction tree where a root clique is carefully selected, and where the conditional dependence structure is directed from the root to the leaves cliques. The Bayes tree is assembled with a variable ordering obtained with the column approximate minimum degree ordering (COLAMD) algorithm developed by Davis et al. [39] published in 2004. Empirical study shows the variable ordering obtained from COLAMD, while being a heuristic method, is within a few percent of the optimal ordering for a wide variety of cases. The COLAMD for variable ordering is the current best known method for finding the acyclic Bayes tree.

Algebraic operations and assembly of the Bayes tree is discussed in Section 5.3.1 as part of our development and defer the reader there for more detail. In broad terms, small, local pieces of the factor graph are grouped into cliques, in such a manner that an acyclic tree structure is formed. The Bayes tree represents a symbolic refactoring of the original factor graph model, from where an inference algorithm can consider operations at cliques level. A solution is found by "combining" information from the outermost leaf cliques up towards the one root clique. Once a solution has been found at the root, the combined information is sent back down the tree towards the leaves to recover the full posterior state estimates.

In the case of a singular Gaussian system, as used by the iSAM2 algorithm

from Kaess et al. [115], dense linear matrix operations can be used to locally solve portions of the information matrix. The entire Bayes tree represents how different dense portions of the triangular factor of the square root information factor is put together. The advantage of this approach approximations and estimates are local to measurement functions withing each clique rather than in batch over the entire information matrix. In iSAM2, this allows relinearization of measurement functions only in cliques that see larger shifts in parametric state estimates.

Furthermore, incremental updates to the entire system becomes more natural on the Bayes tree structure. By forcing the most likely affected variables to be near the root of the Bayes tree, large parts of all the branches remain unaffected as new variables and factors are added to the factor graph. In turn, these unaffected branches can be "unhooked" from the tree and be reattached after the root portion is re-eliminated with updated factor graph. Inference information only needs to passed up from the reattach point in the new tree, and downward passing only needs to propagate as far as meaningful updates to the states estimates are made – known as the wildfire algorithm.

Interestingly, Paskin et al. [185] had, in 2003, suggested a "thin" junction tree approach to a HMM style filtering solution to EKF-SLAM. The approach also kept the dimension of the problem small by marginalizing out old states to achieve a fixed lag smoother type of operations. The major difference was that Paskin's approach did not have the COLAMD algorithm to select a variable ordering from a factor graph definition.

The Bayes tree has a broader interpretation than a singular and pure Gaussian model which has been used in iSAM2, and continue the discussion of more general stochastic inference on the Bayes tree in Section 2.8.

2.6.2 Adding Inertial Measurements to Factor Graphs

As discussed earlier, inertial navigation plays a vital role in almost all autonomous system platforms in use today. SLAM research has lead us away from HMM-type inference algorithms such as the Kalman filter, favoring relative frame representations captured by a factor graph model. The introduction of inertial odometry factors for general use of inertial navigation-type measurements is a core competency for navigation and will be discussed in more detail in Chapter 4.

Indelman et al. [102] initially proposed to use individual inertial sensor measurements for odometry constraints in a factor graph, creating a new pose for each sensor measurement. However, the rapid increase in the number of poses makes it difficult to compute real-time solutions for large problems. Their later work [103]

adopted the *preintegrals* of Lupton [143], but again did not present an analytical version of the inertial sensor model they employed.

Martinelli [147] proposed a closed-from solution for visual-inertial odometry constraints, but only considered accelerometer biases during his derivation. While accelerometer bias certainly is an important error source, it is not the most significant. Platform misalignment, which is predominantly driven by gyroscope bias, results in erroneous coupling of gravity compensation terms. This gravity misalignment, when integrated, is a dominant error source in all inertial navigation systems [229].

Recently Leutenegger et al. [137] published work on a visual inertial SLAM solution, which does indeed estimate bias terms. Their work presents an excellent overview of visual inertial systems, but does not present complete analytical models for compensated interpose inertial constraints; their work does mention the need for compensation Jacobians, but are not presented.

Work by Forrester et al. [59] was conducted in parallel with this thesis work and similarly presents an exponential manifold type residual function for interpose constraints with retroactive sensor bias estimation, however, we are able to extend on their work. The interpose residual function, based on preintegrated inertial measurements, is currently unknown, and methods listed above use a linear, first degree approximation of the unknown compensation function.

2.7 Robust Optimization Methods

The sections that follow discuss work on robustness to try overcome many of the errors associated with non-Gaussian measurements being introduced as Gaussian measurement likelihood models. Several robustness efforts have focused identifying and removing ”bad” measurement factors from the factor graph. The assumption of bad factors implies incorrect measurements were made, either by bad data association or otherwise, and should be removed from the inference problem. Many methods in SLAM either avoid ”bad” measurements with highly-engineerined front-end processes, or preprocessing of an existing factor graph before actual variable inference is done. For example, Latif et al. [132] show the value of finding consensus at the front-end stage, delaying loop closure constraints until several new constraints agree and adding them to the factor graph as a batch of new constraints. This section takes a brief look at some of these methods.

2.7.1 Null-hypothesis Approaches

Switch variables were proposed in 2012 by Sunderhauf et al. [220] as binary slack variables into the optimization that can enable or disable each measurement. Measurements which are inconsistent with the rest of the graph are discarded through multiplication by zero. An additional variable, being one or zero, is added to that factor in the least square sum objective function, and introduced as part of the inference procedure. A user specified penalty is used to ensure some attempt is made to keep the factor active during the optimization process. Switch variables are comparable to a null-hypothesis approach [181], and has the disadvantage of ignoring information and relying heavily selecting the correct penalty values.

Further Olson et al. [181] points out that switch variables increase the number of variables and increases fill-in during inference which may result in notable computational performance loss. Furthermore, the null hypothesis approach may easily discard lonely, but valid, measurements and thereby ignore true data.

More recently, Graham [76] suggests using an expectation maximization (EM) approach to smoothly transition poorly matched measurements to assumed "outliers" by adjusting their measurement covariance. The EM algorithm is used to iterate between covariance weight selection and optimal variable assignments and thereby suppresses outlier-like measurements and emphasizing the majority of constraints which form consensus.

Moving closer to benefits offered by the symbolic structure of the Bayes tree, the hybrid continuous-discrete inference by Segal et al. [205,206] uses discrete states to enable or disable measurements, much like switch variables. The difference, however, is that the likelihood of enabling or disabling a factor is encapsulated by a discrete belief. By explicitly splitting posterior belief and optimal variable assignment computations, a best fit solution is found by searching for the posterior belief over all the discrete variable, in multiple passes over the Bayes tree, relative to the initialized state. Segal's work suggests an underlying synergy between ambiguity through belief and consensus amongst multiple hypotheses.

2.7.2 Max-Component Approach (*Max-product*)

Olson et al. [180,181] proposed the max-mixtures approach which selects the local maximal weighted Gaussian from a mixture of Gaussians before continuing with a parametric optimization routine. Their approach is akin to max-product inference, which greatly simplifies the inference problem by discarding all but the most likely hypothesis for each factor before inference, using a local *if* statement. Once each

factor has locally selected the most likely Gaussian hypothesis, a usual solution (as discussed above) is used.

2.7.3 Multi-hypothesis Approaches

Rather than retrofitting a single parametric solution with a null-hypothesis, another approach, as recently suggested by Huang et al. [97], suggest solving multiple parametric problems in parallel and then picking the most likely solution. The FastSLAM approach of Thrun et al. [226] (mentioned earlier) also maintains parallel trajectories, but keep all permutations in separate particles with very high dimension.

While multi-hypothesis approaches seem appealing, one should not underestimate the complexity associated with tracking all possible hypotheses in a system. Consider a case where several data association uncertainties, such as loop closures to objects, is to be deferred to the back-end solution. Following only the forward trajectory in time, each new binary decision introduces a doubling in the possible permutation of choices.

Therefore, if ten binary associations are to be tracked, there are 1024 possible hypotheses. FastSLAM had skirted this problem by requiring the front-end process to not request all hypotheses, assuming a local solution where only the most dominant modes in the current trajectory are being tracked, but the user must make the decision as to which modes are dominant. This behavior is a contrast to our proposed method where the back-end inference solution determines through consensus which modes are dominant, purely based on all uncertainty modeled in the factor graph.

Consider the factor graph view of parametric multi-hypothesis methods. Each possible solution permutation (hypothesis) is explicitly solved with a slightly different factor graph. If all possible permutations are tracked, the correct solution will be contained in one of the many available solutions. It is important to note that using a method like RANSAC on such a collection of solutions is not valid, since each solution represents a different problem and not a Monte Carlo style variation on the same problem which may express consensus.

Intuitively, the factor graph encodes a random field where the belief interpretation can capture all the uncertainty in the system. The Bayes tree is a data structure which precisely encodes the type of structure needed for multi-hypothesis tracking, by aggregating all information during the upward pass from leaves to root. Consensus should occur as information is combined from sibling cliques. Furthermore, the inference algorithm selected modes are passed back down the tree

with the full posterior result already available during the downward pass.

The matrix permanent method introduced by Atanasov et al. [8] for semantic based mapping is considered a multi-hypothesis approach where object recognition with semantic labels are used to help discard false loop closure proposals and help detect other possible loop closures. The matrix permanent computation is used to search across all possible hypotheses and extract the consensus set, including discrete labels on object semantics. The matrix permanent computation is equally expensive as considering all possible combinations and does not support incremental inference offered by the Bayes tree approach.

2.8 Stochastic Inference (Nonparametric Techniques)

This section discusses work on stochastic inference which was done outside the robotic or guidance and control communities. Techniques already developed in the statistical inference literature will be used in our algorithm development. The discussion presented in this section is by no means complete, but represents a bridging between two research communities.

2.8.1 Belief Propagation Methods

Beyond the robotics community, Cowell [37] discusses how Gaussian likelihoods, using the graphical model as guide, propagate information between variables on the graph. For example, Cowell shows the triples method passing three parametric numbers between variables and shows how a complete posterior belief for each variable can be constructed.

Loopy belief propagation by Ranganathan and Dellaert [194] transmit belief directly on the factor graph. Their method still assumes that all posterior distributions have a Gaussian distribution, but does show that methods other than linear algebra optimization are possible. Loopy belief propagation requires repeated iteration of message passing across the entire network which contains loops, in a seemingly random pattern, with little guarantee that an acceptable solution will emerge. In robotic navigation, loop closures introduce cycles in the factor graph.

The expectation propagation algorithm [152] sends summarizing statistics as messages on the factor graph structure. For example, the first two or three statistical moments of the posterior can be propagated across the network, rather than sending an estimate an entire density function. This option may not be well suited for drawing accurate metric solutions from multi-modal distributions, since two

modes imply two different solutions are possible. The first moment of a distribution is its mean, which would incorrectly collapse distinct possibilities into a single parametric value between the modes.

The work of Kuehnel [130] in 2004 investigated Bayesian sampling-type inference techniques with focus on posterior estimation for structure from motion. His work illustrates that non-parametric representations can be used for localization and mapping type approaches. Our work will pursue approximating the posterior distribution of variables and will avoid cyclic graph inference by propagating belief on the Bayes tree instead.

2.8.2 Belief Propagation on the Junction Tree

Given the analytic appeal of acyclic re-factorizations, and the desire for fully fledged posterior belief estimation for each variable in the system, we should expect to find existing techniques in literature that have explored this approach. Indeed the CHURCH algorithm, from the mid 1990's in statistical inference by Kjaerulff [125] perform belief propagation type inference on the junction tree. Previous work on belief propagation have mostly avoided using tree representations due to the increase of dimensionality in cliques. Advent of the Bayes tree, and improvements of technology help us address three existing difficulties.

The CHURCH algorithm was developed before the COLAMD algorithm [39] existed and suffered from poorer variable orderings for larger problem sizes. The improved variable ordering of COLAMD produces lower dimension cliques in the Bayes tree, which greatly reduces the exponential complexity for each clique.

The second problem area is working with nonparametric beliefs. The CHURCH algorithm was developed before the multi-scale Gibbs sampling algorithm from Sudderth & Ihler et al. [218]. Rather than trying to deal with how parametric models interact in a junction tree, as CHURCH requires, any continuous belief can be approximated with a kernel density estimate, and then use the generalized multi-scale Gibbs product as suggested by Sudderth and Ihler.

The last aspect is driven by technology and computational power. The CHURCH algorithm did not have access to greatly enhanced computational performance, or more powerful programming languages such as Julia [19] which we use almost exclusively in our development.

2.8.3 Markov Chain Monte Carlo (Sampling)

Markov Chain Monte Carlo (MCMC) estimation of posterior distributions has received considerable attention, dating back to work by Gibbs and Markov type processes in the early 1900's. These statistical inference methods were mostly used for distribution estimation in the physics community, to infer hidden parameters with non-Gaussian distribution types or higher levels of uncertainty. As we discussed earlier, parametric optimization-type systems are difficult to solve for nonsingular systems.

Metropolis & Hastings et al. [31] connected Komolgorov's criteria for reversibility with Markov chain processes with an expression called detailed balance. Detailed balance ensures that a reversible Markov chain has a stationary distribution, and that a balanced sampling scheme stochastically produce samples from the embedded stationary distribution. The hybrid (Hamiltonian) Monte Carlo scheme was devised during the 1980's by Duane [47] and further explained by [86,91,167] is the best know mechanism to conduct truly non-parametric posterior inference. Oh et al. [178] points out many of the MCMC [165] methods have been focused on unimodal probability densities, with difficulty exploring multi-modal posteriors.

The multi-modal problem complexity is dramatically inflated as the dimension of the posterior increases. Starting in the 1990's, many authors have proposed approaches, such as tempered sampling or multi-scale methods, to both improve speed of MCMC sampling and effectiveness in finding all modes. Latent variables introduce a further layer of complexity in the sampling process, but are fundamentally part of any SLAM solution and our approach must therefore be able to cater for this case. To this end, our approach will rely on consistency of likelihood measurement models for wide mode proposal.

Metropolis-Hastings sampling of multi-modal posteriors has proved exceedingly difficult, and several works by Langevin [7], Neal's tempered transitions [166] and a few others [58,74] have been proposing methods for circumvent the mode discovery problem. Further samplers, including importance sampling using the *mixture importance function* [178] and Stratified importance sampling have also been proposed. The Avramidis AISDE importance sampler, Wang-Landou sampler [131] or many of the Equi-Energy sampler adaptations, such as [13], are worth mentioning. These samplers all intend to improve multi-modal performance but still have limitations in discovering removed modes in the posterior distribution.

The usefulness of Gibbs sampling was popularized in 1984 with a image reconstruction algorithm from Geman and Geman [72], and shows that Gibbs sampling can be affectively used for higher dimensional problems. Gibbs sampling uses the

actual conditional beliefs, assuming they are available, as proposal distributions in the sampling process. This direct use of the user-defined conditional beliefs allows such likelihoods to introduce multiple modes into the proposal distributions.

The imputation method from Tanner and Wong [221] in 1987 further expanded the notion of Gibbs sampling to include approximation of the belief functions themselves using only the current states and user-defined models. Tanner and Wong were able to show the accurate recovery of a bi-modal posterior distribution and rapid convergence rates, although the probability masses were still relatively closely spaced. Their work showed that function approximation could be done using a Gibbs sampling, and was furthered by Gelfand et al. [71] and Celeux et al. [29] during the 1990's. All these methods showed several variations to the Gibbs-type sampling strategies produce high quality results.

A powerful generalization of the Gibbs sampling scheme was presented by Sudderth and Ihler et al. [217,218] in 2003 and 2010. Their approach uses a multi-scale Gibbs sampling scheme [101] to estimate the product between approximate and multi-modal belief functions. Coarser scales are used to explore the entire space between modes, which is then refined down to the actual belief in at the finest scale. In particular their approach is built around kernel density estimation for continuous belief approximations, see Silverman [208], and is a major part of our development. Please see a more detailed discussion in Section 5.5.1.

Other methods such as progressive Bayes by Schrempf et al. [204], or reproducing kernel Hilbert spaces presented by Smola, Song, Fukumizu, and Gretton [69,77,213,214] indicate active research in nonparametric inference methods. Work in kernel Hilbert spaces are promising, since much of the theoretical ground work has already been developed, see Nashed et al. [164] from 1991 as one example. As well as more recent work on developing samplers directly in the embedded feature space, see Beskos [17].

2.9 Varying Timescales

An objective assessment of SLAM solutions suggest that many applications do not necessarily require a high speed SLAM solution to achieve low latency navigation. Concurrent smoothing and filtering work by [117] makes deeper assessment on how fast a large SLAM solution needs to be, and finds a robust, elegant analytical method to separate the low latency estimate from the large data fusion (or SLAM) process. This line of reasoning allows us to investigate slower, but much more versatile data management strategies, such as a database-centric approach.

Large volumes of data and robust inference techniques require larger computation resources, such as multi-processor architectures, which in turn places large concurrency demands on the design of a robotic computational system.

2.10 Database driven SLAM

A last aspect is the technological development which may have a major impact on future robot navigation-type systems, and involves Database systems for achieving concurrency between multiple actors. Classically databases have been used as a common data store, however, it's usefulness extends to online transactional-type systems. This section briefly looks at some of the SLAM based systems which have utilized database technologies.

Previous work by Newman et al. [174] used a database and querying language, in the context of SLAM, as an aid to finding loop closures from robot data. Consider an even bigger dataset [38] where common locations are searched by working through large amounts of camera image data from a car driving multiple kilometers. The requirement for efficient storage and concurrent computation becomes critical as long-term, multi-vehicle solutions are considered [162].

Database systems have traditionally been used for storing large amounts of transactional data, in which tabulated data mostly adheres to a set relational structure. Table and relations are designed beforehand, so that an operational system can add new rows and manipulate values in existing columns.

Relational databases perform well for a variety of applications [149], but are limited when the connections between data are rapidly changing; a dominant feature in SLAM systems being used and developed in the robotics community today.

Recent work by Nelson et al. [170] strongly argues for database driven long-term storage and data retrieval in robotics. Such data accumulation systems show performance benefits when using a database for searching over temporal and spatial cues, or integrating the back-end navigation solution to a database server [34, 162].

However, past work has focused on using a relational database structure as the primary mechanism of data storage, together with SQL queries for data retrieval. The required Entity-Relationship model for designing such databases assumes a specific mechanism of mapping and localization acting on the raw data, a mechanism which might not satisfy the interface needs of a new technique [161]. Moreover, these database methods require validity checking for each new datum and are not able to integrate new data into the map online [162]. This method doesn't

scale because of the necessary re-incorporation of all data for each query of metric computation, thus precluding online augmentation of the map [161].

2.11 Critical Analysis

Localization and mapping strategies developed in recent years have focused on cameras or laser scanners as primary navigation sensors. Simultaneous localization and mapping (SLAM) has been extensively studied by the robotics community and is converging towards a factor graph representation [129] of robot localization and mapping measurements. The literature suggests a period of independent development between the Aerospace type guidance and control community having developed the deeply coupled INS/GPS system and MSC-KF visual-inertial algorithms, and the robotics communities having developed factor graph based inference for SLAM. A shortcoming in incorporating inertial measurements at high fidelity into a factor graph model is identified, while supporting retroactive sensor calibration.

A navigation and mapping framework that can operate over varying timescales is also not clearly defined in literature, and a future area where this thesis can contribute. Realistically, robust inference and high-bandwidth inertial odometry should be combind for real-time state estimation, while extracting aiding information from a variety of sensing modalities. There are several areas of literature where greater convergence between classic state estimation and SLAM communities are possible.

Table 2.1 shows current state-of-the-art methods using inertial information for localization. There are many complimentary aspects between the works. Many of the existing works focus on visual (monocular and/or stereo) aiding only.

Most SLAM systems today require the data association process to be completed in the robot front-end process. A few works have investigated *back-end* solver adaptations to deal with ”bad” factors. Our opinion is the probabilistic interpretation of outliers must be updated to fit reality better. We propose an incrementalized belief propagation algorithm which is capable of dealing with such multi-modal constraints, while maintaining as much of the parameterized unimodal simplifications as possible.

Table 2.1: Existing inertial navigation/localization techniques, using either factor graph (FG) or filtering (HMM) inference methodologies.

	Infern.	Aiding	IMU	biases $\delta(\cdot)$	nonparm.
Multi-modal iSAM	FG	Any	so(3)	2^{nd}, d/dt	x
Lupton et al. [143]	FG	Camera	Euler	numeric	
Indelman et al. [103]	FG	Any	so(3)	numeric	
Leute.. et al. [137]	FG	Camera	so(3)	numeric	
Forster et al. [59]	FG	Any	so(3)	1^{st}, Δt	
MSC-KF [90, 98, 158]	HMM	Camera	instant.		
Jones et al. [107]	HMM	Camera	instant.		
INS/GNSS [229]	HMM	GPS	instant.		
FastSLAM [226]	HMM	Laser	N/A		x

2.12 Conclusion

There is rapid growth navigation technology and algorithm development as the prospect of widespread robotics increases. More people today are actively working on state estimation type task, to enable technologies such as self driving cars or industrial and domestic robots. Literature suggests a strong synergy between belief interpretations and consensus among multiple hypotheses, but this has not previously been studied in detail. The advent of: (i.) the Bayes tree, with near optimal clique dimension size, outperforms previous ad-hoc tree construction methods; (ii.) a better understanding of Bayesian inference to advance factor graph inference to support highly non-Gaussian posterior distributions; and (iii.) increased computational performance, will allow us to pursue multi-modal solutions to navigation-type factor graphs. We believe a multi-modal approach to navigation-type factor graphs will contribute significantly to the shortcomings in our capabilities in robotic navigation, as discussed in Chapter 1. This thesis will combine combing many of the existing techniques in a novel way and arrive at, what we believe is the first, larger scale multi-modal inference solution for robotics and more general computation.

Chapter 3

Probabilistic Modeling

Chapters 1 and 2 identify several limitations in current robotic navigation strategies. This chapter provides an overview and introduces new parametric and nonparametric sensor measurement models, and discusses how a front-end process can construct more realistic factor graph representations of real-world measurement uncertainties. Chapter 4 details the high-bandwidth, real-time nature of a new inertial odometry measurement factor. Once a nonparametric factor graph is assembled, the *Multi-modal iSAM* algorithm—discussed in Chapter 5—performs the required inference task.

3.1 Introduction

In broad terms, a platform (robot) navigation system should be able to interpret measurements from multiple sensors and fuse them into a consistent statistical estimate of platform location and world states. States vary from application to application, but generally include positions, velocities, orientations of objects in the world, includding key points along the platform trajectory called poses. A chain of previous poses is then also used to represent a set of relative relations to landmark features in the world.

SLAM casts robot navigation as a statistical inference problem. To obtain estimates of state variables with measurements from non-ideal sensors, we need an inference system and language for describing practical measurement uncertainties. Rather than use only Gaussian error models, nonparametric representations can represent non-Gaussian and multi-modal belief densities.

This chapter discusses several new parametric and nonparametric factors that allows the front-end designer more flexibility in modeling real world measure-

ment uncertainties, and focus on the importance in defining a residual function when constructing conditional likelihoods. Our discussion will focus around four major factor groups. A humanoid robot navigation example is used to illustrate how a front-end process can assemble a factor graph from nonlinear residual functions and probabilistic densities. The importance of marginalizating discrete variables, as well as consistent modeling, is also emphasized.

3.2 Probabilistic Modeling

Fig. 3-1 illustrates a navigation-type factor graph—derived from [129]—as the graphical modeling language of choice to describe relations between state variables and sensor measurements. The figure shows a hypothetical factor graph depicting desirable sensor measurement likelihood models.

After a brief prelude, the four major factor groups are discussed in the following sections. Fig. 3-1 shows pure inertial odometry constraints, aided by forward kinematics from the legs and monocular visual feature sightings through the head mounted camera; a trusted loop closure feature sighting is also incorporated at pose P_n. The figure further illustrates a multi-modal alternative to loop closure sighting, while using semantic level information to aid the inference. Fig. 3-1 also illustrates future (possibly multi-modal) trajectory planning towards manipulating an object affordance.

A factor graph should be interpreted as the product between independent sensor measurement likelihood functions $\varphi(\cdot)$ and unary priors $\psi(\cdot)$, also known as potentials in Markov Random Field theory [42]. In the Bayesian sense, this product is the unnormalized joint posterior of all the variables and encodes a multivariate probability density, using a short-hand notation from Gelfand et al. [71]:

$$p(\Theta|\mathbf{Z}) = [\,\Theta\,|\,\mathbf{Z}\,] \propto \prod_i [\,\mathbf{Z}_i\,|\,\theta_i\,] \prod_j [\,\theta_j\,] \tag{3.1}$$

where $\Theta_k = \{X_1, X_2, ..., X_m\}$ is a high dimensional random variable containing all the state variables to be estimated, and all measurements $\mathbf{Z}_k = \{Z_1, Z_2, ..., Z_{m'}\}$. A necessary condition to make the factor graph a Markov random field[1] is that all factors only dependent on variables according to edges to variables. Note the shorthand for probability densities $p(\mathbf{Z}_i\,|\,\Theta_i) = [\,\mathbf{Z}_i\,|\,\Theta_i\,]$, involving a subset of variables Θ_i.

[1]The Markov property dictates that modifications to variable estimates are only dependent on the current state of the system and not any previous states.

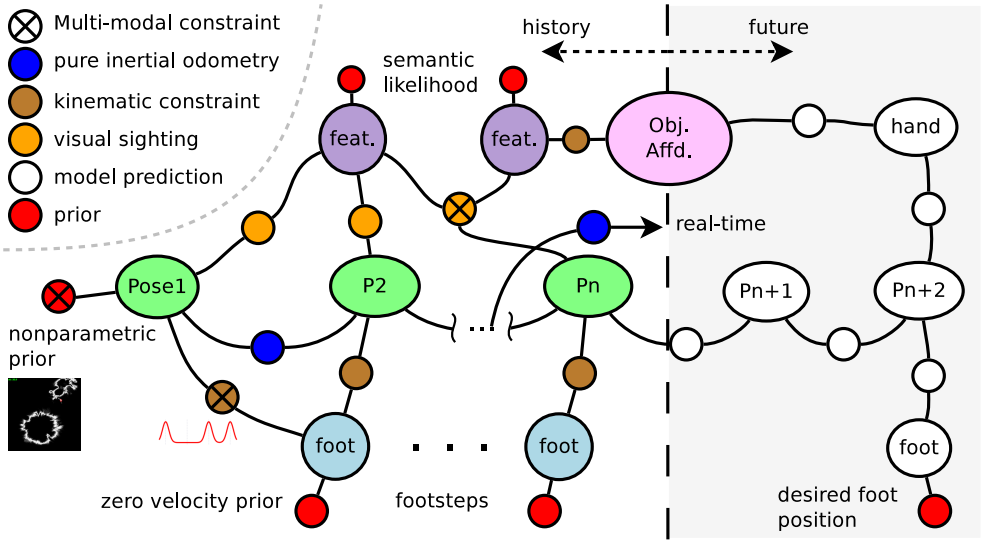


Figure 3-1: Illustration of a nonparametric, multi-modal joint probability density factor graph representing a humanoid robot navigating towards some object (as illustrated in Fig. 1-2). A front-end process is used to construct such a factor graph description. Figs. 3-2, 3-3, 3-7, and 3-11 discuss the four major factor groups of this figure in detail. Our posterior inference algorithm (back-end solution) is discussed in Chapter 5.

Regarding the size of the inference problem: A variety of reasons influence how many variables should be used for joint inference. Over time, as the platform (robot) moves around, more variables and measurements are collected and added to the joint probability product. In turn, the system size can be reduced by marginalizating out older variables. The incremental inference algorithm is optimized to efficiently recycle previous computations with the addition of new measurements and marginalization of old variables in the constantly changing joint distribution $\left[\Theta_{(k)} \,|\, \mathbf{Z}_k\right]$.

The complexity of inference is also affected by the types of probabilistic error density distributions that are used. Most existing SLAM solutions reduce complexity by using Gaussian noise assumptions for all measurements, implying one large multivariate Gaussian belief $\left[\Theta_{(k)} \,|\, \mathbf{Z}_{(k)}\right] = \mathcal{N}\left(\mu = {\theta_{(k)}}^*, \Sigma\right)$. By ${\theta_{(k)}}^*$ we mean the unimodal mean solution for Gaussian belief $\mathbb{E}_{\Theta_{(k)}|\mathbf{Z}_{(k)}}\left[\Theta_{(k)}\right]$.

While several navigation measurements can be assumed to fit a Gaussian uncertainty many, many situations do not, including incorrect loop closures, robot wheel slip, physical measurement ambiguity, topographic measurement, beam formed

acoustics/radio, or other nonparametric measurements that simply do not fit the Gaussian assumption. In addition, weakly constrained variables may have ambiguous solutions, producing highly non-Gaussian posterior beliefs. We will see an example of weakly constrained variables in Section 6.5. Our *Multi-modal iSAM* algorithm is a belief space approach, but does not exclude hybrid parametric and nonparametric approaches for performing inference, which we discuss in Section 5.5.5. This chapter aims to describe non-Gaussian and multi-modal measurement likelihood representations for general factor graph use.

Assuming a suitable inference procedures is available, we must also specify how to interpret the belief space inference result. The *Multi-modal iSAM* process, described in Chapter 5, does indeed provide a marginal posterior belief $\left[\Theta_i \,|\, \mathbf{Z}_{(k)} \right]$ over any system variable of interest Θ_i. At some point, however, we require a single point value estimate rather than full belief for variable θ_i. There are a variety of ways to obtain a single point estimate value. In the case of a strong unimodal posterior belief, we may choose to fit a normal distribution to the belief and extract the parametric values: mean $\mu_\Theta = \mathbb{E}\left[\Theta_i\right]$ and covariance $\mathbb{E}\left[(\mu_\Theta - \Theta_i)^T(\mu_\Theta - \Theta_i)\right]$. For multi-modal beliefs we may instead be interested in the maximum a posteriori (MAP) estimate

$$ {\theta_{(k)}}^* = \underset{\theta_{(k)} \in \Xi}{\operatorname{argmax}} \left[\Theta_{(k)} \,|\, \mathbf{Z}_{(k)} \right] \tag{3.2} $$

where ${\theta_{(k)}}^* \in \Xi$ represents the most likely point value assignment.

3.3 Importance of Residual Functions

The joint posterior belief is assembled from likelihood prior and conditional beliefs, see eq. (3.1). By residual function we mean a process or equation that equates to zero between a local system of variables, such as differencing a measurement and prediction. The conditional beliefs in particular are assumed to depend on some residual function which relates the associated subset of variables $\Theta_i \in \Xi_i$.

$$ \delta : \Xi_i \to \mathbb{R}^d. \tag{3.3} $$

The space Ξ_i is taken as the union of continuous and discrete domains on which θ_i may exist. Assuming real world events are approximated by an algebraic residual expression, we can develop a practical sensor measurement model $\mathbf{z}_i \sim [\, \mathbf{Z}_i \,|\, \Theta_i \,]$. The residual function is assumed locally differentiable, and the residual value ex-

ists in a regular real vector space of dimension d. In general, the on-manifold ($\oplus$ & $\ominus$) residual has vectorized prediction $\mathbf{h}$ subtracted from measurement $\mathbf{y}_k$

$$\delta_k = \ominus \mathbf{h}\left(\cdot\right) \oplus \mathbf{y}_k. \tag{3.4}$$

The residual function $\delta\left(\cdot\right)$—which is likely non-linear—is used in combination (or composed $\circ$) with a probability density function $p' \in \mathcal{P}$, from the space of all probability density functions $\mathcal{P}$, to construct the sensor likelihood model

$$\mathbf{z}_i \sim \left[\, \mathbf{Z}_i \,|\, \Theta_i \,\right] = p\left(\mathbf{Z}_i | \Theta_i\right) = \left(p' \circ \delta\right)\left(\Theta_i; \Lambda_i\right). \tag{3.5}$$

Where Λ_i represents additional parameters to complete the likelihood model. We note, all elements $\theta_i \in \Xi_i$ make out the alphabet of possible events for the measurement likelihood model $p \in \mathcal{P}$. The measurement $\mathbf{z}_i \in \mathcal{Z}_i$ represents a sample from the space of possible measurement residuals $\mathcal{Z}_i$.

We call eq. (3.5) the generic measurement likelihood potential and note that all measurement potentials discussed in this work are based on this equation. For example, a simple linear, normally distributed (on-manifold) conditional density might be defined as

$$z \sim \left[\, Z \,|\, X, Y \,\right] = \mathcal{N}\left(\mu = \delta\left(x, y\right), \sigma\right), \qquad \delta\left(x, y\right) = -y + 2x + 3. \tag{3.6}$$

We note the algebraic equivalence to $\tilde{y} = y + \delta = 2x + 3 + \eta, \;\; \eta \sim \mathcal{N}\left(0, \sigma\right)$.

The remainder of this chapter is devoted to illustrating how different residual functions are composed into measurement likelihood functions, and ultimately how the collections of measurement likelihoods in a factor graph (Fig. 3-1) describe the robot navigation problem as sketched for a humanoid robot in Fig. 1-2.

3.4 Navigation-type Factor Graphs

This section discusses several parametric and nonparametric factor types that a user front-end process could use to assemble a navigation-type factor graph. The discussion builds around a step-by-step example of a humanoid robot would navigate up to a door and grabbing the handle to open, as illustrated in Fig. 1-2.

The humanoid example (Fig. 1-2 from Chapter 1) illustrates four major groups of factors used to model the robot navigation task in factor graph form: (i.) multi-sensor data fusion; (ii.) high-bandwith and real-time state estimation; (iii.) multi-modality and nonparametric measurement likelihoods; and (iv.) future trajectory

planning tightly coupled to the state estimation process.

Fig. 3-1 illustrates a factor graph representation which allows the state estimate accuracy to fluidly improve as the humanoid robot approaches the door. The figure shows how all opportunistic measurements such as leg kinematics, inertial odometry, and visual sightings—including loop closures—can interact in a common framework. Furthermore, future motion planning can be included by enforcing vehicle model predictions up to some desired goal. For example, a future footstep location or how the robot hand should be placed on a virtual object representation (affordance[2]).

In Fig. 3-1, the green nodes represent robot pose state variables in the world frame ${}^{w}(\cdot)$, such as vehicle position ${}^{w}\mathbf{p}$, orientation ${}^{w}_{b_i}\mathring{\mathbf{q}}$, velocity ${}^{w}\mathbf{v}$, and possibly sensor calibration parameters such as IMU bias ${}^{b_i}\mathbf{b}$:

$$ {}^{w}\mathbf{x}_i = \begin{bmatrix} {}^{w}_{b_i}\mathring{\mathbf{q}} & {}^{b_i}\mathbf{b}_{\omega} & {}^{w}\mathbf{v} & {}^{w}\mathbf{p} & {}^{b_i}\mathbf{b}_{a}. \end{bmatrix}^{T} \tag{3.7} $$

The factor graph illustration shows several poses and many footsteps ${}^{w}\mathbf{fs}_k$, but is collapsed between pose nodes P_2 and P_n. Footsteps

$$ {}^{w}\mathbf{fs}_k = \begin{bmatrix} {}^{w}_{f_k}\mathring{\mathbf{q}} & {}^{w}\mathbf{v} & {}^{w}\mathbf{p} \end{bmatrix}^{T}, \tag{3.8} $$

accompany all body poses as the robot is walking or standing. Separate landmark variables ${}^{w}\mathbf{l}_i$ require only a position estimate in the world frame

$$ {}^{w}\mathbf{l}_i = \begin{bmatrix} {}^{w}\mathbf{p} \end{bmatrix}^{T}. \tag{3.9} $$

Visual feature sightings from a camera are shown as mustard color factors, and brown factors represent joint encoder derived kinematic factors between the robot pelvis and feet. Footstep locations are estimated as the light blue nodes with zero velocity priors, leaving the footstep locations and orientations to be estimated from as many leg measurements as possible. Inertial odometry constraints are also shown for one inertial measurement unit (IMU) between pose locations. We note many more factors, such as vehicle model dynamics, or multiple IMUs, or cameras can be included in a similar manner.

The next sections present an overview of parametric and nonparametric factors in our approach to achieving fluid location awareness for a robotic system. Factor graphs to combine the wide variety of parametric and nonparametric measurement likelihoods factor into a common graph representation. The inference

[2]Affordances are discussed in Section 3.4.5.

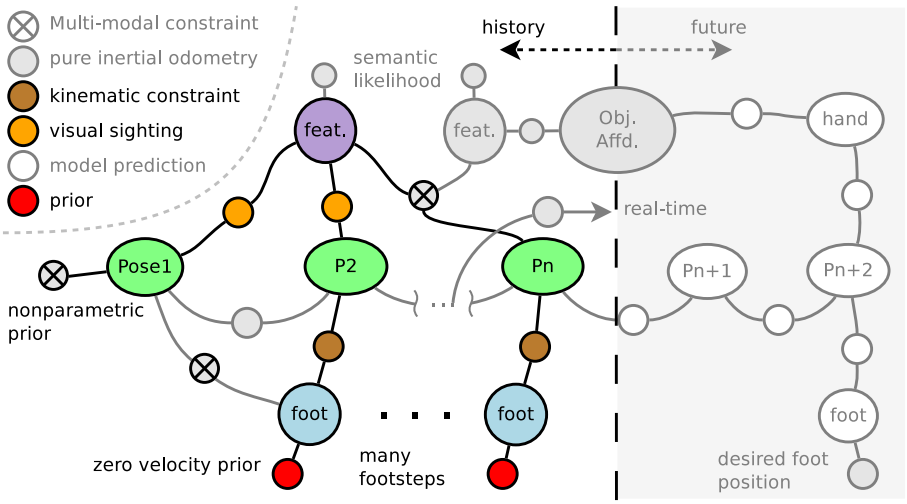


Figure 3-2: Group I: Shaded factor graph illustration from Fig. 3-1 of a humanoid robot state estimation process, multi-sensor data fusion.

process solves the navigation problem by finding approximations to the posterior belief represented by the factor graph.

3.4.1 Group I: Multi-sensor Data Fusion (SLAM)

We group general parametric models as the first of four factor groups, and are most similar to existing SLAM factors. We stress, however, the likelihood prior and conditional models can consist of distributions from which we can draw samples, and are no longer restricted to a Gaussian only assumption. Fig. 3-2 illustrates a portion of our humanoid example factor graph (from Fig. 3-1) where sensor data is collected into a common multi-sensor data fusion framework through a variety of measurement likelihood factors. The next factor group discussion discusses ambiguous measurements by allowing multiple parametric values to "coexist" in the same factor.

Prior: Regular Parametric Prior

Consider the body pose positions connected through kinematic relations by joint encoders in the legs from body to foot. Fig. 3-2 illustrates taking multiple measurements to multiple footsteps in the recent history. Assuming the feet do not slip, we can constrain each footstep $\mathbf{X}_{j,vel}$ with a zero velocity prior, represented

by a red unary factor. We note this is a partial factor which only constrains the velocity terms in the foot step variables. For illustration, assume the prior density $[\,\mathbf{X}_j\,]$ is Gaussian (but could be any density which can generate samples):

$$\mathbf{x}_{j,vel} \sim [\,\mathbf{X}_{j,vel}\,] = \mathcal{N}\left(\mu = \mathbf{0}, \Sigma\right). \tag{3.10}$$

Conditional: Range and Bearing

A different sensing modality may include a relative measurement between two variables, similarly illustrated by the visual sighting factor in Fig. 3-2. Consider laser scanner measurements taking range and bearing between the sensor ${}^{w}\mathbf{X}_j$ and feature ${}^{w}\mathbf{L}_k$ positions in the world frame. The conditional belief is defined with the appropriate on-manifold residual function:

$$\delta = \begin{bmatrix} D - \|\Delta\mathbf{l}_{pos}\| \\ \psi - atan2\left(\Delta l_y, \Delta l_x\right) \end{bmatrix}, \qquad \Delta\mathbf{l} = \ominus_s^w \mathbf{X}_j \oplus {}^{w}\mathbf{L}_k = {}_w^s\mathbf{T}\,{}^{w}\mathbf{l}. \tag{3.11}$$

On-manifold operations are achieved by the transform operator from the special Euclidean group ${}_w^s\mathbf{T} \in SE(n)$, consisting of the rotation and translation from world to sensor frame. The resulting vector from the robot sensor to the landmark feature is $\Delta\mathbf{l}$, and the measured range residual D is obtained by differencing and norm of the vector $\Delta\mathbf{l}$. The bearing measurement residual is the difference projected angle on the scanner plane using the arc tangent and measurement ψ. A conditional density is obtained by composing the residual function with a multivariate normal

$$\mathbf{z} \sim [\,\mathbf{Z} \,|\, \mathbf{X}_j, \mathbf{L}_k\,] = \mathcal{N}\left(\mu = \delta\left(\mathbf{x}, \mathbf{l}\right), \Sigma = \begin{bmatrix} \sigma_\rho & 0 \\ 0 & \sigma_\psi \end{bmatrix}\right). \tag{3.12}$$

Conditional: Range only measurements

Radio or acoustic ranging information, although not shown in Fig. 3-1, has been the backbone of most navigation systems. We define a range residual between two positions in the same reference frame ${}^{r}(\cdot)$ with a simple residual function

$$\delta\left({}^{r}\mathbf{x}_i, {}^{r}\mathbf{x}_j\right) = \|{}^{r}\mathbf{x}_{i,pos} - {}^{r}\mathbf{x}_{j,pos}\|. \tag{3.13}$$

Note that this function only induces a one degree of freedom potential on the system (single relative constraint relation). In the nonparametric factor group in Section 3.4.3, we will see how multi-path errors can be modeled, which is a common

error source in time-of-flight ranging systems [79].

Conditional: Bearing only (Image Features)

Bearing measurements can be both two and three dimensional, eq. (3.11). Visual features (from camera), are a common three dimensional bearing measurement. Image features are generally introduced as sparse feature points with projection and back projection functions. We refer the reader to references [89,93] for a regular treatise of projective geometry when using cameras features. For example, the projective geometry pinhole camera model with intrinsic distortion matrix $\mathbf{K}$ can be used to construct a measurement residual between pose ${}^{w}\mathbf{X}_j$ and feature point ${}^{w}\mathbf{L}_k$ as follows

$$\delta\left({}^{w}\mathbf{X}_j, {}^{w}\mathbf{L}_k\right) = \mathbf{m} - \mathbf{K}\left[\mathbf{R}\left({}^{w}\mathbf{X}_j\right) \quad \mathbf{t}\left({}^{w}\mathbf{X}_j\right)\right]\mathbf{h}\left({}^{w}\mathbf{L}_k\right). \tag{3.14}$$

The camera measurement (u, v) is stacked in a vector $\mathbf{m} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$, and landmark position into homogeneous coordinate $\mathbf{h}\left({}^{w}\mathbf{L}_k\right) = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$. Camera extrinsic rotation and translation pose information is collected with $\left[\mathbf{R}\left({}^{w}\mathbf{X}_j\right) \quad \mathbf{t}\left({}^{w}\mathbf{X}_j\right)\right]$. We note this residual has two degrees of freedom, while the pose has six and the landmark has three. Multiple residuals can be composed together into a single, higher-dimensional factor and are discussed more in Section 3.4.6.

Conditional: Non-Gaussian

Nonparametric inference allows the user more flexibility in the error distributions that can be used. For example, one factor of interest for acoustic structure from motion [99], which we will show in Chapter 6, is a normally distributed range and bearing constraint as well as uniform elevation distribution, giving the measurement likelihood:

$$\left[\,\mathbf{Z} \,|\, \Theta_i\,\right] = \begin{bmatrix} \mathcal{N}\left(\mu = \delta_{\text{range}}, \sigma_r\right) \\ \mathcal{N}\left(\mu = \delta_{\text{bearing}}, \sigma_\psi\right) \\ \mathcal{U}\left(\pm\text{fov}\right) \end{bmatrix} \tag{3.15}$$

where 'fov' implies vertical field of view from the linear array type sensor. Our inference algorithm requires that samples can be generated from the probability density or distribution.

Conditional: Kinematic constraints

Fig. 3-2 shows kinematic constraints between body pose and foot positions. In this example, kinematic constraints are considered as any direct, on-manifold, rigid transform relation originating from the joint kinematics of the robot legs or arms. Kinematic relations in three dimensional space enforce six degrees of freedom (DOF) in the reference frame ${}^{r}(\cdot)$,

$$\delta\left(\mathbf{X}_i, \mathbf{X}_j\right) = \ominus_j^i \Delta\tilde{\mathbf{X}} \oplus \left(\ominus^r \mathbf{X}_i \oplus {}^r\mathbf{X}_j\right), \tag{3.16}$$

with a relative measured separation ${}_j^i\Delta\tilde{\mathbf{X}}$. The on-manifold operations can be implemented with the special Euclidean group transform $\mathbf{T} \in SE(3)$ in n dimensions

$$\ominus^r \mathbf{A} \oplus {}^r\mathbf{B} = \mathbf{T}^{-1}\left({}^r\mathbf{A}\right)\mathbf{T}\left({}^r\mathbf{B}\right). \tag{3.17}$$

The $SE(n)$ group is based on the rotation and translation of each pose variable [32]. These relations give the inference process a relative distance and orientation constraint between the body end-effector of the vehicle.

By extension, several practical situations require a rigid transform operation with additional ambiguities, for example the mechanical backlash through a gearbox. Such uncertainty are best modeled with non-Gaussian, multi-modal or nonparametric measurement likelihoods. We discuss the nonparametric factor group in more detail in Section 3.4.3 below.

3.4.2 Group II: Real-time Odometry (Inertial Sensors)

We discuss odometry measurements as a time separated form of kinematic relations. Odometry generally involves solving an ordinary differential equation (ODE) from velocity or acceleration measurements. Interpose odometry measurements are treated similarly to any other measurement model, as suggested in eq. (3.5), while using the kinematic relation similar to eq. (3.16). The difference in ODE-type kinematics is that the interpose measurement ${}_j^i\Delta\tilde{\mathbf{X}}$ is created by integrating platform velocities or accelerations.

Conditional: ODE-type Kinematic (Odometry)

Noisy body relative velocity measurements ${}^{b_i}\tilde{\mathbf{v}}$, such as vehicle wheel odometry or Doppler based measurements, are accumulated into a relative distance traveled

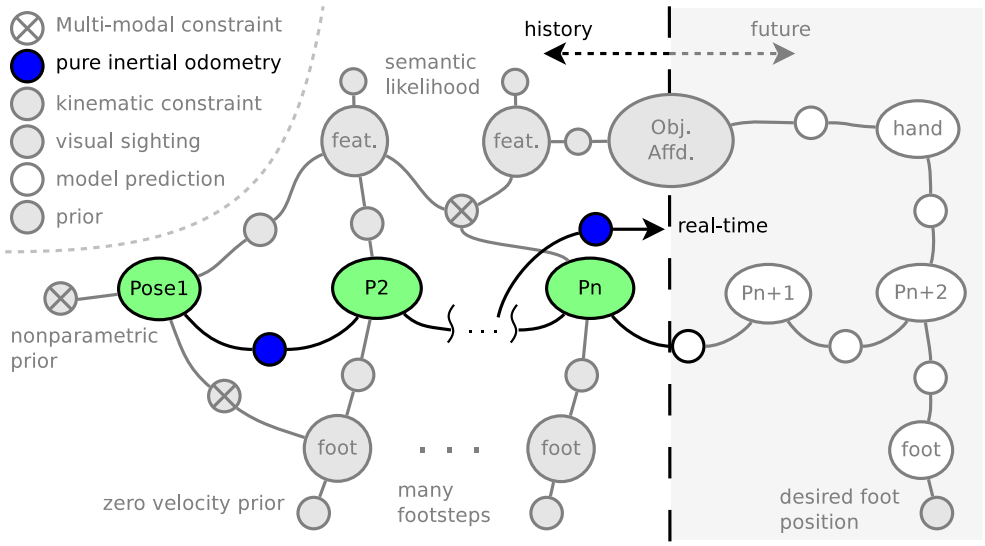


Figure 3-3: Factor Group II: Shaded factor graph illustration from Fig. 3-1 for high-bandwidth and real-time inertial navigation aspects of a humanoid robot state estimation process. A more elaborate example is shown in Fig. 3-6.

${}^{b_i}_{b_j}\Delta\tilde{\mathbf{X}}$ since the last pose ${}^{w}_{b_i}\mathbf{X}$,

$$ {}^{b_i}_{b_j}\mathbf{R} = \prod_{t=t_i}^{t_j} \int_t \left[{}^{b}\omega_{\times}\right] \, d\tau \qquad\qquad {}^{b_i}_{b_j}\Delta\tilde{\mathbf{X}}_{pos} = \int_{t_i}^{t_j} {}^{b_i}_{b}\mathbf{R}\, {}^{b}\tilde{\mathbf{v}} \, d\tau, \tag{3.18} $$

where ${}^{b_i}_{b}\mathbf{R}$ represents the vehicle body rotation in the previous pose ${}^{b_i}(\cdot)$ frame. Rotation could be computed from angular rotation rates ${}^{b}\omega$. This expression uses Lie algebra $\left[{}^{b}\omega_{\times}\right] \in \mathfrak{so}(n)$ to mechanize the rotation manifold.

However, in preparation for nonparametric measurement model discussion, consider a situation with robot wheel slip, which we will model with non-Gaussian, multi-modal or nonparametric distribution.

Conditional and Prior: Body Relative Velocity (Doppler Velocity Log)

Underwater autonomous vehicles (AUVs) generally use a Doppler velocity log combined with an inertial measurement unit to perform sea floor relative navigation [56]. We show a separate factor graph illustration in Fig. 3-4 for using body relative velocities as a navigation aid.

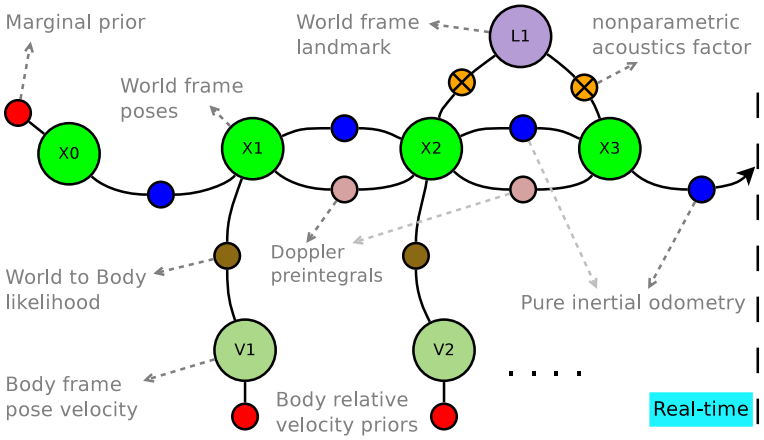


Figure 3-4: Example of a navigation-type factor graph for autonomous underwater vehicle navigation. Green nodes represent vehicle poses relative to the world frame, separated body frame velocities shown as V_1, V_2. Integrated Doppler measurements between poses are shown in light red, as well as absolute velocity snapshots as priors on body frame velocity states. This illustration also shows a landmark feature of interest being estimated from nonparametric-type acoustic factors.

Fig. 3-4 illustrates an AUV near the sea floor, where an onboard inertial measurement unit, acoustic Doppler velocity log and acoustic ranging measurements are available. Inertial odometry factors are in blue along with integrated velocity measurements from a Doppler velocity log in light red. Mapping of an acoustic feature landmark measurement is also shown, and we note an example a nonparametric acoustic measurement is shown in Fig. 3-9.

The factor graph description shown in Fig. 3-4 is grounded by a marginal prior on the pose X_0, which is based on the previous event leading up to the estimated position of the vehicle at X_0. The factor graph segment shown spans the time between the marginal prior and real-time indicated on the right. We can think of this factor graph segment as a fixed lag smoothing framework, whereby the marginal prior is updated and moved forward simultaneous to new poses being introduced on the right, as real-time progresses. We use the inertial odometry factor to project the real-time state estimate from the last pose in the factor graph.

We discuss the time scales of computation and inertial odometry factor in the next section.

Conditional: Real-time Inertial Odometry (Preintegration)

In this section, we provide an overview of using inertial odometry factors in concert with all other measurement likelihoods. Chapter 2 describes conventional methods for processing and fusing inertial sensory data into a navigation solution, generally using Kalman filtering techniques. These methods have become a crucial component of all high-bandwidth, real-time state estimation systems to date [229], and note that factor graph style inclusion of inertial measurements will play an increasingly vital role in all future agile robots.

Sensor bias and scale factor errors result in significant dead reckoning drift when integrated without dynamics calibration and compensation. These errors are a main driver developing dynamical calibration and retroactive compensation models particularly for inertial measurement units (IMUs). Calibration is done in concert with other, possibly lower bandwidth, navigational information. Inertial sensor measurement errors have the advantages of being well understood, strongly unimodal, and accurately modeled with a Gauss-Markov process. Inertial sensor noise input terms are normally distributed, resulting in highly unimodal relative odometry trajectory segments. The strong unimodal nature of inertial measurements has an additional advantage in multi-modal inference schemes, which search for consensus among various sensor data sources. Strong unimodal conditional likelihoods may still transmit multi-modal belief, but quickly help cull out low likelihood modes from other ambiguous measurements, resulting in a reduced computational burden.

The intricate manifold on which the gyroscope and accelerometer bias terms influence an long interpose residual function is not yet been well understood. Consider long interpose integration times with rapidly varying orientation of the platform. Knowing how much to correct the velocity, position, and orientation dimensions for a change in each of the sensor error terms becomes intricate.

A major contribution of this thesis is the analytic derivation of an efficiently computable residual function for pure *inertial odometry* with retroactive sensor bias compensation, shown as the blue factors in Fig. 3-3. This residual function approximates the sensor error manifold with a higher order Taylor expansion. The newly developed inertial odometry measurement likelihood model can then be used as a generic measurement factor in any combination with other factors in a navigation-type factor graph. Chapter 4 is dedicated entirely to the analysis of the new inertial odometry-type factor (measurement preintegration) and defer the reader there for an in depth discussion.

As illustrated in Fig. 3-5, the inertial odometry factors collapse an IMU

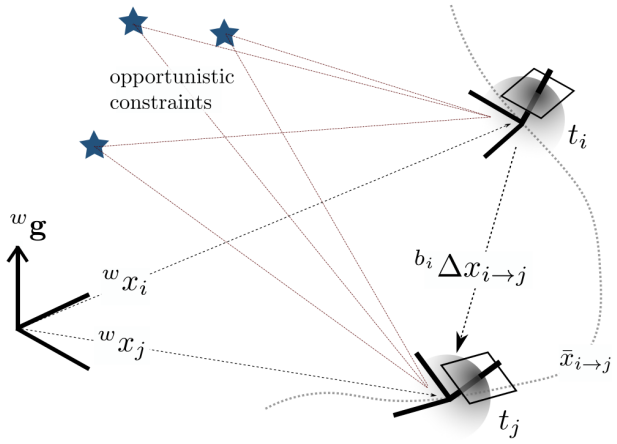


Figure 3-5: Conceptual overview of using pure inertial odometry constraints between world frame poses at time t_i and t_j. Any opportunistic constraint would aid dynamic sensor calibration.

measured trajectory—using inertial navigation-type equations in preintegration form—between two poses $\mathbf{X}_i, \mathbf{X}_{i+1}$ to calculate a relative and noisy preintegral measurement ${}^{b_i}_{b_{i+1}}\Delta\tilde{\mathbf{X}}$. Through the process described in Chapter 4, we define a prediction function to exploit the sensor error parameter manifolds,

$$ {}^{b_i}_{b_{i+1}}\Delta\hat{\mathbf{X}} = (\mathbf{f}_{\ominus} \circ \mathbf{f}_B)\left({}^{w}_{b}\mathbf{X}_i, {}^{w}_{b}\mathbf{X}_{i+1}\right). \tag{3.19} $$

The combination of preintegral measurement and prediction produces a residual function between the two poses

$$ \delta\left({}^{w}_{b}\mathbf{X}_i, {}^{w}_{b}\mathbf{X}_{i+1}; {}^{w}\mathbf{g}\right) = {}^{b_i}_{b_{i+1}}\Delta\tilde{\mathbf{X}} - (\mathbf{f}_{\ominus} \circ \mathbf{f}_B)\left({}^{w}_{b}\mathbf{X}_i, {}^{w}_{b}\mathbf{X}_{i+1}\right). \tag{3.20} $$

The inertial odometry residual is a linear map embedding similar to the kinematic relations in eq. (3.18). The residual function contains all high bandwidth platform motion dynamics, as well as measured gravity ${}^{w}\mathbf{g}$ and sensor biases.

We also note that the nonparametric factor graph solution may result in multimodal estimates of sensor calibration terms, even though the inertial odometry is itself generally combined with a multivariate normal error distribution.

Any number of inertial odometry factors can be assembled from separate IMU measurements in the same factor graph for centralized inference. We specifically note the possibility of placing IMUs on each of the robot's feet, hands or head for

better joint estimation for the extremities. The ensemble of IMUs is then individually calibrated from the aggregate of all sensor information.

Inertial odometry factors give statistical equivalence between all navigation measurements in the centralized factor graph inference scheme. The strong unimodal Gaussian nature of inertial measurements is a great boon for finding consensus amongst other nonparametric, multi-modal sensor measurements.

The inertial odometry measurement likelihoods, as shown in Fig. 3-3, play two vital parts in the overall data fusion process. The inertial odometry factor introduces a dynamic calibration-type factor, where sensor error terms are retroactively estimated in concert with all other available sensory information in the data fusion inference process, while supplying a strong unimodal odometry measurement likelihood.

Before addressing the second advantage of real-time state estimation, we note the principle of feed-forward sensor calibration. The topic is more thoroughly discussed in Section 4.5, but we briefly mention the feed forward aspect here. The residual function in eq. (3.20) is approximated relative to the true rotation rates and accelerations the platform is subjected to. By slowly moving the bias estimates out of the factor graph and into the sensor processing, we can asymptotically improve the approximated residual function accuracy. We assume feed forward corrections are being applied to the real-time state estimate discussion in the next section.

Prior: Intrapose Boundary Marginal (Real-time)

The second major advantage of inertial odometry factors is the ability to obtain a real-time state estimate. We can use the most current inertial odometry preintegral to make real-time state estimate predictions relative to one of the recent poses. We indicate the real-time dead reckoning with an inertial odometry factor and arrow in Fig. 3-3. Using precisely the same inertial odometry residual function, we can produce an algebraically equivalent state estimate solution to a conventional inertial navigation system:

$$ {}^{b_i}_{b}\delta\mathbf{x} = \delta\left({}^{b_i}\mathbf{x}_{0,i}, \mathbf{0}; {}^{w}\mathbf{g}\right), \qquad {}^{w}_{b}\mathbf{x} = {}^{w}_{b_i}\mathbf{x} \oplus {}^{b_i}_{b}\delta\mathbf{x}, \tag{3.21} $$

where ${}^{b_i}\mathbf{x}_{0,i}$ keeps only the bias estimate from pose $\mathbf{X}_i$ and zeros all other terms. The residual function performs gravity and bias compensation to produce ${}^{b_i}_{b}\delta\mathbf{x}$ relative to pose ${}^{w}_{b_i}\mathbf{X}_i$. The $\oplus$ operation stands off, or projects, the current inertial odometry derived trajectory segment to the real-time state estimate. We select the reference pose location from a marginal belief estimate ${}^{w}_{b_i}\mathbf{x} = \operatorname{argmax}_{\mathbf{x}} \left[{}^{w}_{b_i}\mathbf{X}_i \,|\, \mathbf{Z} \right]$. These computations are fairly small and can be done at full sensor rate. These steps

are the preintegral equivalent to previous recursive inertial navigation schemes. We emphasize that the real-time prediction does not need to consist of a purely inertial odometry factor.

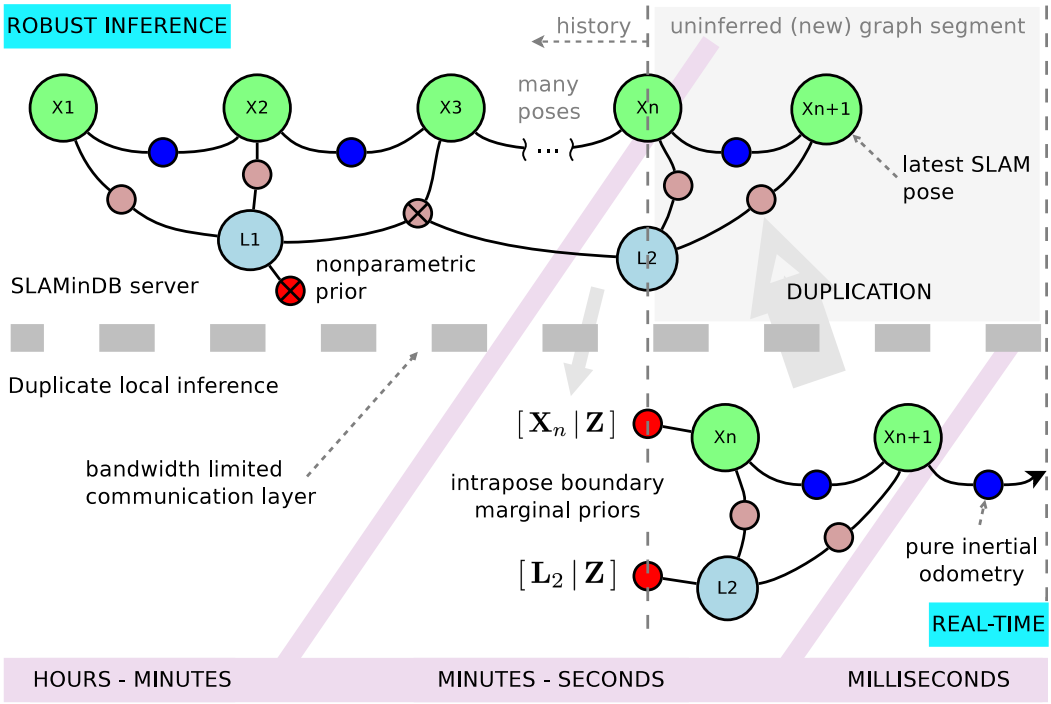


Figure 3-6: Timescales illustration of combining real-time state estimate with robust inference. The top part of the figure illustrates robust inference on a large factor graph, with the most recent inference result for poses $\mathbf{X}_k, \;\; k = 1, 2, ..., n$ on the left. A smaller duplicate solution beyond pose $\mathbf{X}_n$, grounded by two boundary marginals, is shown on the bottom part. The real-time state estimate is achieved with most recent inertial odometry factor. This arrangement allows for both slower robust computation times and fast hard real-time state estimation. Information between the SLAM server and local inference task travels across a limited communication layer.

Fig. 3-6 illustrates the real-time state estimation process further. We can choose to project the current inertial odometry preintegration residual from a recently inferred SLAM pose. We can also make a local duplicate copy, shown in the bottom part of the figure, which fuses the most recent sensor data. A previous suggested approach, concurrent filtering and smoothing [117], advocates a serial mechanism

where a smaller local factor graph solves variables first and then pass factors and variables over to the larger solution.

In contrast, we suggest duplicating a portion of the factor graph in two copies and solving simultaneously, while "grounding" the smaller graph against the intrapose boundary marginal prior from the larger solution. We call the marginal belief on the intermediate pose the intrapose boundary marginal. A smaller simultaneous factor graph can immediately duplicate all new factors and variables beyond the intrapose boundary marginal, even though the larger solution will still be computing on the previous factor graph snapshot. The replicated small factor graph structure is solved much faster and in parallel to the larger factor graph. The combination of large robust factor graph, small local duplicate factor graph, and inertial odometry residual function allows great flexibility and longer computation time, while maintaining real-time state estimation capability. This duplication process between local shorter and larger long term factor graphs was recently shown to also occur in biological systems: Research in memory mechanisms shows that mammals simultaneously create both long and short term memories [124], and do not transfer short term to long term memory as many researchers had thought.

The larger factor graph takes longer to compute, but shares the most recent inferred pose marginals to the smaller local factor graph solution as new marginals become available. Once a new marginal prior is available from the larger solution, we insert a new boundary condition on the local duplicate. The smaller local factor graph is then reduced in size and the process continues. Note the current inertial odometry preintegral is grounded against the most stable pose in recent history to give a real-time state estimate.

3.4.3 Group III: Nonparametric Measurement Likelihoods

Fig. 3-7 highlights the nonparametric likelihoods in the common humanoid factor graph example. Gaussian measurement model assumptions are often oversimplifications of true events which do not correctly capture the underlying uncertainty or ambiguities. Nonparametric inference allows for likelihood models that pass more enhanced measurement uncertainties to the back-end solver, over pure Gaussian-type SLAM techniques. Nonparametric factors result in a multimodal, nonparametric posterior which describes the possible states causing the particular sensor measurements to be made.

Consider the humanoid robot in Fig. 1-2 being uncertain about foot-to-ground contact events, gearbox backlash, or repeated patterns observed by a camera or laser scanner. Each of these measurement uncertainties is poorly described by

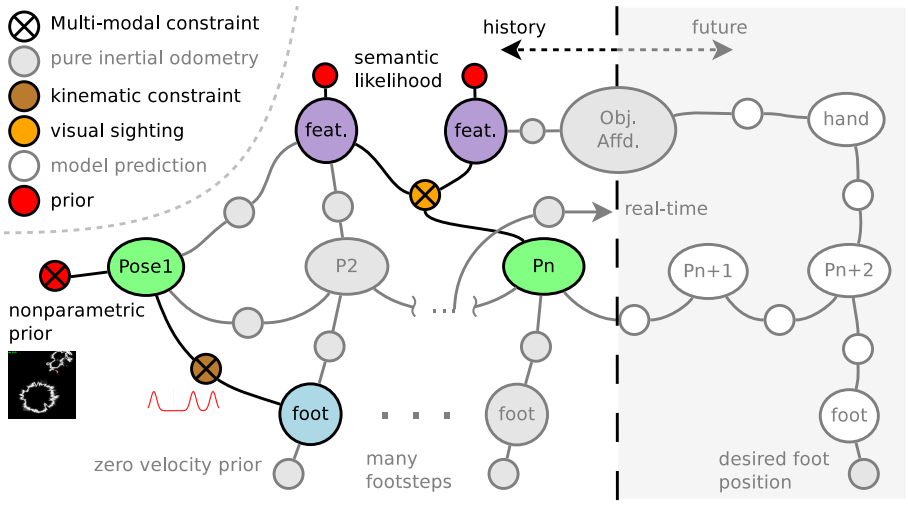


Figure 3-7: Factor Group III: Shaded factor graph illustration from Fig. 3-1 for multi-modal and nonparametric measurement aspects of a humanoid robot state estimation process.

the normal probability distribution. In the sections that follow, we look at new nonparametric-type measurement likelihoods, allowing for multi-modal belief to exist in the factor graph posterior distribution. We refer the reader to Chapter 5 for the in-depth discussion on marginal posterior estimation of these navigation-type factor graphs. We also emphasize that the nonparametric factor types are intended to remain unchanged in the factor graph structure before, during and after inference, unlike a prefiltering-type approach where the structure of the graph is changed before inference is performed.

Conditional: Multi-modal Loop Closures (Uncertain Data Association)

We first look at the multi-modal data association type factors which introduce uncertainty through a discrete selection variable, as indicated by the multi-modal factor between the two features in Fig. 3-7. A major challenge in all SLAM based systems is obtaining a high degree of certainty in proposed loop closures, before introducing the factor into the factor graph. By loop closure, we mean a previously observed feature of interest is re-observed, creating a relative constraint between two time separated portions of the factor graph. An alternative approach common in the SLAM literature uses marginalized interpose transformations as loop closures instead.

Consider ambiguous measurements where we are not sure if a new feature observation corresponds to a previously observed feature. Describing this uncertainty in a Gaussian unimodal measurement model is impossible. To this end, we introduce multi-modal constraints, as shown in Fig. 3-8.

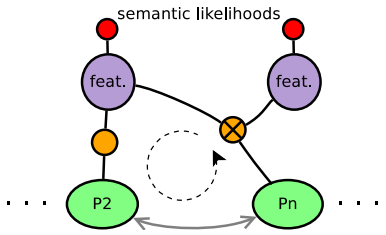


Figure 3-8: Multi-modal data association measurement model, where poses P_2 and P_n are separated by some long chain of events in a larger factor graph. A feature, first sighted from pose P_2, may or may not have been sighted again at pose P_n. A multi-modal constraint indicates the measurement sample obtained could be from a possibly new feature, or the same feature as before. A feature re-sighting should enforce a loop closure as strong piece of navigation information.

The multi-modal constraint between two features in Fig. 3-7 shows a measurement likelihood model where the user is uncertain about the data association of a particular measurement. Rather than make a hard decision about which feature actually produced the measurement, we can instead introduce uncertainty through a multi-modal measurement and let the back-end inference process decide. In turn, consensus in the nonparametric posterior distribution estimate reduces all unknown data association decisions to a limited number of multiple likely modes.

A categorical distribution is a natural choice for describing the decision γ on which landmark feature the measurement originated from. We can define the residual function between the pose and possible landmark features as

$$\delta\left(\mathbf{X}_i, \mathbf{L}_1, \mathbf{L}_2, \gamma\right) = \ominus \mathbf{X}_i \oplus \mathbf{L}_\gamma, \qquad \gamma \sim Cat\left(\rho\right). \tag{3.22}$$

The weight vector ρ represents the likelihood statistics of a true or false loop closure. For a practical example, consider a normally distributed range measurement

from current pose $\mathbf{X}_i$ to one of two equally probable targets $\mathbf{L}_1, \mathbf{L}_2$

$$\begin{aligned}\gamma &\sim [\,\Gamma\,] = Cat\left(\rho = \begin{bmatrix}0.5\\0.5\end{bmatrix}\right), \\ \mathbf{z} &\sim [\,\mathbf{Z}\,|\,\mathbf{X}_i, \mathbf{L}_1, \mathbf{L}_2, \Gamma\,] = \mathcal{N}\left(\mu = \delta\left(\mathbf{X}_i, \mathbf{L}_1, \mathbf{L}_2, \gamma\right), \sigma\right).\end{aligned} \tag{3.23}$$

Note, we are omitting the numbering of measurements $\mathbf{Z}_k \doteq \mathbf{Z}$.

Once the categorical sampling weights are known, the overall factor graph inference procedure can then recover the correct multi-modality in concert with all factors and variables. We may either know the probabilistic weighting ρ as a hyper parameter as specified by the user, or by selection from knowing additional information. We now turn our attention to the case where more information is available and use semantic information as the working example, and note that a further discussion on how ambiguous associations arise is illustated and discussed in Fig. 8-16.

Dirichlet process for Incorporating Semantic Labels

Semantic priors can be incorporated with multi-modal factors for uncertain data association between two features, as shown in Fig. 3-7. Intuitively speaking, semantic information should factor into the loop closure data association process when such measurements are available. Consider a simple example where objects in the world are detected by some process, with likelihood predictions for their semantic label (say class A or B). The platform later returns to the same area and sights an object with likelihood of being either object B or C. In the case where the first and second sightings of objects were strongly classified with semantic labels A, C respectively, the odds of being a valid loop closure is low.

In the alternate case, we may have two objects within the odometry drift uncertainty where both sightings are strongly of the same label B, B, improving the odds of a loop closure. We note in the later B, B case that two of the same objects may well be in close proximity, and we are still not sure if a loop closure is in fact in effect. It is important to note that a correctly modeled factor graph should not care if a loop closure is valid or not, but rather the consistency. The measurement likelihood models are representative of the measurement uncertainty. The inference driven posterior belief estimates will then allocate probable belief using all available information. A last case involves measurement pairs A, B or B, C. We also note the classification system could be in error or weakly preferring one label over another.

We can account for the different cases listed above by introducing a new variable Υ, indicating the odds of the first and second sighting have the same semantic classification label, and the associated conditional belief $[\,\Gamma \,|\, \Upsilon\,]$. For example, we can adjust the weighting ρ of eq. (3.23) according to visual segmentation [9] likelihoods of classified objects. The weighting vector ρ is affected by semantic likelihoods and guides the categorical prior likelihood $[\,\Gamma \,|\, \cdot\,]$ of the multi-parametric model:

$$[\,\mathbf{Z}_k \,|\, \mathbf{X}_i, \mathbf{L}_1, \mathbf{L}_2\,] \propto \sum_{\Gamma} [\,\mathbf{Z}_k \,|\, \mathbf{X}_i, \mathbf{L}_1, \mathbf{L}_2, \Gamma\,] \times \sum_{\Upsilon} [\,\Gamma \,|\, \Upsilon\,]\,[\,\Upsilon\,]\,. \tag{3.24}$$

The new conditional is defined as

$$\left[\Gamma = \begin{bmatrix}1 & 2\end{bmatrix} \,|\, \Upsilon = \begin{bmatrix}\text{same} \\ \text{different}\end{bmatrix}\right] \equiv \begin{bmatrix}0.5 & 0.5 \\ 0.1 & 0.9\end{bmatrix}. \tag{3.25}$$

We take $\Gamma = 1$ as associating the original feature measurement with the first landmark feature – thereby enacting that loop closure. Similarly, $\Gamma = 2$ implies associating the feature measurement with the alternate landmark feature. We choose $[\,\Gamma \,|\, \Upsilon = \text{same}\,] = Cat\,([0.5, 0.5])$ uniform in the case that two semantic measurements are the same, since two instances of the same object may exist in close proximity. Conversely, we choose $[\,\Gamma \,|\, \Upsilon = \text{different}\,] = Cat\,([0.1, 0.9])$ to favor landmarks $\mathbf{L}_1, \mathbf{L}_2$ being different features and thereby avoiding the loop closure. We keep 0.1 uncertainty to allow for errors in classification which would otherwise result in a false negative. This factor type can be used in any combination between features and landmarks resulting in interesting loop closure dynamics. We could also increase the uncertainty to any M number of proposals, $\gamma \in \{1, 2, 3, ...M\}$.

As a side note, Dirichlet priors are a natural proposal distribution for the Categorical distribution

$$\rho \sim Dir\,(\alpha)\,, \tag{3.26}$$

based on the concentration parameter α, in which a simulated system can use the parameter to propose valid or invalid loop closures. Large values of α would correspond to very strong data association likelihoods and indicate whether a valid loop closure has been found. Small values of α would allow large variability in the eventual sampled loop closure decision γ. From this we deduce $Dir\,(\alpha)$ is representative of a real loop closure proposal situation. Small values of α correspond to situations where it is difficult to conduct data association accurately.

Conditional: Multi-modal (Multi-parametric)

At the bottom left in Fig. 3-7 we show a different type of multi-modal belief, in this case associated with a kinematic constraint. We note that parametric residuals, as discussed in 3.4.1, can be enhanced to multiple measurement modes using a categorical selection process similar to the selection process described above. Instead here we show how a regular parametric model can be associated with uncertain parameters. The residual function between variables Θ_i could be affected by any number of parameter selections Λ_γ:

$$\delta\left(\Theta_i, \Lambda_\gamma\right), \qquad \Lambda_1 = \mu_1, \;\; \Lambda_2 = \mu_2, \qquad \gamma \sim Cat\left(\rho\right). \tag{3.27}$$

This is an ambiguous measurement likelihood, and is represented by selecting different measurement values along with an existing parametric measurement model. For example, the gearbox in a kinematic joint may have a certain amount of play, which results in additional uncertainty in the kinematic position of joints. Another example may be estimating kinematic contact with objects in the world, but are corrupted with multiple observed "contact events". For example, the foot of a robot may prematurely hit an obstruction while trying to estimate the ground position.

A clear example of this multi-parametric case is when a humanoid robot, while walking, is trying to estimate when a foot-to-ground contact event has occurred. Consider the robot busy putting one foot down on the ground, while receiving varying messages from the foot mounted force sensor, depending on how weight and balance are transferred. From experience, we know this trigger of foot contact is critical to estimating state accurately. In the multi-modal framework, we can store the possible measurements to represent simultaneously in contact and not in contact states. The back-end inference will through consensus with other information allocate or defer which of the modes were actually in effect at each measurement factor.

The concept extends to other ambiguous measurements. Consider backlash between gears or joints in the robot kinematic structure. We can model each of the two extremes as two modes, assuming a kinematic structure under load will push to either of the extremes:

$$\delta\left(\mathbf{X}_{body}, \mathbf{X}_{foot}, \Delta\mathbf{X}_\gamma\right) = \ominus\mathbf{X}_{foot} \oplus \left(\mathbf{X}_{body} \oplus \Delta\mathbf{X}_\gamma\right), \qquad \gamma \sim Cat\left(\rho\right), \tag{3.28}$$

where $\Delta\mathbf{X}_\gamma$ represents either of the body-to-foot transforms.

Conditional: Nonparametric

The idea of a multi-parametric relative constraint, by means of a residual function, can be extended to a fully nonparametric representation. Consider defining a nonparametric relative transform between variables in the system. Beam forming type measurements, where a heat map of acoustic or electromagnetic signals are produced, can be approximated by a nonparametric belief. Fig. 3-9 shows an actual relative range and correlator output recovered from an underwater acoustic ultra short baseline acoustic system. A range and bearing measurement is made as to where a ping originated from, relative to the acoustic microphone array. We would have to convolve the correlator output with a channel model to obtain a more accurate belief estimate, but concepts for working with nonparametric factors are the same. Consider using a nonparametric measurement such as Fig. 3-9 as a factor in the factor graph.

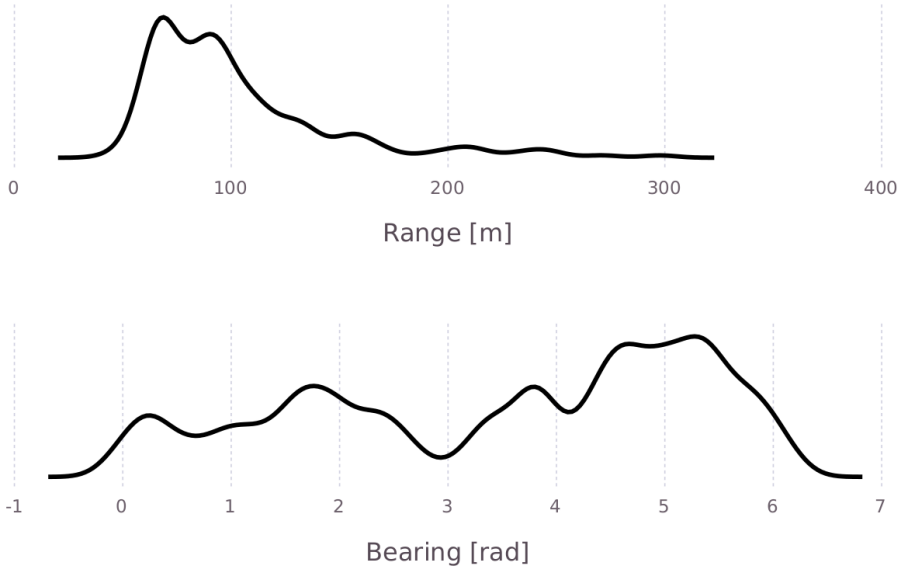


Figure 3-9: A nonparametric, relative range and bearing measurement from underwater beam formed acoustic data.

We approximate the underlying belief with a nonparametric kernel density estimate [208] function, from which relative sample points can easily be generated. We can use the samples to estimate the convolution between a nonparametric conditional and a transmitted belief.

Futhermore, we can use nonparametric conditional beliefs to represent a new

and emerging type of intepose relation, originating from convolutional neural networks. Consider a recognition system that has been trained to transcribe camera images of familiar objects into beliefs of the relative pose between the camera and a familiar object [84]. These beliefs can be approximated in much the same way as shown in Fig. 3-9 and introduced into our general nonparametric factor graph model.

Prior: Nonparametric and Multi-session Priors

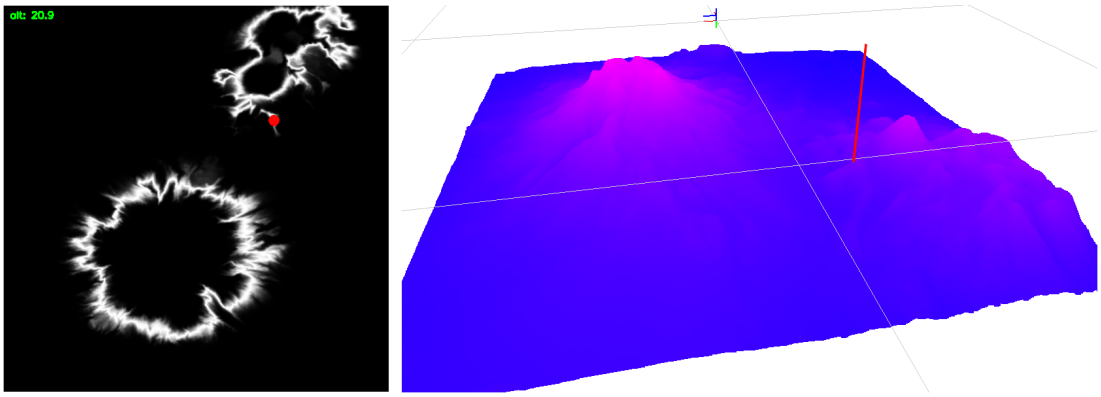


Figure 3-10: Nonparametric navigation example, showing measurement likelihood function on the left and corresponding true position over known topographic terrain on the right. Given a known map and altitude measurement above the terrain, shown as red line and dot, we can construct a nonparametric measurement function of the likely position of the vehicle.

Next, we discuss the nonparametric prior, shown as the red cross unary factor in Fig. 3-7. Consider a navigation measurement which limits the likely position of the platform to some region of the state space, but is poorly approximated by any parametric model. For example, consider trying to incorporate topographic information from a known map into a navigation solution, given an estimate of altitude above or below mean sea level, as illustrated in Fig. 3-10. Current state-of-the-art parametric factor graph descriptions are not able to represent such general prior likelihood beliefs. We suggest a unary nonparametric factor as a natural and elegant way to fully utilize such measurements in a centralized manner. Fig. 3-10 illustrates the nonparametric style measurement obtained when using the exponent of negative squared difference of altitude from a known topography heightmap.

Furthermore, consider that the posterior from a previous navigation state estimate may also be nonparametric and highly non-Gaussian. We will later see with the SLAM-e-donut canonical example — in Section 6.5 — that a purely Gaussian factor graph can indeed result in a highly non-Gaussian posterior distribution. We suggest a nonparametric prior belief, using kernel density estimation [208], to our SLAM state estimation process.

Next, we consider using the marginal belief estimate of landmark features in one robot navigation session as a prior in a separate or parallel robot navigation session. We avoid an in-depth discussion on multi-session robot navigation and mapping here, but introduce the idea of using multi-parametric or nonparametric priors from adjacent sessions as prior information to a current robot session.

Consider either the same robot, or multiple robots, operating in the same space observing the same landmarks in the world. We would like to incorporate information from all sessions together into the same inference for better global map accuracy and coverage. By using a common set of landmark features and latest marginal estimates from each session, we can form a relation between each of the navigation solutions. At the start of a new inference cycle — using the centralized architecture discussed in Chapter 7 — we take the marginal position beliefs from common landmarks found in other robot navigation sessions and create a new prior for on the same landmark in the local factor graph session. Each of the factor graph inference sessions are then left to continue independently, but update the stacked nonparametric inter-session priors at the start of each inference iteration for each of the robots individually.

3.4.4 Group IV: Trajectory Planning

Trajectory planning relates to the last highlighted grouping of factors in the generic factor graph overview. The future trajectory planning of the robot state is shown to the right in Fig. 3-11. Although we will not analyze these factors in detail, it is important to note their significance in the robot navigation task. Furthermore, a nonparametric inference solution will allow the user more flexibility in multi-policy trajectory planning.

Conditional: Vehicle Dynamics (Knot Points)

Given the residuals above, and in particular the ODE-type kinematic examples, we can define residual functions based on predictive dynamic models of the vehicle. Each evaluation of the model prediction factor internally performs integration

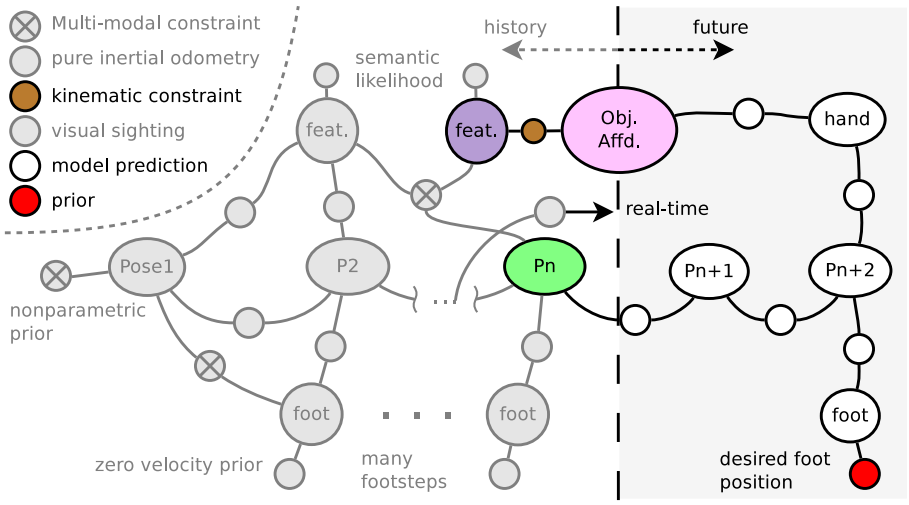


Figure 3-11: Factor Group IV: Shaded factor graph illustration from Fig. 3-1 of future trajectory planning (and possibly multi-modal) aspects of a humanoid robot state estimation process.

of an ODE model, producing a relative interaction between possible future poses states. This approach is essentially an embedded version of well known trajectory optimization or model predictive control strategies. We note, however, our approach requires the belief space understanding of vehicle states. Instead, with the belief formulation, we limit the possible state of control actuators output as a likelihood model – what is the likely value of actuator output.

With the humanoid robot example (shown in Fig. 1-2), consider guiding the robot toward a user specified foot stance position while reaching out with a hand actuator to grab the handle on a door. The factor graph in Fig. 3-11 indicates the desired foot stance position with the red unary prior, and the hand manipulator constrained to an affordance of the door handle. Affordances are discussed in Section 3.4.5, but note here that an affordance is a virtual representation of some object of interest. The virtual model affordance position is aligned and tracks the true position of the door handle, and constrains the factor graph future hand position estimate to the required kinematic position (or relative relation).

Conventionally, open or closed loop model dynamics are modeled between "knot points" in the trajectory, to constrain future motion using an optimization problem. Platform dynamics are constrained through penalty functions or hard equality constraint terms. For illustration, we discuss a model much simpler than the humanoid robot. Consider a two dimensional $\mathbf{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$, veloc-

ity model representing vehicle dynamics, with throttle and steering angle input $\mathbf{u} = \begin{bmatrix} {}^{b}v & {}^{b}\psi_u \end{bmatrix}^T$, such that

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right), \qquad \delta\left(\mathbf{X}_n, \mathbf{X}_{n+1}, \mathbf{u}\right) = \ominus \mathbf{X}_{n+1} \oplus \mathbf{X}_n \oplus \left(\int_{t_n}^{t_{n+1}} \dot{\mathbf{x}}\, d\tau \right). \tag{3.29}$$

Where we assumed some transition function $\mathbf{f}$ which describes the model dynamics between two possible future pose states $\mathbf{X}_n, \mathbf{X}_{n+1}$. Future pose states are commonly referred to as knot points. The integration between the predicted pose points is repeated at each inference step and can be simplified to reduce computation complexity. The inference residual function now includes states $\mathbf{u}$ relating to the required input to modify the trajectory of the vehicle towards the user desired goal.

Multi-policy Inference

While existing trajectory optimization methods consider a mean, or single mode, one could also simultaneously introduce multiple policies through a categorical selection criteria – similar to the multi-parametric residual case above. Multi-modal planning is then directly based on the centralized robot state estimation process. Consider navigating around either the left or right side of an obstruction. These two options can be considered as as two policies (modes) in the planning mechanics.

3.4.5 Affordances

The last aspect of our humanoid example factor graph is the affordance node, in Fig. 3-11. Affordances [54] play a vital role in robot manipulation, and will likely play a major role in robot navigation and state estimation as well. An affordance is a virtual geometric representation for an object of interest. Computer software can be developed to manipulate and interact with an affordance, rather than raw sensor data, which may be difficult to interpret. In turn, the geometric position of any or all affordances must be aligned with the real positions of the objects they represent. We can include object models with known relative kinematic constraints to mapped features in the real world, thereby incorporating object tracking, mapping, state estimation and trajectory planning into a central "location awareness" inference framework.

3.4.6 Compound Factors Example: Multiple Features

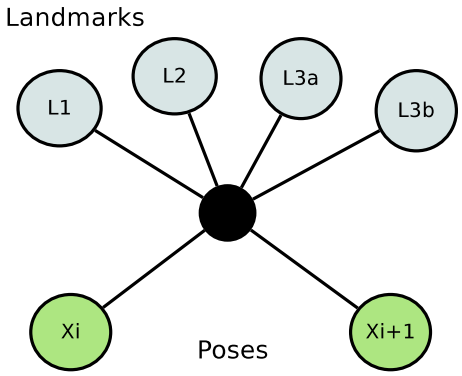


Figure 3-12: Multiple features factor, illustrating compound functions for higher dimension factors in a two dimensional robot navigation example. The factor contains multiple bearing measurements combined together for full dimension constraints between poses $\mathbf{X}$ and landmarks $\mathbf{L}$. The factor represents a balance between computational expedience and robustness in a multi-modal inference framework. Landmarks $\mathbf{L}_{3a}, \mathbf{L}_{3b}$ represent data association uncertainty between observed features.

In the earlier sections, we discussed measurements which may have lower degrees of freedom than the variable nodes they depend on. As a result, probabilistic inference by convolution of the conditional belief becomes more difficult, but not impossible. In the bearing measurement section we suggested a compound type factor be used, where multiple measurements are accrued into a single factor which has higher dimensionality. In this section we present an example were multiple bearing measurements can be combined into a single factor between multiple variables in the factor graph. As an added benefit, we incorporate some data association uncertainty to create a new type of navigation factor which blends computational considerations with robust inference techniques, called the multiple features factor.

Fig. 3-12 shows an example factor between two poses and several landmarks, composed from multiple bearing measurements. The factor represents the analytic solution between all variables given three independent bearing measurements to different landmarks $\mathbf{L}$. This is a two dimensional example where we can completely resolve the three degrees of freedom, position and orientation, of either pose $\mathbf{X}_i$ or $\mathbf{X}_{i+1}$ relative to the other variables and the three bearing measurements associated with that pose. In addition, we allow for data association uncertainty

if two bearing measurement associations to features are uncertain, as indicated by $\mathbf{L}_{3a}, \mathbf{L}_{3b}$. There are eight bearing measurements ψ, two pose and four landmark associations built into the factor illustrated in Fig. 3-12. We can write down the required six dimensional residual function, using the bearing function $\Psi(\cdot)$

$$\delta(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_{3a}, \mathbf{l}_{3b}, \gamma) = \begin{bmatrix} \psi_{i,1} - \Psi(\mathbf{x}_i, \mathbf{l}_1) \\ \psi_{i,2} - \Psi(\mathbf{x}_i, \mathbf{l}_2) \\ \psi_{i,3\gamma} - \Psi(\mathbf{x}_i, \mathbf{l}_{3,\gamma}) \\ \psi_{i+1,1} - \Psi(\mathbf{x}_{i+1}, \mathbf{l}_1) \\ \psi_{i+1,2} - \Psi(\mathbf{x}_{i+1}, \mathbf{l}_2) \\ \psi_{i+1,3\gamma} - \Psi(\mathbf{x}_{i+1}, \mathbf{l}_{3,\gamma}) \end{bmatrix}, \quad \gamma \sim Cat(\rho). \tag{3.30}$$

This residual function can be combined with a probabilistic density to form a full dimension, robust probabilistic measurement likelihood.

The multiple features factor can similarly be extended to three dimensions for camera based feature measurements, and is then equivalent to common five or eight point relative orientation estimation approaches [93, 176]. We note that this factor can be formed from randomly selecting groupings of tracked features into a couple multiple feature factors, along with the data association ambiguities to form a robust interpose odometric inference procedure. This procedure should, through consensus, find the so called "inlier" set of features by ignoring bad associations. The compound factor solves inefficiencies associated our current Gibbs sampling based nonparametric inference procedure, and refer the reader to Chapter 5 for more details.

3.4.7 Null-hypothesis

A further aspect common to all factors is how to handle the null-hypothesis case and do inference over such as factor, or lack thereof. In principle, a null-hypothesis implies that a relation between variables is void and should be ignored. Null-hypothesis implies the factor that exists in the factor graph should not be considered during the inference procedure. We note that our approach is aimed at keeping the structure in the factor graph constant during the inference procedure, and emulate the null-hypothesis case through a categorical distribution. We do not ignore the constraint, but instead inject a large amount of entropy to the variable of interest under the null-hypothesis case. Through this mechanism we are at least able to explore the local region around the current state of variables during the inference procedure and negate any factors influence through consensus with

other measurement information. We note the extreme case where all factors have the possibility of being null-hypothesis, which would result in a uniform belief over all system variables. We argue that by introducing large uncertainty to each factor in the region of the current state emulates the local unform uncertainty.

3.5 Marginalizing out Discrete Variables

Many, but not all, of the measurement likelihood models described above introduce multi-modality through a discrete variable γ. We propose to always marginalize out the decision variables when taking the product between different belief functions. The marginalization of discrete decision variable produces multi-modality in the remaining continuous domain variables. Therefore, in general, we expect to marginalize out discrete variables from any such sensor measurement likelihood functions, and will only happen at inference time, as follows:

$$[\, Z \,|\, \Theta_j \,] = \sum_{\gamma}^{\Gamma} [\, Z \,|\, \Gamma, \Theta_j \,] [\, \gamma \,], \tag{3.31}$$

where, the prior belief $[\, \Gamma \,]$ represents the data association certainty obtained from user input, or some other likelihood process, represented by the weighting vector ρ.

A key advantage in our approach of marginalizing out discrete variables at inference time is that that all transmitted beliefs will be in a common continuous domain. We will see in Chapter 5 how convolution and infinite product operations are standardized around continuous nonparametric belief functions. By marginalizing all discrete variables, the contents of a particular measurement likelihood is "black boxed" and standardizes the inference operations. New factors or variables allow the inference process to incorporate new information, which may focus computation on an altered subset of dominant modes – without having to update any discrete variables. This approach allows deferred data association decisions to be left as late as possible, and treat the Categorical weighting ρ as a hyper parameter.

3.6 Inconsistent Modeling

This section discusses a further restriction on the types of problems to be considered, and relates to the issue of model consistency. A factor graph could be constructed with any random variables and likelihood potentials between them, but

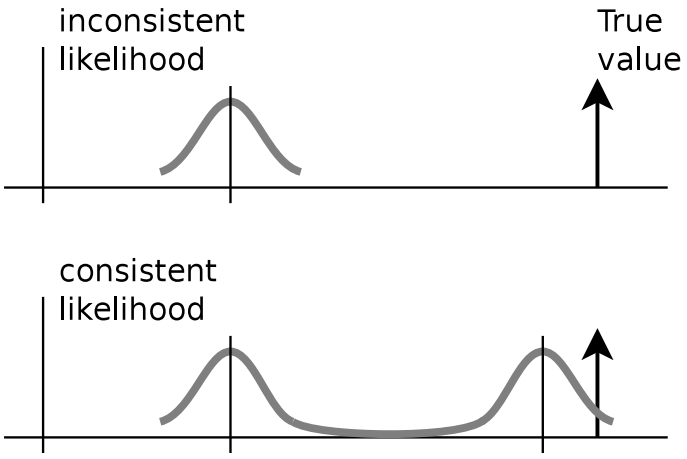


Figure 3-13: Illustration of inconsistent modeling, where assumed model as gray bell curve is a poor probability density estimate of actual value shown by Dirac spike (black arrow) on the right. The bottom figure shows a different multi-modal likelihood which does support probability mass around the true value which is considered a consistent likelihood model for the true event.

is not representative of any true sequence of events. Alternatively, we could model a system of variables with extremely wide uncertainties, resulting in a correct but uninformative posterior distribution.

By means of an example, consider a two pose system where a robot drives 10 units in a one dimensional world. We instantiate a prior position unary factor to the first pose and pairwise odometry measurement likelihood factor to the second pose. The odometry factor can be made inconsistent by specifying an exceedingly unlikely situation. For example, the odometry measurement uncertainty could be taken as normally distributed with mean of 5 units traveled, with only 1 unit standard deviation (σ); resulting in the actual 10 units traveled being extremely unlikely (5σ). We would consider this situation inconsistent, and illustrated graphically in the top portion of Fig. 3-13. We contrast a consistent multi-modal likelihood belief in the bottom portion of Fig. 3-13.

In the case of loop closures, we note that the odometry drift estimates must be representative of the actual drift accumulated. We regard a loop closure potential between two variables, which by odometry are unlikely to be in close proximity as an inconsistent model. Our inference algorithm will use user supplied measurement likelihood functions to propose regions of the state space that must be explored. The posterior solution is found by stacking all likelihood proposals and

resolving the regions of consensus.

3.7 Conclusion

Most, if not all, SLAM systems to date perform inference using squared error cost functions aggregated into a optimization objective function. The squared cost implies a normally distributed measurement likelihood model, which we have shown as too restrictive in many situations. The difficult cases arise from ambiguous measurements. For example, a normally distributed measurement model biases data association uncertainty, or makes poor use of information when underlying uncertainty is non-Gaussian. Instead, we choose to use kernel density estimation for approximating all marginal beliefs in the system, and naturally encapsulates multi-hypothesis and nonparametric type beliefs that an associated inference procedure can exploit for more representative posterior estimation. This approach allows us to perform inference on a static factor graph structure, without trying to modify the graph structure before or after inference.

This chapter introduces a number of nonparametric measurement likelihood functions by focusing on the importance of defining an on-manifold measurement residual function combined with a consistent probabilistic error distribution. The measurement models presented in this chapter provide a solution for many of the problems faced by current SLAM type measurement factors in use today. We show several measurement likelihood potentials to be used in any combination for constructing nonparametric navigation-type factor graphs. To our knowledge, none of the current SLAM solvers are able utilize semantic information in a factor graph during inference without modifying the actual structure of the graph. Our approach is able to perform incremental inference over semantically labeled and the ever changing factor graphs, and recover the most dominant modes according to all available measurement information.

The measurement likelihood functions presented are by no means a complete list. In particular, we discussed how inertial odometry factors (inertial sensor preintegrals) play a special role in real-time, high-bandwidth state estimation. We illustrated how longer running computation times required for robust multi-sensor data fusion that can be combined with a smaller duplicate factor graph inference task for a combined fast localization and robust mapping system. Chapter 4 investigates the inertial odometry factor in detail and Chapter 5 discusses the *Multi-modal iSAM* incremental inference algorithm to user specified nonparametric factor graphs.

Chapter 4

Inertial Odometry

In Chapter 3, we described new parametric and nonparametric measurement likelihood models for navigation type factor graphs. This chapter derives and details the new continuous time and second order inertial odometry measurement factor for high-bandwidth, real-time aspects of the navigation-type factor graph descriptions. Robust multi-sensor fusion through nonparametric inference with *Multimodal iSAM* is discussed in Chapter 5.

4.1 Introduction

Inertial sensors are a corner-stone of high bandwidth navigation systems. They offer a means to "black box" dead reckoning, but suffer difficulties in double or triple integration of sensor errors. Inertial sensors capture vehicle dynamics and other effects such as gravity at high rate, but also include measurement errors. For example, a gyroscope bias offset results in accrued orientation error, which in turn results in incorrect gravity compensation of acceleration measurements. Integration of misaligned gravity quickly results in significant positioning error.

Over the past two decades, simultaneous localization and mapping [135] has been a major area of navigation-related research. Sparse factor graph methods have been developed to allow a new perspective on navigation and localization, bringing into question whether Kalman filtering [56] is still the best way of inferring inertial sensor calibration parameters.

Fig. 4-1 illustrates how summarized inertial odometry constraints can be used in navigation type factor graphs. The figure shows how leg kinematics, inertial odometry and visual sightings of opportunistic features as well as loop closures can interact in a centralized framework. We note that a pure inertial odometry

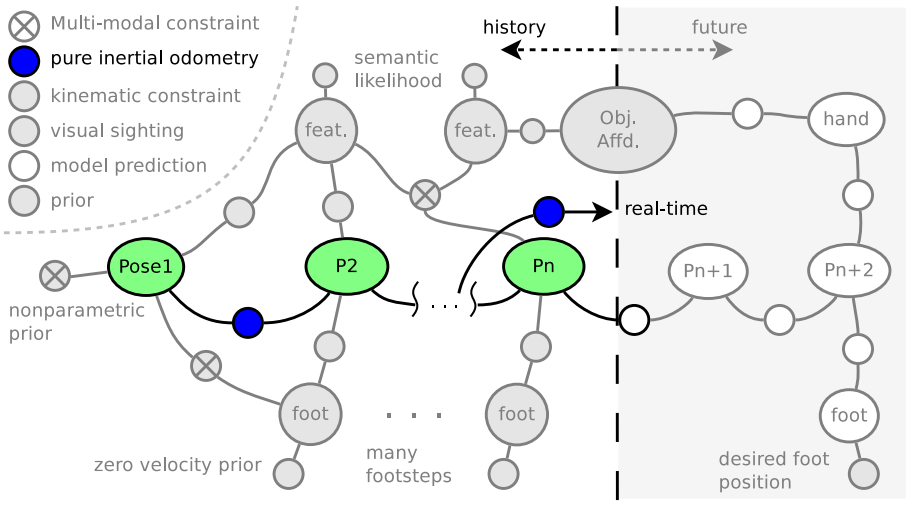


Figure 4-1: Factor graph showing pure inertial odometry constraints, aided by forward kinematics from the legs and monocular visual feature sightings through the head mounted camera (associated with Section 8.3).

constraint can also be used to predict real-time state estimates based on a robust factor graph inference result, as shown. Future motions can also be optimized by enforcing vehicle model dynamics towards a desired goal.

However, when we try to use inertial measurements as odometry likelihoods in a factor graph formulation, we find that compensation of sensor errors (dynamic calibration) is not trivial. We further find that use of pure inertial sensory information in factor graph based navigation systems is limited, due to the lack of a clear inertial odometry measurement residual or likelihood model. We identify the need for a computationally tractable *inertial odometry* measurement likelihood model for easy integration of multiple sensors in factor graph based methods.

Fig. 4-2 conceptually shows how a smooth trajectory is discretized into discrete poses, and how high-rate inertial information summarized into direct ${}^{b_i}_{b_j}\Delta x$ terms. Our approach follows from the work of Lupton et al. [143], who suggested raw inertial measurements are first integrated and offset compensation is only done later. These directly integrated inertial sensor values are called *inertial preintegrals*.

We emphasize that allocating error to the inertial sensor bias terms is very different from just allocating error to position or orientation states, sometimes also called "bias". By modeling native inertial sensor bias terms, we introduce a mechanism to compensate position and orientation errors along the entire trajectory

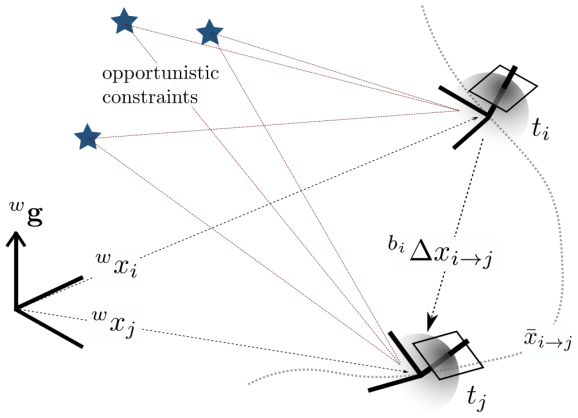


Figure 4-2: Conceptual overview of using pure inertial odometry constraints between world frame poses at time t_i and t_j. Any opportunistic constraint would aid dynamic sensor calibration. Analytical description in Section 4.3.

according to their true sensor measurement errors. Note that by allocating error to inertial sensor biases, we effectively require reintegration of sensor error influences across each affected odometry likelihood model.

In this chapter we improve on the preintegral method by condensing high-rate gyroscope and accelerometer measurements into lossless inertial preintegral terms with a continuous time and exponential parameterization residual model. We describe the accumulation of a second set of values, referred to as inertial odometry compensation gradients, which are simultaneously accumulated at sensor rate. The compensation gradients can then later be used for retroactive estimation of sensor bias terms by means of a residual function, defined in eq. (4.21). We also discuss propagation of the covariance matrix of the inertial odometry terms. The inertial odometry process is illustrated with examples in Section 4.4. Chapter 8 further explores navigation examples with a hand-held inertial with monocular camera localization solution; and concept demonstration on a Boston Dynamics Atlas humanoid robot.

Our work presents a theoretical development of inertial preintegral bias compensation models and best relates to work by [143], [103] and [137]. Our work extends that of [103] and [143] by presenting a continuous time analytical derivation of a Taylor expansion of the sensor error terms manifold inside each odometry likelihood function. To the best of our knowledge, an analytical continuous time gradient model derivation together with higher order Taylor expansion of

the residual function for retroactive calibration is not yet available, and is the main contribution of this chapter.

This chapter is structured as follows: We present a motivation for our work alongside contributions from other authors. Our development begins with familiar interpose odometry constraints and is then extended into the retroactive inertial sensor calibration model. We discuss computation of the required compensation gradients. Finally, validation experiments from synthetic and specific purpose recorded data is discussed at the end of the chapter.

4.2 Approach

Dynamic calibration of inertial measurement unit (IMU) errors is a vital aspect of inertial navigation. To this end, we define the standard gyroscope and accelerometer sensor models [56, 79, 229]. We take the gyroscope measurements $\tilde{\omega}$ as the sum of true rotation ω plus a Gauss-Markov style process for additive bias $\mathbf{b}_\omega$ and additive noise ν_ω:

$$^b\tilde{\omega} = {}^b\omega + \mathbf{b}_\omega + \nu_\omega. \tag{4.1}$$

Similarly, the accelerometer measured force $^b\tilde{\mathbf{f}}$ in the body frame include true acceleration $^b\mathbf{a}$, bias $^b\mathbf{b}_a$, non-inertial gravity in the local level frame $^l\mathbf{g}$ and high frequency noise ν_a

$$^b\tilde{\mathbf{f}} = {}^b\mathbf{a} + {}^b\mathbf{b}_a - {}^b_l\mathbf{R}\,{}^l\mathbf{g}. \tag{4.2}$$

Note, we treat gravity as only vertical, such that the wandering azimuth local and world referenced gravities coincide, i.e. $^l\mathbf{g} = {}^w\mathbf{g}$.

4.2.1 Strapdown inertial computations

The current state-of-the-art relies upon various Kalman filtering architectures to produce an in-situ estimate of sensor errors [56, 79, 229]. Sensor error estimates are often fed back, much like a control loop, to compensate for sensor errors in

real-time before integration is performed:

$$
\begin{aligned}
{}^{w}\mathbf{p}_j &= \iint_{t_i}^{t_j} {}^{w}\mathbf{a}\, d\tau^2 \\
{}^{w}\hat{\mathbf{p}}_j &= \iint_{t_0}^{t_j} \left({}^{w}_{b}\mathbf{R} \left({}^{b}\tilde{\mathbf{f}} - \hat{\mathbf{b}}_a \right) + {}^{w}\mathbf{g} \right) d\tau^2.
\end{aligned} \tag{4.3}
$$

This equation represents a position estimate at time t_j, relative to a starting condition at time t_i.

Filtering solutions in inertial navigation only have one opportunity to compensate estimated sensor biases when producing a naive odometry constraint. Furthermore, use of pre-marginalized odometry constraints from a separate filtering process as odometry relations in a SLAM solution may incorrectly reuse measurements, as was the case in [159]. Access to an independent and pure inertial odometry factor graph measurement likelihood function is desirable.

Instead, we argue that information from each sensor should only contribute to one measurement likelihood in a factor graph, but we lack the models for incorporating pure inertial sensory information in a computationally feasible manner. Inference on factor graphs models can offer robust multi-sensor data fusion, as discussed in Chapters 3 and 5. Pure inertial odometry measurement likelihood models – in a factor graph context – would allow aiding from any other available measurements, without suffering permanent sensor bias errors currently inhibiting filtering style approaches.

4.2.2 Computing preintegrals

Lupton [143] suggested the *preintegral* method where the navigation integral eq. (4.3) is distributed and sensor bias terms are only compensated after inertial measurements have been integrated into the factor graph model,

$$
{}^{b_i}_{b_j}\Delta\hat{\mathbf{p}} = \iint_{t_i}^{t_j} {}^{b_i}_{\tilde{b_\tau}}\mathbf{R}^{b_\tau}\tilde{\mathbf{f}}\, d\tau^2 - \iint_{t_i}^{t_j} {}^{b_i}_{\tilde{b_\tau}}\mathbf{R}^{b_\tau}\mathbf{b}_a\, d\tau^2 + \iint_{t_i}^{t_j} {}^{b_i}_{\tilde{w}}\mathbf{R}^{w}\mathbf{g}\, d\tau^2. \tag{4.4}
$$

The approach rests on the assumption that outside contributions, such as raw sensor bias or gravity, vary much slower than the dynamics of the system and can be efficiently subtracted later using low order integration assumptions.

To make the definition of the *preintegrals* explicit: We accumulate the change in position, velocity and orientation from the i^{th} to the j^{th} pose, as shown in Fig. 4-2.

Accelerations are resolved to the previous pose's body frame, ${}^{b_i}(\cdot)$. The accumulated quantities represent the noisy change in vehicle state, as well as integrated gravity and other effects.

Interpose changes in position, velocity and orientation are computed at sensor rate $t_{k-1} \rightarrow t_k$:

$$
\begin{aligned}
{}^{b_i}_{b_k}\Delta\mathbf{p}^+ &= {}^{b_i}_{b_{k-1}}\Delta\mathbf{p}^+ + \int_{t_{k-1}}^{t_k} {}^{b_i}\Delta v \, d\tau \\
{}^{b_i}_{b_k}\Delta\mathbf{v} &= {}^{b_i}_{b_{k-1}}\Delta\mathbf{v} + \int_{t_{k-1}}^{t_k} {}^{b_i}_{b}\mathbf{R}\,{}^{b}\mathbf{f} \, d\tau \\
{}^{b_i}_{b_k}\mathbf{R} &= {}^{b_i}_{b_{k-1}}\mathbf{R}\,{}^{b_{k-1}}_{b_k}\mathbf{R} = {}^{b_i}_{b_{k-1}}\mathbf{R}\,\exp\left([\Delta\varphi_\times]\right)
\end{aligned}
\tag{4.5}
$$

where $\Delta\varphi \triangleq \int_{t_{k-1}}^{t_k} {}^{b}\omega \, d\tau$. Numerical integration is commonly used to estimate (4.5). Coning, sculling and scrolling compensated multi-rate algorithms, such as those developed by Savage [201, 202], can also be used.

For completeness, we could represent the orientation estimate with quaternions. The body orientation of the i^{th} pose with reference to the world frame ^{w} can be written as ${}^{w}_{b_i}\mathbf{R} = \mathbf{R}\left({}^{w}_{b_i}\mathring{\mathbf{q}}\right)$, which is the conversion from positive scalar quaternion to an equivalent rotation matrix. Orientation propagation is then achieved by:

$$
{}^{b_i}_{b_k}\mathring{\mathbf{p}} = {}^{b_i}_{b_{k-1}}\mathring{\mathbf{p}}\,{}^{b_{k-1}}_{b_k}\mathring{\mathbf{q}} \qquad\qquad \|{}^{b_i}_{b_k}\mathring{\mathbf{p}}\| = 1 \tag{4.6}
$$

where the Hamiltonian quaternion product is assumed, and ${}^{b_i}_{b_k}\mathring{\mathbf{p}}$ is the orientation of the body, at high frequency instant k, relative to the previous (i^{th}) body pose.

4.3 Model Derivation

Two pieces of machinery enable dynamic, efficient, retroactive gyroscope and accelerometer bias compensation: First, we require a measurement preintegration process. Second, we need a mechanism to retroactively compensate sensor error terms at each pose, based on the measurement error model given in Section 4.2. We stress that the same machinery can then also be used for real-time state estimation of platform motion, as shown in Fig. 4-1.

We preemptively note that the inertial odometry process involves accumulating three sets of numbers at sensor rate during the interpose integration period. The

preintegrals, compensation gradients which describe how retroactive compensation is done, and the measurement covariance matrices to correctly weigh the error contributions from error sources in the integration process.

Fig. 4-2 conceptually shows an inertial odometry constraint ${}^{b_i}_{b_j}\Delta x = \iint {}^{w}_{b}\mathbf{R}^{b}\mathbf{a}\, d\tau^2$ between the i^{th} and j^{th} pose positions and is a summary of the true high bandwidth trajectory segment. We note the odometry measurement, ${}^{b_i}_{b_j}\Delta x$, is relative to the previous body pose in the chain and not integrated with respect to a gravity aligned global frame.

Pose definition

We now extend standard SLAM pose vectors to contain not only position and orientation but velocity and inertial sensor bias estimates also. The elements for pose ${}^{w}\mathbf{x}_i$ are therefore:

$$ {}^{w}\mathbf{x}_i = \begin{bmatrix} {}^{w}_{b_i}\mathring{\mathbf{q}} & {}^{b_i}\mathbf{b}_{\omega} & {}^{w}\mathbf{v} & {}^{w}\mathbf{p} & {}^{b_i}\mathbf{b}_{a} \end{bmatrix}^T \tag{4.7} $$

where ${}^{w}_{b_i}\mathring{\mathbf{q}}$ is the orientation quaternion, velocity and position ${}^{w}\mathbf{v}$, ${}^{w}\mathbf{p}$, which are with respect to world frame, ${}^{w}(\cdot)$. World to pose orientations are represented with quaternions to avoid singularities. Gyroscope ${}^{b_i}\mathbf{b}_{\omega}$ and accelerometer ${}^{b_i}\mathbf{b}_{a}$ bias components are maintained with respect to each pose also.

Inertial measurements

We define inertial preintegrals from eq. (4.5) as a vectorized sensor measurement:

$$ {}^{b_i}_{b_j}\Delta\tilde{\mathbf{x}} = \begin{bmatrix} {}^{b_i}_{b_j}\tilde{\varphi} & {}^{b_i}_{b_j}\Delta\tilde{\mathbf{b}}_{\omega} & {}^{b_i}_{b_j}\Delta\tilde{\mathbf{v}} & {}^{b_i}_{b_j}\Delta\tilde{\mathbf{p}}^{+} & {}^{b_i}_{b_j}\Delta\tilde{\mathbf{b}}_{a} \end{bmatrix}^T . \tag{4.8} $$

We take the preintegral measurement with error slack variable ν as

$$ \begin{aligned} \mathbf{z}_{ij} &= {}^{b_i}_{b_j}\Delta\tilde{\mathbf{x}} \\ &= {}^{b_i}_{b_j}\Delta\mathbf{x} + \nu. \end{aligned} \tag{4.9} $$

Note the interpose constraint is defined relative to body pose at time t_i. Probabilistically, our measurement $\mathbf{z}_{ij}$ is a sample from the generating process

$$ \mathbf{z}_{ij} \sim p\left(\mathbf{Z}_{ij} | {}^{w}\mathbf{X}_i, {}^{w}\mathbf{X}_j\right) \propto p\left({}^{w}\mathbf{X}_j | {}^{w}\mathbf{X}_i, \mathbf{Z}_{ij}\right) p\left(\mathbf{X}_i\right). \tag{4.10} $$

The first term expression on the right eases the marginalization operation in a recursive filtering context (hidden Markov model), but conjugate terms obscure the underlying convolution. In reality though, a robot is certainly moved between two distinct points—the expression on the left—while an uncertain odometry measurement is collected between them. We assume no prior pose information is known, making $p(\mathbf{X}_i)$ uniform and only affected by the scale of the expression on the left.

4.3.1 Interpose definitions

We begin our derivation with the standard interpose odometry representation, but vary the standard notation of Lu and Milios [141] slightly, where the body pose ${}^{w}\mathbf{x}_i$ at time t_i is spatially separated from the body pose ${}^{w}\mathbf{x}_j$ at time t_j:

$$\ominus\, {}^{w}\mathbf{x}_i \oplus {}^{w}\mathbf{x}_j = \mathbf{f}_{\ominus}\left({}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j\right), \tag{4.11}$$

We must somehow incorporate the influence of sensor biases and gravity, which have been integrated into the preintegral measurements. We argue intuitively that some function $f_B(\cdot)$ exists, such that f_B captures the measured effects of sensor bias, gravity, and additional error terms. For convenience we collect inertial compensation terms, accelerometer and gyroscope biases as well as accelerometer measured gravity as $B \triangleq \{\mathbf{b}_a, \mathbf{b}_\omega, \mathbf{g}\}$. More error sources can be included in the model as desired.

We propose the to-be-defined composition $(\cdot \circ \cdot)$ of standard interpose distance $f_{\ominus}$ with this *compensation function* f_B. We do not know whether this function has a closed-form expression.

We do not yet have an analytic understanding of how f_B influences the measurement constraint function. An intuitive example of $(f_{\ominus} \circ f_B)$ is, consider if the roll gyroscope had a constant bias offset during pre-integration, the accrued orientation error would produce erroneous gravity coupling into the horizontal channels of the navigation solution. Also see Fig. 4-3 for graphical illustration of preintegral prediction function.

Elaborating on the residual function, the measurement model prediction for preintegral measurements can be written as:

$$\begin{aligned}\delta_{ij}\left({}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j; {}^{l}\mathbf{g}\right) &= \mathbf{z}_{ij} - h\left({}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j\right) \\ &= {}^{b_i}_{b_j}\Delta\tilde{\mathbf{x}} - \left(\mathbf{f}_{\ominus} \circ \mathbf{f}_B\right)\left({}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j\right),\end{aligned} \tag{4.12}$$

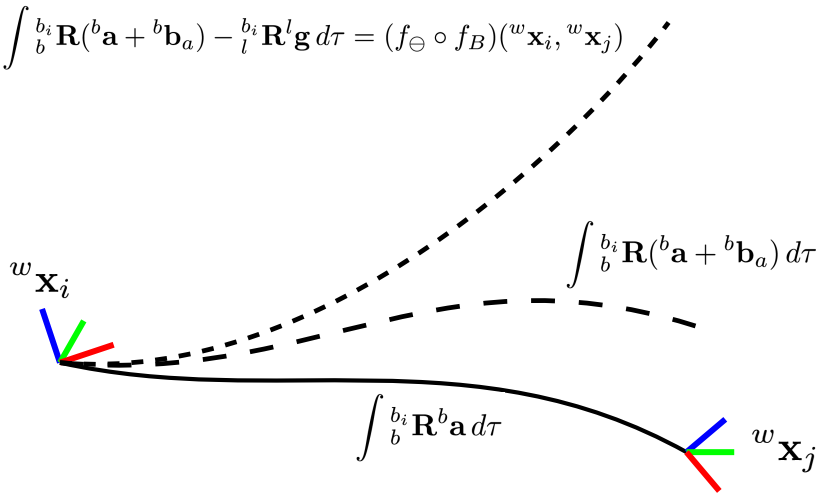


Figure 4-3: Illustration of inertial odometry $(f_\ominus \circ f_B)\,({}^w\mathbf{x}_i, {}^w\mathbf{x}_j)$ computation components. The true interpose distance trajectory between ${}^w\mathbf{x}_i, {}^w\mathbf{x}_j$ is the solid line, where introduction of biases and gravity to the preintegral is shown by the dashed lines.

where δ_{ij} represents the residual between the measured and predicted preintegral.

Measurements from inertial sensors are strongly unimodal, and measurement errors are well modeled with a Gauss-Markov process plus random Gaussian-white noise. We will make the assumption that each preintegral measurement, from a random variable $\mathbf{Z}_{ij}$, is independently sampled from a multivariate normal distribution:

$$\mathbf{z}_{ij} \sim p\left(\mathbf{Z}_{ij} | {}^w\mathbf{X}_i, {}^w\mathbf{X}_j\right) = \mathcal{N}\left(\mu_{ij}, \mathbf{\Sigma}_{ij}\right) \qquad \mu_{ij} = \left(\mathbf{f}_\ominus \circ \mathbf{f}_B\right)\left({}^w\mathbf{x}_i, {}^w\mathbf{x}_j\right). \tag{4.13}$$

We note that each new interpose trajectory segment from an IMU essentially creates a new measurement likelihood model $p\left(\mathbf{Z}_{ij} | {}^w\mathbf{X}_i, {}^w\mathbf{X}_j\right)$, where we only afford a single sample from that likelihood. By assuming the normally distributed structure, we are able to continue our inference development.

However, the complications begin when we try to recover the actual hidden states (odometry dynamics) which form the preintegral likelihood measurement model, $p\left({}^w\mathbf{X}_i, {}^w\mathbf{X}_j | \mathbf{Z}_{ij}\right)$. To illustrate the difficulty, consider that during preintegration the gravity and accelerometer errors are accumulated at various orientations with respect to the previous pose frame $(\cdot)^{b_i}$ and that the orientation estimate itself is in error given gyro biases.

The headache

The main difficulty arises from the non-commutative nature of rotations, $\mathbf{R}_1\mathbf{R}_2 \neq \mathbf{R}_2\mathbf{R}_1$. More explicitly for our case, the pre-integration of noisy gyroscope measurements is a product of exponentials. At each IMU measurement we compute an update in orientation which introduces some orientation error due to gyro bias, $\mathbf{e}^{[\mathbf{b}_\omega \delta t_\times]}$:

$$ {}^{b_i}_{b_k}\mathbf{R} \approx \mathbf{e}^{[\omega \delta t_\times]}\mathbf{e}^{[\mathbf{b}_\omega \delta t_\times]}\mathbf{e}^{[\omega \delta t_\times]}\mathbf{e}^{[\mathbf{b}_\omega \delta t_\times]} \ldots \mathbf{e}^{[\omega \delta t_\times]}\mathbf{e}^{[\mathbf{b}_\omega \delta t_\times]} \tag{4.14} $$

We see individual bias errors are baked into the product that cannot easily be separated out. A further complication arises because the erroneous orientation estimate is used at each IMU measurement epoch to compute velocity and position deltas, as shown in eq. (4.5). From this, we see that predicting the compensation function f_B is not straight forward, and we would need a different approach.

4.3.2 Multidimensional Taylor Expansion

Finding a closed from solution to $(f_\ominus \circ f_B)$ in eq. (4.12) has proven difficult, and we are not aware of any other available in literature. Instead, we will resort to a Taylor expansion and develop its terms in more detail.

To predict the preintegral measurement ${}^{b_i}_{b_j}\Delta\hat{\mathbf{x}}$, consider the partial Taylor series approximation about bias terms in $B'_i = \nabla_{B'}{}^i \left(f_\ominus \circ f_B\right)$ and gravity $G_i = \nabla_G{}^i \left(f_\ominus \circ f_B\right)$ of eq. (4.12). We use a vectorized form of state ζ_{ij} to induce a linear operator framework, and stack terms $B' = \begin{bmatrix}\mathbf{0}_{15} & {}^b\mathbf{b}_\omega & \mathbf{0}_6 & {}^b\mathbf{b}_a & \mathbf{0}_3\end{bmatrix}^T$ and $G = \begin{bmatrix}\mathbf{0}_{27} & {}^w\mathbf{g} & \end{bmatrix}^T$. We also note that higher derivatives are enabled through tensor products $\otimes$:

$$ \begin{aligned} {}^{b_i}_{b_j}\Delta\hat{\mathbf{x}} = \left(\mathbf{f}_\ominus \circ \mathbf{f}_B\right)\left(\zeta_{ij}\right) &\approx B_0 + B'_1 + B'_2 + \ldots + G_\infty \\ B_0 &= \left(\mathbf{f}_\ominus \circ \mathbf{f}_B\right)_{\backslash B_+}\ \zeta_{ij} = \left(f_\ominus\right)\ \zeta_{ij} \\ B'_1 &= \nabla_{B'}\left(\mathbf{f}_\ominus \circ \mathbf{f}_B\right)\ \zeta_{ij} = \left[\frac{\partial\ \left(\mathbf{f}_\ominus \circ \mathbf{f}_B\right)}{\partial\ B'^T}\right]\ \zeta_{ij} \\ B'_2 &= \frac{1}{2}\left(\nabla_{B'}{}^2\left(\mathbf{f}_\ominus \circ \mathbf{f}_B\right)\ \zeta_{ij}^2\right)^\otimes \end{aligned} \tag{4.15} $$

Where $(\cdot)_{\backslash B_+}$ indicates the Taylor expansion around the additive identity of B, that is $\mathbf{b}_a = \mathbf{b}_\omega = \mathbf{0}$ and ${}^w\mathbf{g}$ as the known local gravity acceleration. We separate out

gravity as an infinite partial expansion

$$G_\infty = \sum_{i=1}^{\infty} \nabla_G{}^i \left(f_\ominus \circ f_B\right) \zeta_{ij}{}^i \tag{4.16}$$

We simplify the zeroth terms to a known linear interpose mapping

$$B_0 = \left(\mathbf{f}_\ominus \circ \mathbf{f}_B\right)_{\backslash B_+} \zeta_{ij} = \mathbf{L}\, \zeta_{ij}. \tag{4.17}$$

given a constant gravity assumption, which allows us to write down a simple closed form expression for the gravity component. For convenience, we collapse gravity G_∞ and first order bias B' contributions into a common linear operator:

$$-\mathbf{C}_1\, \zeta_{ij} = B_1{}' + G_\infty. \tag{4.18}$$

The major advantage of interpreting the integral of bias offsets as a Taylor expansion is that the second order Taylor terms follow directly from the derivation for first order terms, which we will see in Section 4.3.6. Note however that higher order terms require tensor products to correctly embed higher dimension operations into a linear map, for example the second order bias compensation term

$$\frac{1}{2}\left(\nabla_B^2 \left(\mathbf{f}_\ominus \circ \mathbf{f}_B\right)\, \zeta_{ij}^2\right)^\otimes = -\frac{1}{2}\mathbf{C}_2^\otimes \zeta_{i\odot j}^2. \tag{4.19}$$

We will see in the next section how many terms in the second order tensor expansion are zero – we find only seven nonzero second order gradients and therefore limit $\mathbf{C}_2^\otimes \zeta_{i\odot j}^2$ to a simplified definition. The second order gradients are organized into a new linear map, but still satisfy eq. (4.15). We elaborate on the tensor product $\otimes$ and element wise product $\odot$ at eq. (4.28).

We populate each of the gradient matrices $\mathbf{L}, \mathbf{C}_1, \mathbf{C}_2^\otimes$ in the next section, but first complete our understanding of the Taylor expansion given above. The Taylor expression given in eq. (4.15) approximates the manifold on which inertial sensor error terms such as biases and gravity operate. With access to the gradient matrices and state vector ζ_{if}, we can retroactively predict the preintegral measurements made by during the process outlined in eq. (4.5):

$$\begin{aligned} {}^{b_i}_{b_j}\Delta\hat{\mathbf{x}} &\approx \left[\mathbf{L} - \mathbf{C}_1\right] \zeta_{ij} \\ &\approx \left[\mathbf{L} - \mathbf{C}_1\right] \zeta_{ij} - \frac{1}{2}\mathbf{C}_2^\otimes \zeta_{ij}^{\otimes 2} \end{aligned} \tag{4.20}$$

We can exploit slow sensor error dynamics to make the post-integration compensation computationally tractable. The slow sensor error dynamics allow the first, second and third order approximation of $(f_{\ominus} \circ f_B)$ to be computationally tractable while remaining reasonably accurate.

To reiterate, during inference we would like to compute the residual δ_{ij} between the noisy interpose preintegral measurement ${}^{b_i}_{b_j}\Delta\tilde{\mathbf{x}}$ and likelihood prediction ${}^{b_i}_{b_j}\Delta\hat{\mathbf{x}}$. Following from eq. (4.12), we can write this residual computation as:

$$\begin{aligned} {}^{b_i}\delta_{ij}\left({}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j, {}^{w}\mathbf{g}\right) &= \left({}^{b_i}_{b_j}\Delta\mathbf{x} + \nu\right) - {}^{b_i}_{b_j}\Delta\hat{\mathbf{x}} \\ &= {}^{b_i}_{b_j}\Delta\tilde{\mathbf{x}} - \left[\mathbf{L} - \mathbf{C}_1\right]\zeta_{ij} \\ &= {}^{b_i}_{b_j}\Delta\tilde{\mathbf{x}} - \left(\left[\mathbf{L} - \mathbf{C}_1\right]\zeta_{ij} - \frac{1}{2}\mathbf{C}_2^{\otimes}\zeta_{ij}^{\otimes 2}\right), \end{aligned} \tag{4.21}$$

where the uncertainty $\nu \sim \mathcal{N}\left(\mathbf{0}, \Sigma_{ij}\right)$ accounts for measurement and computational approximation errors.

At this point, we do not yet have expressions for the gradient matrices $\mathbf{L}, \mathbf{C}_1, \mathbf{C}_2^{\otimes}$, and must therefore find a mechanism to describe them. Fortunately, we do have access to the attitude and velocity error models the inertial mechanization, based on our inertial sensor error models eqs. (4.1) and (4.2). We also have access to the boundary condition for what each of the compensation matrix terms should be at the start of an integration cycle of eq. (4.5).

Therefore, our approach finds the time differential equations for each of the gradient matrices in the Taylor expansion, eq. (4.15), and uses their boundary conditions to compute numerical solutions for $\mathbf{L}, \mathbf{C}_1, \mathbf{C}_2^{\otimes}$. Furthermore, an unlimited number of interpose measurements can be made in this manner, recorded and then used in combination with other outside information to recover the best estimate sensor bias terms retroactively, by using the inertial odometry residual function in a factor graph as depicted in Fig. 4-1.

The next section discusses assembly of the gradient matrix terms and how the ordinary time differential equations are assembled and solved. Finally, in Section 4.3.6 we will derive the required time differential equations for computing the Taylor gradient matrices discussed in eq. (4.15).

4.3.3 Linear Maps

For practical computational appeal, we construct a vector ζ_{ij} as a vectorized function $c\left({}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j; {}^{w}\mathbf{g}\right)$, which maps states from poses ${}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j \in \mathbb{R}^{16}$ and outside in-

fluences, such as gravity ${}^{w}\mathbf{g} \in \mathbb{R}^3$, into a column vector $\zeta_{ij} \in \mathbb{R}_{30}$:

$$c_1 \; : \; \mathbb{R}^{16} \times \mathbb{R}^{16} \times \mathbb{R}^3 \; \rightarrow \; \mathbb{R}^{30}$$

$$\zeta_{ij} = c_1 \left({}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j; {}^{w}\mathbf{g} \right) = \begin{bmatrix} {}^{b_i}_{b_j}\varphi \\ {}^{b_j}\mathbf{b}_{\omega,j} \\ {}^{w}\mathbf{v}_j \\ {}^{w}\mathbf{p}_j \\ {}^{b_j}\mathbf{b}_{a,j} \\ {}^{b_i}\mathbf{b}_{\omega,i} \\ {}^{w}\mathbf{v}_i \\ {}^{w}\mathbf{p}_i \\ {}^{b_i}\mathbf{b}_{a,i} \\ {}^{w}\mathbf{g} \end{bmatrix} . \tag{4.22}$$

The vector ζ_{ij} contains all the information relating to the relative interpose inertial measurement from $i \rightarrow j$. Other inputs may include effects such as earth rotation, Coriolis or transport rate.

The interpose rotation ${}^{b_i}_{b_j}\varphi$ is a vectorized matrix exponentials parameterization of interpose rotation and can be computed by Lie group to Lie algebra logarithm mapping [32]:

$${}^{b_i}_{b_j}\varphi = \left(\log_{\text{SO3}} \left({}^{b_i}_{w}\mathbf{R} \, {}^{w}_{b_j}\mathbf{R} \right) \right)^{\vee} \qquad\qquad \| {}^{b_i}_{b_j}\varphi \| < \pi \tag{4.23}$$

The $log_{SO(3)}$ function and vee operator are defined in Appendix A.

Zeroth Gradient Matrix

Given ζ_{ij}, we can construct a linear transformation $\mathbf{L}$ to operate mapping $\left(\mathbf{f}_{\ominus} \circ \mathbf{f}_B \right)_{\backslash B_+} (\cdot)$, such that the interpose odometry is predicted between poses

${}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j$

$$
\begin{aligned}
&\mathbf{L} \;:\; \mathbb{R}^{30} \;\rightarrow\; \mathbb{R}^{15} \\
&\mathbf{L} = \begin{bmatrix} L_1 & L_2 \end{bmatrix} \\
&L_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}^{b_i}_{w}\mathbf{R} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & {}^{b_i}_{w}\mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
&L_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -{}^{b_i}_{w}\mathbf{R} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -{}^{b_i}_{w}\mathbf{R}\,\Delta t & -{}^{b_i}_{w}\mathbf{R} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.
\end{aligned} \tag{4.24}
$$

First Gradient

Still with eq. (4.15), we construct the first order bias and gravity compensation transform operator $\mathbf{C}_1$, and note that it will depend on a linearization point.

$$\mathbf{C}_1 \;:\; \mathbb{R}^{30} \;\rightarrow\; \mathbb{R}^{15}. \tag{4.25}$$

The matrix-vector product $\mathbf{C}_1\,\zeta_{ij}$ computes the first order dynamic inertial bias compensation term, along with the full gravity compensation G_∞ in the last column.

$$
\begin{aligned}
&\mathbf{C}_1 = \begin{bmatrix} C_{1,1} & C_{1,2} \end{bmatrix} \\
&C_{1,1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \\
&C_{1,2} = \begin{bmatrix} \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{b}_\omega^T} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \frac{\partial \mathbf{v}}{\partial \mathbf{b}_\omega^T} & \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{v}}{\partial \mathbf{b}_a^T} & -{}^{b_i}_{w}\mathbf{R}\Delta t \\ \frac{\partial \mathbf{p}}{\partial \mathbf{b}_\omega^T} & \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{p}}{\partial \mathbf{b}_a^T} & -{}^{b_i}_{w}\mathbf{R}\frac{\Delta t^2}{2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \end{bmatrix}
\end{aligned} \tag{4.26}
$$

The partial derivative gradient terms in C_1 is of particular interest, since we do not yet have expressions which govern them. In Section 4.3.6 we will find time differential equations, and — starting from known boundary conditions — find each of the partial gradient terms numerically at or near sensor rate.

Second Gradient Matrix

We first complete our description of the Taylor expansion terms (gradient matrices), from eq. (4.15). We take the second order gradient matrix as

$$\mathbf{C}_2^{\otimes} \; : \; \mathbb{R}^{18} \; \rightarrow \; \mathbb{R}^{15}. \tag{4.27}$$

which is operates on a vector $\zeta^2_{i \odot j}$:

$$\begin{aligned} c_2 \; : \; \mathbb{R}^{30} \; &\rightarrow \; \mathbb{R}^{18} \\ \zeta^2_{i \odot j} = c_2 \left(\zeta_{ij} \right) &= \begin{bmatrix} {}^{b_j}\mathbf{b}_{\omega,j}^{\odot 2} \\ {}^{b_i}\mathbf{b}_{\omega,i}^{\odot 2} \\ {}^{b_j}\mathbf{b}_{\omega,j} \odot {}^{b_j}\mathbf{b}_{a,j} \\ {}^{b_j}\mathbf{b}_{a,j} \odot {}^{b_j}\mathbf{b}_{\omega,j} \\ {}^{b_i}\mathbf{b}_{\omega,i} \odot {}^{b_i}\mathbf{b}_{a,i} \\ {}^{b_i}\mathbf{b}_{a,i} \odot {}^{b_i}\mathbf{b}_{\omega,i} \end{bmatrix}, \end{aligned} \tag{4.28}$$

where $\mathbf{u}^{\odot 2} = \mathbf{u} \odot \mathbf{u}$ is the Hadamard product and implies element wise multiplication of terms. We note $C_2^{\otimes}$ for compactness only represents non-zero coefficients, unlike C_1. Furthermore, the two cross terms are in fact equivalent, but kept separate here for illustration of the tensor product expansion, ${}^{b_i}\mathbf{b}_{\omega,i} \odot {}^{b_i}\mathbf{b}_{a,i} = {}^{b_i}\mathbf{b}_{a,i} \odot {}^{b_i}\mathbf{b}_{\omega,i}$

We deduce all the second degree terms of gyro and accelerometer biases $\frac{\partial}{\partial \mathbf{b}_\omega{}^T}, \frac{\partial}{\partial \mathbf{b}_a{}^T}$, resulting from the tensor decomposition in eq. (4.15), and provide their derivation in Section 4.3.6. We first provide matrix $\mathbf{C}_2^{\otimes}$ here as an overview:

$$\mathbf{C}_2^{\otimes} = \begin{bmatrix} \mathbf{0} & \frac{\partial^2 \varphi}{\partial \mathbf{b}_\omega^{T\,2}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial^2 \mathbf{v}}{\partial \mathbf{b}_\omega^{T\,2}} & \mathbf{0} & \mathbf{0} & \frac{\partial^2 \mathbf{v}}{\partial \mathbf{b}_\omega^T \mathbf{b}_a^T} & \frac{\partial^2 \mathbf{v}}{\partial \mathbf{b}_a^T \mathbf{b}_\omega^T} \\ \mathbf{0} & \frac{\partial^2 \mathbf{p}}{\partial \mathbf{b}_\omega^{T\,2}} & \mathbf{0} & \mathbf{0} & \frac{\partial^2 \mathbf{p}}{\partial \mathbf{b}_\omega^T \mathbf{b}_a^T} & \frac{\partial^2 \mathbf{p}}{\partial \mathbf{b}_a^T \mathbf{b}_\omega^T} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{4.29}$$

We note that the given Taylor gradient matrices only involve the previous pose

bias error terms. To include the next pose bias terms with the midpoint rule involves halving and duplicating the gradient terms in the space provided for both $\mathbf{b}_i, \mathbf{b}_j$.

We now have a model for inertial preintegral compensation gradients. The two linear transformations $\mathbf{L}, \mathbf{C}_1$ give the first order Taylor approximation in B for eq. (4.15), and can be extended to include second order Taylor terms with $\mathbf{C}_2^{\otimes}\zeta_{i\odot j}^2$. We turn our attention towards high speed accumulation of the inertial compensation gradients in $\mathbf{C}_1, \mathbf{C}_2^{\otimes}$.

4.3.4 Continuous time model

We do not yet have analytical expressions for the compensation gradients present in eqs. (4.26), (4.29), but we can determine a starting boundary condition at the start of a preintegration cycle. We understand that if the preintegral integration occurs over zero time, $\Delta t = t_j - t_i = 0$, then the measurement will be zero, ${}^{b_i}_{b_j}\Delta\mathbf{x} = \mathbf{0}$. Similarly, the Taylor expansion prediction function in eq. (4.15) should then also be zero. Given that the first term $\mathbf{L}\zeta_{ij} = \mathbf{0}$, we know all the compensation gradient terms of $\mathbf{C}_1, \mathbf{C}_2^{\otimes}, \ldots$ should also be zero. Considering bias estimate terms are not zero, we therefore know all the gradient components should also be zero, and thereby determine the boundary condition.

With a boundary condition in hand, we can develop a matrix system of ordinary differential equations (ODE) to calculate each of the compensation gradient matrix terms. In the sections hereafter, we analytically derive each sensor error time differential equation model. We are able to assemble the ODE into a multidimensional, linear time-varying state space system:

$$\begin{aligned} \dot{\xi} &= \mathbf{F}\,\xi + \mathbf{G} \\ \frac{\partial}{\partial t}\begin{bmatrix}\xi_1 \\ \xi_2\end{bmatrix} &= \begin{bmatrix}\mathbf{F}_{1,1} & \mathbf{F}_{1,2} \\ \mathbf{F}_{2,1} & \mathbf{F}_{2,2}\end{bmatrix}\begin{bmatrix}\xi_1 \\ \xi_2\end{bmatrix} + \begin{bmatrix}\mathbf{G}_1 \\ \mathbf{G}_2\end{bmatrix} \end{aligned} \tag{4.30}$$

where $\mathbf{G}$ contains affine terms associated with the time evolution of ξ. Note, $\mathbf{G}$ is not multiplied by an input vector, since this is a coordinate-free representation. We are accumulating the directions in which sensor error terms affect an interpose trajectory segment, and therefore have unit directions $\mathbf{G} \times \mathbf{I}$.

For first order Taylor approximation, we collect the relevant compensation gra-

dients in the matrix $\xi_1 \in \mathbb{R}^{15\times 3}$:

$$\xi_1 = \begin{bmatrix} \frac{\partial^{b_i}_{b}\varphi}{\partial \mathbf{b}_\omega^T} & \frac{\partial \mathbf{v}}{\partial \mathbf{b}_\omega^T} & \frac{\partial \mathbf{v}}{\partial \mathbf{b}_a^T} & \frac{\partial \mathbf{p}}{\partial \mathbf{b}_\omega^T} & \frac{\partial \mathbf{p}}{\partial \mathbf{b}_a^T} \end{bmatrix}^T. \tag{4.31}$$

We also extend the matrix linear system for second order Taylor expansion $\xi_2 \in \mathbb{R}^{21\times 3}$:

$$\xi_2 = \begin{bmatrix} \frac{\partial^2 \varphi}{\partial \mathbf{b}_\omega^{T2}} & \frac{\partial^2 \mathbf{v}}{\partial \mathbf{b}_\omega^{T2}} & \frac{\partial^2 \mathbf{p}}{\partial \mathbf{b}_\omega^{T2}} & \frac{\partial^2 \mathbf{v}}{\partial \mathbf{b}_a^T \mathbf{b}_\omega^T} & \frac{\partial^2 \mathbf{p}}{\partial \mathbf{b}_a^T \mathbf{b}_\omega^T} & \frac{\partial^2 \mathbf{v}}{\partial \mathbf{b}_\omega^T \mathbf{b}_a^T} & \frac{\partial^2 \mathbf{p}}{\partial \mathbf{b}_\omega^T \mathbf{b}_a^T} \end{bmatrix}^T. \tag{4.32}$$

The process outlined above can be used to continue adding the desired number of Taylor expansion terms.

In the next section we discuss integration to a discrete time propagation of ξ, followed by the analytic derivation of the matrices $\mathbf{F}, \mathbf{G}$ thereafter.

4.3.5 Continuous to Discrete

The next important component is our ability to accurately integrate the continuous time matrix systems with a discrete time integration process. In this section we briefly present the trapezoidal method used for all the continuous to discrete integration. Through integration, we propagate both the compensation gradients and the covariance estimate forward at sensor rate k.

We collapse eq. (4.30) into single terms

$$\frac{\partial}{\partial t}\begin{bmatrix} \xi \\ \mathbf{I} \end{bmatrix} = \mathbf{M}\begin{bmatrix} \xi \\ \mathbf{I} \end{bmatrix} \qquad\qquad \mathbf{M}_{(k)} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{4.33}$$

We note the terms in matrix $\mathbf{M}_{(k)}$ depend on various values, including gyro, accelerometer measurements, preintegral rotation estimates ${}^{b_i}_{b_j}\mathbf{R}$, and more. Since some terms are dependent on these terms, we only have $M_{(k)}$ available at each sensor measurement epoch k.

We define the sensor rate time increment $\delta t = t_{k+1} - t_k$ and solve the differential system using the matrix exponential. Either a zeroth order or trapezoidal integration schemes for eq. (4.30) can be used:

$$\begin{aligned} \begin{bmatrix} \xi_{(k+1)} \\ \mathbf{I}_3 \end{bmatrix} &\approx \exp\left[\mathbf{M}_{(k)}\,\delta t\right]\begin{bmatrix} \xi_{(k)} \\ \mathbf{I}_3 \end{bmatrix} \\ \begin{bmatrix} \xi_{(k+1)} \\ \mathbf{I}_3 \end{bmatrix} &\approx \left(\exp\left[-\mathbf{M}_{(k)}\,\tfrac{\delta t}{2}\right]\right)^{-1}\exp\left[\mathbf{M}_{(k+1)}\,\tfrac{\delta t}{2}\right]\begin{bmatrix} \xi_{(k)} \\ \mathbf{I}_3 \end{bmatrix}. \end{aligned} \tag{4.34}$$

We use the Pade' approximation for all matrix exponentials $\exp\left[\cdot\right]$. Lastly, when a new pose is instantiated, we take $\xi_{ij} = \xi_{(k)}$ at time $t_k = t_j$ and reset to the boundary condition $\xi_{(0)} = \mathbf{0}$. The interpose period is the total time $k = 1, 2, .., K$, giving $\Delta t = t_j - t_i = \sum_{k=1}^{K} \delta t$.

4.3.6 Accumulating compensation gradients

Unlike filtering, we do not need accurate bias estimates at integration time. Instead, we knowingly integrate erroneous IMU measurements into their summarized preintegral form (Section 4.2.2) and compensate error terms later. The compensation gradients, discussed in Section 4.3.2, describe the manifold with which we correct measurement errors at inference time. We discuss feed-forward bias compensation in Section 4.5.

For example, orientation error is predominantly caused by gyroscope bias $\mathbf{b}_\omega$ and we will use the first order bias gradient $\frac{\partial \bar{\varphi}}{\partial \mathbf{b}_\omega} \hat{\mathbf{b}}_\omega$ to account for accumulated bias error in the noisy $i \rightarrow j$ rotation preintegral measurement ${}^{b_i}_{b_j}\tilde{\varphi}$:

$$ {}^{b_i}_{b_j}\tilde{\varphi} \approx {}^{b_i}_{b_j}\hat{\varphi} + \frac{\partial^{b_i}_{b} \bar{\varphi}}{\partial \mathbf{b}_\omega{}^T} \hat{\mathbf{b}}_\omega, \tag{4.35} $$

where ${}^{b_i}_{b}\bar{\varphi}$ is the true rotation increment of the body relative to previous pose frame ${}^{b_i}(\cdot)$.

In order to calculate the bias gradients, we require time propagation models of bias influence on each preintegral measurement. This implies we need the partial derivative for each gradient against time. We now derive a differential equation for each bias gradient.

First order Attitude error model

We convert rotation into its matrix exponential parameterization, which is essentially a manifold expression at each pose and allows locally linear manipulation of rotations. We use the closed form Lie algebra to Lie group exponential mapping $\exp_{SO(3)}(\varphi) = \mathbf{e}^{[\varphi\times]}$ given in eq. (A.4).

Take the partial derivative to gyro bias model, see eq. (4.1), with temporary shorthand $[\varphi_\times] = [\varphi]$,

$$ \frac{\partial}{\partial \mathbf{b}_\omega^T} \mathbf{e}^{[\varphi]} = \mathbf{e}^{[\varphi]} \frac{\partial}{\partial \mathbf{b}_\omega^T} \left[\varphi\right]. \tag{4.36} $$

Take partial scalar derivative to time $\frac{\partial}{\partial t}\frac{\partial}{\partial \mathbf{b}_\omega^T}\mathbf{e}^{[\varphi\times]}$

$$\begin{aligned}\frac{\partial}{\partial t}\frac{\partial}{\partial \mathbf{b}_\omega{}^T}\mathbf{e}^{[\varphi]} &= \left(\frac{\partial}{\partial t}\mathbf{e}^{[\varphi]}\right)\frac{\partial}{\partial \mathbf{b}_\omega^T}\left[\varphi\right] + \mathbf{e}^{[\varphi]}\frac{\partial}{\partial \mathbf{b}_\omega^T}\left[\frac{\partial \varphi}{\partial t}\right] \\ &= \mathbf{e}^{[\varphi]}\left(\left[\frac{\partial \varphi}{\partial t}\right]\frac{\partial}{\partial \mathbf{b}_\omega^T}\left[\varphi\right] + \frac{\partial}{\partial \mathbf{b}_\omega^T}\left[\frac{\partial \varphi}{\partial t}\right]\right)\end{aligned} \tag{4.37}$$

Premultiplying by the inverse rotation

$$\mathbf{e}^{[-\varphi]}\frac{\partial}{\partial t}\frac{\partial}{\partial \mathbf{b}_\omega^T}\mathbf{e}^{[\varphi]} = \left[\frac{\partial \varphi}{\partial t}\right]\frac{\partial}{\partial \mathbf{b}_\omega^T}\left[\varphi\right] + \frac{\partial}{\partial \mathbf{b}_\omega^T}\left[\frac{\partial \varphi}{\partial t}\right]. \tag{4.38}$$

Next, we evaluate the vector derivative $\frac{\partial}{\partial \mathbf{b}_\omega}$ by looking at each of the three dimensions in $\mathbf{b}_\omega$ independently. We are going to embed the above expression, which require group operations in $SO\left(3\right)$, into a new linear system with only algebraic operations in the associated algebra $so\left(3\right)$. This enables linear bias compensation as proposed in eq. (4.35).

Recall, the decomposition of skew-symmetric matrices associated with a smooth curve on $SO\left(3\right)$ is

$$\frac{\partial}{\partial \mathbf{b}_\omega^T}\left[\omega\right] = \frac{\partial}{\partial \mathbf{b}_\omega^T}\left(\omega_x\hat{\mathbf{E}}_x + \omega_y\hat{\mathbf{E}}_y + \omega_z\hat{\mathbf{E}}_z\right) \tag{4.39}$$

with usual Lie algebra basis elements $\{\hat{\mathbf{E}}_x, \hat{\mathbf{E}}_y, \hat{\mathbf{E}}_z\}$. We also note, from [32], the Vee operation on basis elements $\left(\hat{\mathbf{E}}_x\right)^\vee = \hat{\mathbf{e}}_1$, where $\hat{\mathbf{e}}_i$ are the natural basis.

We take the vectorized gyroscope bias contribution along $\hat{\mathbf{E}}_x$ of eq. (4.38) as:

$$\left(\mathbf{e}^{[-\varphi]}\frac{\partial}{\partial t}\frac{\partial\, \mathbf{e}^{[\varphi]}}{\partial b_{\omega,x}}\right)^\vee = \left[\frac{\partial \varphi}{\partial t}\right]\begin{bmatrix}\frac{\partial \varphi_x}{\partial b_{\omega,x}} \\ \frac{\partial \varphi_y}{\partial b_{\omega,x}} \\ \frac{\partial \varphi_z}{\partial b_{\omega,x}}\end{bmatrix} + \frac{\partial}{\partial t}\begin{bmatrix}\frac{\partial \varphi_x}{\partial b_{\omega,x}} \\ \frac{\partial \varphi_y}{\partial b_{\omega,x}} \\ \frac{\partial \varphi_z}{\partial b_{\omega,x}}\end{bmatrix} \tag{4.40}$$

Keeping in mind $\partial\omega = -\partial\mathbf{b}_\omega$, accumulation on the local manifold is given by $\Delta\varphi = \int_{t_k}^{t_{k+1}} {}^b\omega\, d\tau$, and that at infinitesimal angles operations on the vector field can commute [32], we find

$$\left(\mathbf{e}^{[-\varphi]}\frac{\partial}{\partial t}\frac{\partial\, \mathbf{e}^{[\varphi]}}{\partial b_{\omega,x}}\right)^\vee = \lim_{\Delta t\to 0} -\left(\mathbf{e}^{[-\Delta\varphi]}\frac{\partial \mathbf{e}^{[\Delta\varphi]}}{\partial \varphi_x}\right)^\vee \tag{4.41}$$

We repeat the vectorization and augment each column to produce a new set of linear equations, one for each of the three rotational degrees of freedom.

$$\mathbf{J}_R = \left[\left(\mathbf{R}^T \frac{\partial \mathbf{R}}{\partial \varphi_x} \right)^\vee, \left(\mathbf{R}^T \frac{\partial \mathbf{R}}{\partial \varphi_y} \right)^\vee, \left(\mathbf{R}^T \frac{\partial \mathbf{R}}{\partial \varphi_z} \right)^\vee \right] \tag{4.42}$$

This is the right differential, or Jacobian, acting as a mapping from the smooth rotation rate manifold into Lie group. Gyroscope bias is coincident with the body frame rotation at each point in time.

We note the Rodriguez formula [33], given in eq. (A.4), can produce a closed form expression for the right differential, as shown in Park [184] in Lemma 2.1:

$$\mathbf{J}_R = \mathbf{I} - \frac{1 - \cos \|\varphi\|}{\|\varphi\|^2} [\varphi_\times] + \frac{\|\varphi\| - \sin \|\varphi\|}{\|\varphi\|^3} [\varphi_\times]^2 \tag{4.43}$$

Finally, we can rewrite eq. (4.38) as a differential equation of bias gradients on a linear manifold at each point in time

$$\lim_{\Delta t \to 0} -\mathbf{J}_R (\Delta \varphi) = \left[{}^b\omega_\times \right] \frac{\partial \varphi}{\partial \mathbf{b}_\omega} + \frac{\partial}{\partial t} \frac{\partial \varphi}{\partial \mathbf{b}_\omega} \tag{4.44}$$

and concludes the first order attitude error model.

Second order Attitude model

As before, there is no direct coupling from accelerometers to attitude. Aside, reverse coupling happens through the velocity error model.

$$\frac{\partial}{\partial t} \frac{\partial^2 {}^{b_i}_{b}\varphi}{\left(\partial \mathbf{b}_a{}^T \right)^2} = \mathbf{0} \tag{4.45}$$

$$\frac{\partial}{\partial t} \frac{\partial^2 {}^{b_i}_{b}\varphi}{\partial \mathbf{b}_a{}^T \partial \mathbf{b}_\omega{}^T} = \mathbf{0} \tag{4.46}$$

$$\frac{\partial}{\partial t} \frac{\partial^2 {}^{b_i}_{b}\varphi}{\partial \mathbf{b}_\omega{}^T \partial \mathbf{b}_a{}^T} = \mathbf{0} \tag{4.47}$$

The second order gyro bias term, together with eq. (4.1) we have $\frac{\partial}{\partial \mathbf{b}_\omega{}^T} \left[{}^b\omega_\times \right] =$

$-(-\mathbf{I})$:

$$\frac{\partial}{\partial t}\frac{\partial^2 {}^{b_i}_{b}\varphi}{\left(\partial \mathbf{b}_\omega{}^T\right)^2} = \frac{\partial}{\partial \mathbf{b}_\omega{}^T}\left[{}^{b}\omega_\times\right]\frac{\partial^{b_i}_{b}\varphi}{\partial \mathbf{b}_\omega{}^T} - \frac{\partial \mathbf{J}_R}{\partial \mathbf{b}_\omega{}^T}$$
$$\approx \frac{\partial {}^{b_i}_{b}\varphi}{\partial \mathbf{b}_\omega{}^T} + \left[{}^{b}\omega_\times\right]\frac{\partial^2 {}^{b_i}_{b}\varphi}{\left(\partial \mathbf{b}_\omega{}^T\right)^2} \tag{4.48}$$

First order Velocity error model

The measured acceleration in the previous pose's (i^{th}) frame is given by:

$${}^{b_i}\dot{\mathbf{v}} = {}^{b_i}_{b}\mathbf{R}\left({}^{b}\tilde{\mathbf{f}} - \mathbf{b}_a\right) + {}^{b_i}_{w}\mathbf{R}\,{}^{w}\mathbf{g} \tag{4.49}$$

where true force is equal to measured force minus accelerometer bias ${}^{b}\mathbf{f} = {}^{b}\tilde{\mathbf{f}} - \mathbf{b}_a$. We take true body force as body acceleration minus gravity, which is a noninertial force, ${}^{b}\mathbf{f} = {}^{b}\mathbf{a} - {}^{b}\mathbf{g}$.

We find velocity dependence on gyroscope bias with the partial derivative to gyro bias, and by using the relation given by eq. (A.4)

$$\begin{aligned}\frac{\partial}{\partial \mathbf{b}_\omega^T}{}^{b_i}\dot{\mathbf{v}} &= \frac{\partial}{\partial \mathbf{b}_\omega^T}\mathbf{e}^{\left[{}^{b_i}_{b}\varphi_\times\right]}\left({}^{b}\tilde{\mathbf{f}} - \mathbf{b}_a\right)\\ &= \mathbf{e}^{\left[{}^{b_i}_{b}\varphi_\times\right]}\frac{\partial}{\partial \mathbf{b}_\omega^T}\left[{}^{b_i}_{b}\varphi_\times\right]\left({}^{b}\tilde{\mathbf{f}} - \mathbf{b}_a\right)\\ &= \mathbf{e}^{\left[{}^{b_i}_{b}\varphi_\times\right]}\frac{\partial}{\partial \mathbf{b}_\omega^T}\left(-\left[{}^{b}\tilde{\mathbf{f}}_\times\right]{}^{b_i}_{b}\varphi + [\mathbf{b}_{a\times}]\,{}^{b_i}_{b}\varphi\right)\end{aligned} \tag{4.50}$$

and since $\frac{\partial \mathbf{f}}{\partial \mathbf{b}_\omega} = \mathbf{0}$, we have

$$\frac{\partial}{\partial t}\frac{\partial}{\partial \mathbf{b}_\omega^T}{}^{b_i}\mathbf{v} = \mathbf{e}^{\left[{}^{b_i}_{b}\varphi_\times\right]}\left([\mathbf{b}_{a\times}] - \left[{}^{b}\tilde{\mathbf{f}}_\times\right]\right)\frac{\partial {}^{b_i}_{b}\varphi}{\partial \mathbf{b}_\omega^T}. \tag{4.51}$$

However, we do not have access to an accelerometer bias estimate during the pre-integration and must therefore ignore the term $[\mathbf{b}_a\times]$.

$$\frac{\partial}{\partial t}\frac{\partial^{b_i}\mathbf{v}}{\partial \mathbf{b}_\omega^T} \approx -{}^{b_i}_{b}\mathbf{R}\left[{}^{b}\tilde{\mathbf{f}}_\times\right]\frac{\partial^{b_i}_{b}\varphi}{\partial \mathbf{b}_\omega^T} \tag{4.52}$$

Although, we do note that if bias estimates are accumulated and compensated at

the sensor, the error introduced by this assumption decreases.

Next, we find velocity dependence on accelerometer bias directly through the second partial derivative:

$$\frac{\partial}{\partial \mathbf{b}_a^T}\frac{\partial}{\partial t}{}^{b_i}\mathbf{v} = -{}^{b_i}_{b}\mathbf{R} \tag{4.53}$$

Second order Velocity model

The second order terms for velocity are:

$$\frac{\partial}{\partial t}\frac{\partial^2 {}^{b_i}\mathbf{v}}{\left(\partial \mathbf{b}_a{}^T\right)^2} = \mathbf{0} \tag{4.54}$$

$$\frac{\partial}{\partial t}\frac{\partial^2 {}^{b_i}\mathbf{v}}{\partial \mathbf{b}_\omega{}^T \partial \mathbf{b}_a{}^T} = -\mathbf{e}^{[\varphi]}\frac{\partial \varphi}{\partial \mathbf{b}_\omega{}^T} \tag{4.55}$$

and

$$\begin{aligned}
\frac{\partial}{\partial t}\frac{\partial^2 {}^{b_i}\mathbf{v}}{\left(\partial \mathbf{b}_\omega{}^T\right)^2} &= -\mathbf{e}^{[\varphi]}\left(\frac{\partial}{\partial \mathbf{b}_\omega{}^T}\left[\varphi_\times\right]\right)\left[{}^b\mathbf{f}_\times\right]\frac{\partial \varphi}{\partial \mathbf{b}_\omega{}^T} - \mathbf{e}^{[\varphi]}\left[{}^b\mathbf{f}_\times\right]\frac{\partial^2 \varphi}{\left(\partial \mathbf{b}_\omega{}^T\right)^2} \\
&= -\mathbf{e}^{[\varphi]}\frac{\partial \varphi}{\partial \mathbf{b}_\omega{}^T}\left[{}^b\mathbf{f}_\times\right]\frac{\partial \varphi}{\partial \mathbf{b}_\omega{}^T} - \mathbf{e}^{[\varphi]}\left[{}^b\mathbf{f}_\times\right]\frac{\partial^2 \varphi}{\left(\partial \mathbf{b}_\omega{}^T\right)^2} \\
&\approx -\mathbf{e}^{[\varphi]}\frac{\partial \varphi^{(k-1)}}{\partial \mathbf{b}_\omega{}^T}\left[{}^b\tilde{\mathbf{f}}_\times\right]\frac{\partial \varphi}{\partial \mathbf{b}_\omega{}^T} - \mathbf{e}^{[\varphi]}\left[{}^b\tilde{\mathbf{f}}_\times\right]\frac{\partial^2 \varphi}{\left(\partial \mathbf{b}_\omega{}^T\right)^2}
\end{aligned} \tag{4.56}$$

We note the first term is a higher degree, as well as implicit in $\frac{\partial \varphi}{\partial \mathbf{b}_\omega{}^T}$. When practically computing, we start with the uncompensated accelerometer measurement ${}^b\tilde{\mathbf{f}} = {}^b\mathbf{f} + {}^b\mathbf{b}_a$ until better bias estimates become available. We also break the implicit expression by using the previous sensor time-step gradient $\frac{\partial \varphi^{(k-1)}}{\partial \mathbf{b}_\omega{}^T}$, but maintain the implicit equation for continuous time interpretation.

The second order velocity error to accelerometer bias is obtained with the partial derivative, again using $\frac{\partial}{\partial \mathbf{b}_a{}^T}\left[{}^b\mathbf{f}_\times\right] = -\left(-\mathbf{I}\right)$:

$$\begin{aligned}
\frac{\partial}{\partial t}\frac{\partial^2 {}^{b_i}\mathbf{v}}{\partial \mathbf{b}_a{}^T \partial \mathbf{b}_\omega{}^T} &= -\mathbf{e}^{[\varphi]}\frac{\partial}{\partial \mathbf{b}_a{}^T}\left[{}^b\mathbf{f}_\times\right]\frac{\partial \varphi}{\partial \mathbf{b}_\omega{}^T} \\
&= -{}^{b_i}_{b}\mathbf{R}\frac{\partial \varphi}{\partial \mathbf{b}_\omega{}^T}
\end{aligned} \tag{4.57}$$

First order Position models

We directly know relations for positions:

$$\frac{\partial}{\partial \mathbf{b}_\omega^T}\frac{\partial}{\partial t}{}^{b_i}\mathbf{p} = \frac{\partial\, {}^{b_i}\mathbf{v}}{\partial \mathbf{b}_\omega^T} \quad \text{and} \qquad \frac{\partial}{\partial \mathbf{b}_a^T}\frac{\partial}{\partial t}{}^{b_i}\mathbf{p} = \frac{\partial\, {}^{b_i}\mathbf{v}}{\partial \mathbf{b}_a^T} \tag{4.58}$$

This concludes the derivation of the inertial error models. We proceed to develop a computational framework for computing matrix $\mathbf{C}_1$.

Second order Position models

The second gyro bias derivative gives

$$\frac{\partial}{\partial t}\frac{\partial^2\, {}^{b_i}\mathbf{p}}{\left(\partial {\mathbf{b}_\omega}^T\right)^2} = \frac{\partial^2\, {}^{b_i}\mathbf{v}}{\left(\partial {\mathbf{b}_\omega}^T\right)^2} \tag{4.59}$$

$$\frac{\partial}{\partial t}\frac{\partial^2\, {}^{b_i}\mathbf{p}}{\partial {\mathbf{b}_a}^T \partial {\mathbf{b}_\omega}^T} = \frac{\partial^2\, {}^{b_i}\mathbf{v}}{\partial {\mathbf{b}_a}^T \partial {\mathbf{b}_\omega}^T}. \tag{4.60}$$

While the second accelerometer bias derivative gives

$$\frac{\partial}{\partial t}\frac{\partial^2\, {}^{b_i}\mathbf{p}}{\left(\partial {\mathbf{b}_a}^T\right)^2} = \frac{\partial^2\, {}^{b_i}\mathbf{v}}{\left(\partial {\mathbf{b}_a}^T\right)^2} = \mathbf{0} \tag{4.61}$$

$$\frac{\partial}{\partial t}\frac{\partial^2\, {}^{b_i}\mathbf{p}}{\partial {\mathbf{b}_\omega}^T \partial {\mathbf{b}_a}^T} = \frac{\partial^2\, {}^{b_i}\mathbf{v}}{\partial {\mathbf{b}_\omega}^T \partial {\mathbf{b}_a}^T}. \tag{4.62}$$

This concludes the second order derivatives required for the second order time differential equation needed for the Taylor expansion in eq. (4.15).

4.3.7 Defining F and G

We have derived an entirely continuous time inertial sensor error model for the inertial odometry residual function, to be used as a measurement likelihood in a factor graph framework. Recall the continuous time linear system equations established in Section 4.3.4 and continuous to discrete solution presented in Section 4.3.5. In this section we populate the matrices $\mathbf{F}, \mathbf{G}$ with the derived time differential models required for computing the compensation gradient terms in eq. (4.30).

The first order only system matrix is

$$\mathbf{M}_1 = \left[\begin{array}{c|c} \mathbf{F}_1 & \mathbf{G}_1 \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right]$$

$$= \left[\begin{array}{ccccc|c} -\left[{}^b\omega_\times\right] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{J}_R\left({}^b d\varphi\right) \\ -{}^{b_i}_b\mathbf{R}\left[{}^b\mathbf{f}_\times\right] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -{}^{b_i}_b\mathbf{R} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right]. \tag{4.63}$$

We can also define the second order matrix terms as, see eq. (4.30), $\mathbf{F}_{1,2} = \mathbf{0}_{21\times 15}$ and

$$\mathbf{F}_{2,1} \approx \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 18} \\ -{}^{b_i}_b\mathbf{R}\frac{\partial\varphi^{(k-1)}}{\partial\mathbf{b}_\omega{}^T}\left[{}^b\mathbf{f}_\times\right] & \mathbf{0}_{3\times 18} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 18} \\ -{}^{b_i}_b\mathbf{R} & \mathbf{0}_{3\times 18} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 18} \\ -{}^{b_i}_b\mathbf{R} & \mathbf{0}_{3\times 18} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 18} \end{bmatrix} \quad \mathbf{F}_{2,2} = \begin{bmatrix} \left[{}^b\omega_\times\right] & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 15} \\ -{}^{b_i}_b\mathbf{R}\left[{}^b\mathbf{f}_\times\right] & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 15} \\ \mathbf{0}_{3\times 6} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 12} \\ & \mathbf{0}_{3\times 21} & \\ \mathbf{0}_{3\times 9} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 9} \\ & \mathbf{0}_{3\times 21} & \\ \mathbf{0}_{3\times 15} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix} \tag{4.64}$$

This concludes our definition of the continuous time compensation gradients model. To reiterate, we integrate this model as discussed in Section 4.3.5, and use the result to construct the Taylor gradient matrix terms discussed in Section 4.3.2.

We now turn our attention to the error covariance computed during preintegration.

4.3.8 Covariance propagation

In eq. (4.5) we described how to collect a preintegral measurement. In this section we look at the accumulation of measurement uncertainty. The measurement covariance is the third of three sets of values accumulated at sensor rate, along with preintegrals and compensation gradients.

We take the true error covariance as from the residual function defined in

eq. (4.21),

$$\begin{aligned}\delta\mathbf{z}_{ij} &= \tilde{\mathbf{z}}_{ij} - \hat{\mathbf{z}}_{ij} \\ &= \begin{bmatrix} {}^{b_i}\delta\varphi & {}^{b_i}\delta\mathbf{v} & {}^{b_i}\delta\mathbf{p} & {}^{b_i}\delta\mathbf{b}_\omega & {}^{b_i}\delta\mathbf{b}_a \end{bmatrix}^T . \end{aligned} \tag{4.65}$$

Producing the measurement covariance

$$\mathbb{E}\left[[\delta\mathbf{Z}_{ij}]^T [\delta\mathbf{Z}_{ij}] \right] . \tag{4.66}$$

Using either the continuous or discrete time Ricatti equation definitions — similar to covariance propagation in Kalman filtering —, we write the progression of uncertainty as

$$\begin{aligned}\mathbf{P}_{k+1} &= \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k \\ \Psi_k &= \left[\begin{array}{c|c} -\mathbf{F}_k & \mathbf{G}_k\mathbf{Q}\mathbf{G}_k{}^T \\ \hline \mathbf{0} & \mathbf{F}_k{}^T \end{array}\right] \\ \Upsilon_k &= \left[\begin{array}{c|c} * & \mathbf{\Phi}_k{}^{-1}\mathbf{Q}_k \\ \hline \mathbf{0} & \mathbf{\Phi}_k{}^T \end{array}\right] \end{aligned} \tag{4.67}$$

where we compute the discrete noise matrix based on standard INS error accumulation models [56]. Integration of the elements in Ψ_k is done with zeroth or trapezoidal rule, described in Section 4.3.5. The discrete time integrated result $\mathbf{\Upsilon}_k$ is influenced by the integrated process noise matrix $\mathbf{Q}_k = \mathbf{\Phi}_k \left(\mathbf{\Phi}_k{}^{-1}\mathbf{Q}_k \right)$.

$$\mathbf{Q} = \kappa \times \begin{bmatrix} \Theta_A \, \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Theta_V \, \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Theta_{BI,\omega} \, \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Theta_{BI,a} \, \mathbf{I}_3 \end{bmatrix} \tag{4.68}$$

The continuous process noise matrix, $\mathbf{Q}$, is defined by sensor noise characteristics [244]. We use a scaling factor, κ, as a safety factor to ensure that we allocate sufficient uncertainty to accommodate measurement noise and computational errors.

This approach clearly does not accurately model colored noise, but is common practice in the inertial navigation. Increased process noise $\mathbf{Q}$ only reduces the inertial odometry weight in the SLAM solution. The typical range for κ is around two to ten.

Allan variance testing [50] may be used to estimate each of these noise parame-

ters. Spectral densities, Θ_*, are generally supplied with the inertial sensor package and commonly related to $1\ \sigma$ deviations according to [244]:

$$\sigma_* = \sqrt{\frac{\Theta_*}{T}}. \tag{4.69}$$

We refer the reader to yet more references at [229] for more details on uncertainty growth in inertial navigation systems.

4.3.9 Compensating earth rotation rate, and Gyrocompassing

We can extend the attitude and velocity error models to compensate for earth rotation. This would enable factor graph style gyrocompassing, and direct the reader to the authors earlier work [68]. The preintegral accumulation stays unchanged, but would use updated compensation terms in eq. (4.30). The development starts with the updated attitude propagation formulation:

$$\frac{\partial}{\partial t}\mathbf{e}^{\left[{}^{n}_{b}\varphi_{\times}\right]} = \mathbf{e}^{\left[{}^{n}_{n'}\omega_{\times}\right]}\mathbf{e}^{\left[{}^{n'}_{b'}\varphi_{\times}\right]}\mathbf{e}^{\left[{}^{b'}_{b}\omega_{\times}\right]} \tag{4.70}$$

The uncertainty covariance propagation will be affected.

4.3.10 Observability

We will just highlight a few key points of inertial sensor bias observability from great works by [92, 107, 121, 147, 153, 193]. Jones et al. [107] used indistinguishable trajectories to analyze observability properties of an visual-inertial sensor pairing. Mirzaei et al. [153] used Lie derivatives for a similar purpose.

Mirzaei [153] avoids estimation of the gravity vector, and shows there are cases where not all parameters in the system are observable. Accelerometer biases can only be estimated if subjected to varying motion, while being constrained by aiding information. Jones' [107] observability analysis considers the simultaneous estimation of the local gravity vector. This work notes degenerate cases between accelerations and the gravity estimates, and appear when the system is subject to insufficient excitation dynamics. More specifically, Jones states that gravity can be estimated when subjected to varying accelerations and sufficient features are sighted nearby the camera.

Kelly et al. [120, 121, 227] published a detailed Lie derivative nonlinear observability analysis for visual-inertial systems. Kelly showed that with sufficient ex-

citation of the visual-inertial sensor pair, absolute scene scale; the local gravity vector; the IMU biases; and 6DOF transform between the camera and IMU are all simultaneously observable. More specifically, this result is shown to hold while the system is simultaneously rotated about two axes while measuring non-zero acceleration (excluding gravity) on at least two dimensions.

4.4 Validation and Analysis

The best navigation systems are able to dynamically calibrate the IMU from aiding information. The examples that follow intend to illustrate retroactive sensor calibration, which is distinct from Kalman filter style IMU calibration, and the premise that this new inertial odometry technique is more powerful and capable than existing Kalman filter techniques.

4.4.1 Synthetic Examples

Alongside the inertial preintegral measurements eq. (4.5), we use the continuous analytic model in eq. (4.30) with trapezoidal integration to discrete time for computing compensation gradients. The compensation gradients are used to assemble the matrix $\mathbf{C}_1$, as shown by eq. (4.26). As shown in eq. (4.21), the weighted residual is the difference between measured preintegrals and prediction $(f_{\ominus} \circ f_B)\left({}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j\right)$, which is computed via compensation gradients.

We can form the residual in eq. (4.21) to construct a residual function fit for inference and bias recovery. Crucially, sensor bias terms are inferred after raw sensor integration has completed, unlike filtering where sensor bias terms are first estimated and subtracted before integration. We also note the uncertainty estimates, or preintegral covariance shown by eq. (4.67), are also accumulated at sensor rate.

4.4.2 Inference by nonlinear-least-squares

A variety of inference techniques exist. Non-linear least-squares optimization is used for the examples presented in this section, to demonstrate how preintegrals and compensation gradients achieve retroactive bias recovery:

$$\underset{b_\omega,\, b_a}{\operatorname{argmin}} \quad \left({\delta_{i\to j}}^{T}\ \Sigma_{i\to j}^{-1}\ \delta_{i\to j}\right). \tag{4.71}$$

Where the preintegral covariance $\Sigma_{i \to j}$ is propagated according to the uncertainty model presented in eq. (4.67).

Basic averaging, not moving

Consider an IMU standing stationary for duration of time. Naming the IMU start and end position poses as ${}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j$ respectively. A pose is taken to represent the position and orientation relative to some world frame. Dynamic sensor calibration can be obtained from integrated gyroscope ${}^{b}\tilde{\omega}$ and accelerometer ${}^{b}\tilde{\mathbf{f}}$ measurements. Assume IMU measurements are integrated according to the preintegral strategy in eq. (4.5) between time t_i and t_j.

$$ {}^{b_i}\Delta v_{i \to j} + \int_{t_i}^{t_j} {}^{b}\mathbf{g} d\tau + \boldsymbol{\Delta} = \int_{t_i}^{t_j} {}^{b}\tilde{\mathbf{f}} d\tau \tag{4.72} $$

The integration should show natural effects such as gravity ${}^{b}\mathbf{g}$ and earth rotation (omitted), as well as integrated errors Δ. By knowing ${}^{b_i}\Delta v_{i \to j} = \mathbf{0}$ as a form of aiding information, we can take the integrated sensor values as a summary of all the errors $\boldsymbol{\Delta}$. In this case we use knowledge about both poses ${}^{w}\mathbf{x}_i, {}^{w}\mathbf{x}_j$, can be used compare the IMU preintegrals with the expected pose positions. Note that, in general aiding information may include any information pertaining to any the pose states, depicted in eq. (4.7). The factor graph methodology will allow us to reason over all available data in a centralized framework. In this basic example, compensation gradients did not seem necessary, since nothing happened. The examples that follow build in complexity until the need for compensation gradients is clear.

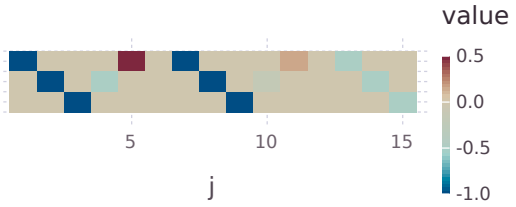
Perfect sensor not moving

Consider an ideal IMU in a resting position with XYZ axes pointing North-West-Up (NWU) in the presence of gravity, and temporarily omitting earth rotation rate. We expect to see rotation rate measurements ${}^{b}\omega = [0, 0, 0]^T$ and accelerations ${}^{b}f = [0, 0, g_e]$. Where the transpose is indicated by ^{T} and earth's gravity is illustrated as $g_e = 1\ m/s^2$ in these examples. Let the timer run for $1\ second$ and find measured preintegral values (according to eq. (4.5))

$$ {}^{b_i}\Delta\tilde{\mathbf{x}}_{i \to j} = \begin{bmatrix} {}^{b_i}_{b_j}\tilde{\varphi} & \Delta\tilde{\mathbf{b}}_{\omega} & \Delta\tilde{\mathbf{v}}_{i \to j} & \Delta\tilde{\mathbf{p}}_{i \to j}{}^{+} & \Delta\tilde{\mathbf{b}}_{a} \end{bmatrix}^T \tag{4.73} $$

$$ = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & 0 & 0 & 10 & 0 & 0 & 5 & \mathbf{0}_3 \end{bmatrix} \tag{4.74} $$

The transpose of compensation gradients vector $\xi_{i\to j} = \int_{t_i}^{t_j} \dot{\xi}\, d\tau$ as presented by eq. (4.31), for this example is depicted below



Solving the resulting least squares minimization problem, we find:

$$\mathbf{b} = \begin{bmatrix}\mathbf{b}_\omega & \mathbf{v}_a\end{bmatrix} = \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3\end{bmatrix} \tag{4.75}$$

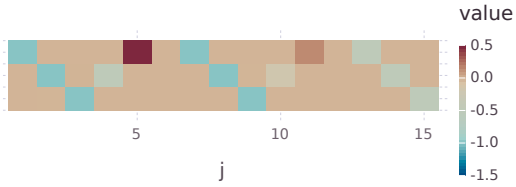
Since the residual, eq. (4.21), balances to zero any change in pose x_i's bias estimates will result in an in-balance and non-zero residual.

Basic gyroscope bias, not moving

Now consider the same example with a stationary NWU facing IMU , but with a slight bias of $0.001\ rad/s$ on the x-axis gyroscope. During preintegration of sensor measurements, eq. (4.5), we find the orientation estimate ${}^{b_i}_{b_j}\mathbf{R}$ slowly rolling about the x-axis. As a result, the term ${}^{b_i}_{b}\mathbf{R}\,{}^{b}\mathbf{f}$ projects the measured gravity from accelerometer measurement ${}^{b}\mathbf{f} = [0, 0, 1]^T$ into the y-axis of the pose-i reference frame. This projected acceleration measurement results in velocity along the y-axis. Therefore the preintegral measurement after $1\ second$ of integration is

$${}^{b_i}\Delta\tilde{\mathbf{x}}_{i\to j} \approx \begin{bmatrix}0.001 & \mathbf{0}_5 & 0 & -0.0004995 & 1 & 0 & -0.000166417 & 0.5 & \mathbf{0}_3\end{bmatrix} \tag{4.76}$$

The associated compensation gradient vector, ${\xi_{i\to j}}^T$ is depicted in the figure below. This vector has numerical differences from the perfect IMU example above.



Again, at some later time the preintegrals, compensation gradients and error covariances are used – along with the constraint information that pose ${}^b\mathbf{x}_i, {}^b\mathbf{x}_j$ are coincident – to retroactively infer the sensor bias estimates. Using the non-linear least squares optimization technique with covariance weighted residual, we find:

```
residual= 0.0 after 308 iterations (returned XTOL_REACHED)
Truth: [0.001,0.0,0.0,0.0,0.0,0.0]
Estim: [0.001,0.0,0.0,-1.0e-6,3.0e-6,1.0e-6]
error: [0.04,-2.64,-0.01,7.24,-30.6,-14.79] %
```

The above text shows computer output for true $\mathbf{b}$ and estimated $\hat{\mathbf{b}}$ gyro and accelerometer biases. The error output is used to show the percentage accuracy, computed according to difference between true and estimated bias values, stabilized by $\epsilon = 1e-5$ to prevent division by zero:

$$\text{percentage error} = 100 \times \frac{b - \hat{b}}{|b| + \epsilon}. \tag{4.77}$$

Note, however, division of relatively larger numerators by ϵ still occurs in the examples that follow, resulting in seeming large errors. These fictitious errors should be read in context of what true sensor performance would allow.

Basic accelerometer bias, not moving

We can repeat the experiment for a y-axis accelerometer bias error of $0.05\ m/s^2$, with accompanying preintegrals:

$${}^{b_i}\Delta x_{i\to j} = \begin{bmatrix}\mathbf{0}_7 & 0.05 & 1 & 0 & 0.025 & 0.5 & \mathbf{0}_3\end{bmatrix}^T \tag{4.78}$$

Computing the minimization, we find:

```
residual= 0.24 after 274 iterations (returned SUCCESS)
Truth: [0.0,0.0,0.0,0.0,0.05,0.0]
Estim: [-8.9e-5,-0.0,-0.0,-0.0,0.049567,2.0e-6]
error: [894.15,0.05,0.0,0.24,0.87,-21.47] %
```

Firstly note the y-axis accelerometer bias is estimated within less than $1\ \%$ error. The large error percentage for x-axis gyroscope is a consequence of noise models and expected performance from the IMU. The estimated value of $-8.9e-5\ rad/s$ is small relative to the sensor characteristics used for this example. The seemingly

large percentage value $894.15\ \%$ is also subject to our choice of stabilizing number ϵ, and more an artifact of how we compute the error.

An important point must be made here, that the error characteristics of the IMU are a vital part of recovering sensor bias estimates. The sensor error covariance from a perintegration period, $\Sigma_{i \to j}$, represents our certainty in each of the preintegral numbers. In this example, the seemingly large percentage error for the x-axis gyroscope is a consequence of the sensor performance characteristics specified in eq. (4.68). The bias estimate of $-8.9e-5\ rad/s$ is well below the specified noise performance used to calculate $\Sigma_{i \to j}$ in this example. Modifying the noise parameters will accordingly modify the scale of bias estimates. Separately, the wide frequency noise performance depends on the quality of the inertial sensors.

In general, we enforce a notion of smoothness in inertial biases across consecutive interpose inertial odometry constraints, owing to the Gauss-Markov assumption in eqs. (4.1) and (4.2). The error in x-gyroscope bias estimate we see here will result in increased residual errors in neighboring interpose constraints and will be constrained through a combination of multiple observations over time. This argument is similar to observability arguments made for Kalman filter style inertial navigation solutions. Observability is discussed further in Section 4.3.10.

Concurrent gyroscope and accelerometer bias errors, not moving

In the more general case, we expect some variation in all sensor biases. Here, we illustrate a similar example with the IMU standing still facing NWU over a $1\ second$ period in an environment with gravity ${}^{w}g_e = [0, 0, 1]^T\ m/s^2$, but allow a random bias on all sensor axis. Let

$$\mathbf{b} = [0.0013, 0.0015, -0.0005, -0.0004, -0.0015, 0.0003]^T \tag{4.79}$$

We find the preintegrals

$$\begin{aligned} {}^{b_i}\Delta\tilde{\mathbf{x}}_{i \to j} \approx [&0.0013, 0.0015, -0.0004, \mathbf{0}_3, 0.00035, -0.00215, & (4.80)\\ &1.0003, 5.0e-5, -0.00097, 0.50015, \mathbf{0}_3]^T & (4.81) \end{aligned}$$

At this point we note that changes in bias terms $\Delta b_\omega, \Delta b_a$ are not yet used in these examples, and seen as the two $\mathbf{0}_3$ elements in the preintegral measurement vector given above.

Using the compensation gradients $\xi_{i \to j}$ to again construct the residual function, eq. (4.21), we can retroactively search for sensor bias estimates as before:

```
residual= 0.001 after 234 iterations (returned XTOL_REACHED)
Truth: [0.0013,0.0015,-0.0005,-0.0004,-0.0015,0.0003]
Estim: [0.00128,0.001459,-0.000504,-0.000419,-0.001531,0.000309]
error: [-0.14,0.06,-0.12,0.22,-1.1,-1.9] %
```

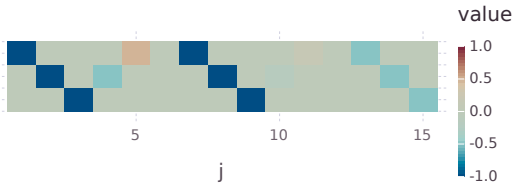
All the bias errors are recovered within 1 or 2 $\%$ of the true values. Imagine that the errors baked into the preintegral measurement vector comes from a randomly drifting and gravity coupled integral estimate. The compensation gradients capture the directions in which errors would propagate, and project them back into the previous pose ${}^{w}\mathbf{x}_i$ reference frame ${}^{b_i}(\cdot)$. The inference process can then use this summary of directions, i.e. the compensation gradients, to predict the errors integrated into the preintegral measurements.

Bias estimates while moving, the power of compensation gradients

All the above examples can easily be reproduced by simple averaging of the measurements, since we know the IMU did not move during the integration period. Now consider an example in which the integration interval the IMU is flipped onto its side for half of the 1 $second$ integration period, after which the IMU returned to a NWU orientation before the integration period expires. Averaging alone would not know how much time to spend in which direction and ultimately estimate completely the wrong bias estimates. The preintegral measurement is again made in the presence of random IMU biases and the IMU is turned 90^o about its y-axis in the presence of gravity ${}^{w}\mathbf{g}_e = [0, 0, 1]^T$:

$$
\begin{aligned}
{}^{b_i}\Delta\tilde{\mathbf{x}}_{i\rightarrow j} \approx [0.00063, 0.0009, 0.00059, \mathbf{0}_3, 0.0007, -0.00027, & \quad (4.82)\\
1.00095, 0.00027, -8.0e-5, 0.50048, \mathbf{0}_3,]^T & \quad (4.83)
\end{aligned}
$$

As before, the compensation gradients vector ${\xi_{i\rightarrow j}}^T$, depicted in the figure below, is accumulated at the same time as the preintegral measurements and stored for later retroactive inference.



Inference with the measurement model and pose constraints, as used in previous examples, results in bias estimates as follows:

```
residual= 0.132 after 249 iterations (returned XTOL_REACHED)
Truth: [0.001,0.0001,0.0003,0.0003,0.0009,0.0007]
Estim: [0.00089,0.000443,-0.000297,5.9e-5,0.000895,7.2e-5]
error: [8.23,-408.57,203.01,79.19,0.21,87.82] %
```

The bias estimation error of $-408.57\ \%$ is immediately obvious, and a result of accumulation of integration errors during the preintegration process. This indicates one of the vital claims of this work, that improved integration of compensation gradients and preintegrals is paramount. If we repeat the above experiment with at a simulated sensor rate of $10000\ Hz$, as opposed to $1000\ Hz$ rate used above, we find greatly improved accuracy:

```
residual= 0.001 after 256 iterations (returned XTOL_REACHED)
Truth: [0.0015,0.0002,0.0,0.002,0.0011,0.001]
Estim: [0.00152,0.000235,3.7e-5,0.001977,0.001052,0.000995]
error: [0.18,-0.57,15.3,0.05,0.01,0.04] %
```

A marked improvement is seen, indicating the importance of sensor side precomputation of the preintegral, compensation gradient and noise covariance computations. The improvement due higher rate accumulation is summarized in the section hereafter. Again, the importance of continuous time preintegral models is not obvious, which this analysis contributes not found in other literature works.

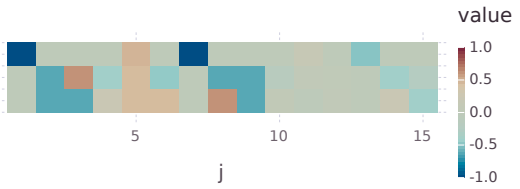
Bias estimates from two different pose locations

As a last basic example, we show how the compensation gradient terms operate when the IMU is moved between different pose locations during the integration interval. Consider an IMU, starting at rest, moved from the zero position in some world coordinates at pose ${}^{w}\mathbf{x}_i$, along some complicated and varying dynamic trajectory, to a pose ${}^{w}\mathbf{x}_j$ at position $[0.7, 0, 0]$ and tilted 90^o. The preintegral measurement vector is

$$\begin{aligned} {}^{b_i}\Delta\tilde{\mathbf{x}}_{i\rightarrow j} \approx [&1.5723, -0.0011, 0.001\mathbf{0}_3, 0.0013, \\ &-0.0005, 1.0004, 0.7008, 0.0001, 0.5006\mathbf{0}_3,]^T \end{aligned} \tag{4.84}$$

As before, the compensation gradients vector ${\xi_{i\rightarrow j}}^T$, which was accumulated at the same time as the preintegral measurements and depicted in the figure below, is stored for later retroactive inference. Notice how these values differ greatly

from the examples before. This implies the direction information throughout the interpose period is captured in this ${\xi_{i\to j}}^T$ vector.



To complete the example, inertial sensor

```
residual= 0.016 after 293 iterations (returned XTOL_REACHED)
Truth: [0.0015,-0.0011,0.001,0.002,0.0011,0.001]
Estim: [0.001608,-0.001575,9.3e-5,0.001988,0.000699,0.001115]
error: [-7.14,42.81,89.82,0.6,36.1,-11.4] %
```

These errors presented here represent the observability from a single trajectory segment, as well as a lower integration speed. As discussed before, multiple trajectory segments together have higher likelihood of full observability, coupled with the regularized bias (Gauss-Markov) assumption. Integration speeds will also be much higher, as discussed in the next section.

4.5 Update speed and feed-forward update

Fig. 4-4 illustrates the importance of high speed accumulation of preintegrals and compensation gradients. As the update frequency increases, the accuracy ofthe bias estimate over this single interpose constraint increases. Furthermore, if feed-forward compensation is applied, the Taylor expansion eq. (4.15) becomes better aligned and the accuracy can be further improved.

4.5.1 Real data Analysis

Experimental validation was done with a handheld IMU moved around in a VICON motion capture room. We moved a Microstrain GX3-25 IMU with VICON markers around for about one minute, in a trajectory snaking from one corner to the opposing corner of the room. Motion capture provides a reference measurement of the sensor package's position in some world frame. The purpose of these

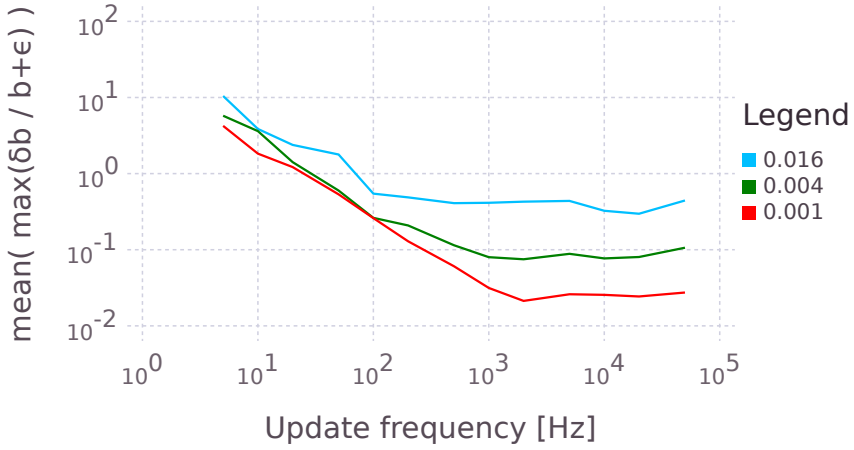


Figure 4-4: Accuracy of inferred inertial sensor biases using compensation gradients computed at higher and higher speeds. Three lines show 1σ level for randomly generated gyroscope, rad/s, and accelerometer, m/s^2, biases over a single interpose period. The y-axis shows the mean of the maximum among gyroscope and accelerometer biases divided by the actual bias, for 500 Monte Carlo runs. Bias error is denoted true sensor bias minus inferred bias, $\delta b = b - \hat{b}$. Note for this plot, a stabilizing term $\epsilon = 1e-5$ has been added to avoid division by zero.

validation tests are not to demonstrate an independent robot localization solution, but to investigate retroactive bias estimation.

Once confidence has been gained in our ability to estimate biases, we present localization and mapping solutions using inertial odometry in combination with aiding measurements such as a camera or kinematics. Results from a localization solution is presented in section 8.2.2.

For analysis, we use ground truth position, but not orientation, measurements (3 degrees of freedom) as priors to every 4 $seconds$ (fourth pose) in the trajectory, as shown in Fig. 4-5; this setup is similar to EKF style INS/GPS systems, and used here to validate and demonstrate the bias estimation characteristics of the proposed inertial odometry factors. Incremental solutions to the constructed SLAM problem were computed with a modified iSAM1.7 algorithm [116] using Powell's Dog Leg trust region method extension [197, 198].

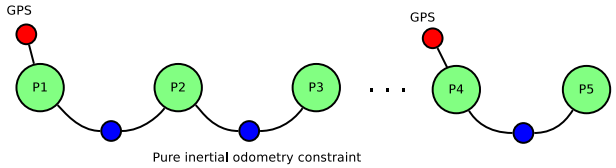


Figure 4-5: Sample factor graph used for validation of preintegral inertial sensor compensation model. Green poses are node points separated by pure inertial odometry constraints, red unary factors represent 3DOF position constraints from a GPS or Vicon position.

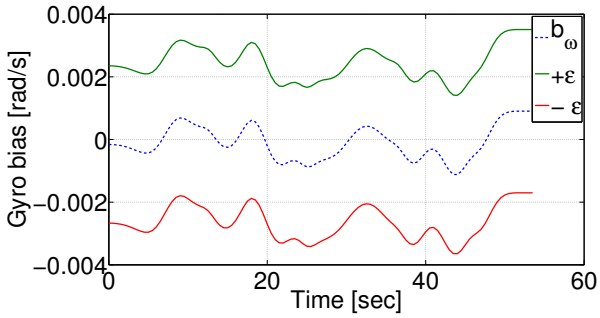


Figure 4-6: Gyro bias perturbation test to validate the Taylor expansion manifold assumption.

Retroactive bias estimation

The continuous time inertial odometry residual model relies on a local Taylor expansion, eq. (4.15). To experimentally explore that this assumption holds, we propose measuring a sufficient region of interest around the operating point of the expansion. We note the Taylor expansion represents an approximation of the manifold on which bias and gravity terms influence the inertial preintegral terms. Our region of interest is defined by the expected inertial sensor bias performance and availability of aiding information.

Using the Microstrain data in the VICON room data, as shown in Fig. 4-5, we quantify the repeatability of gyroscope bias estimation when it is perturbed by a known amount ϵ. The same SLAM solution is computed for three different cases while we artificially inject or remove sensor biases. The first is the nominal case, where all variables are estimated and bias estimates supposedly represent native errors in the system. We use this as a control experiment.

The experiment is then repeated twice, each with an artificial gyroscope bias

offset of $\epsilon = \pm 0.0025\ rad/s$ added to the $x - axis$ rate measurement. The three traces in Fig. 4-6 correspond to the resulting gyroscope bias estimates for each test. Visually we can see the distinct pattern repeated with the expected $\pm 0.0025\ rad/s$ offset. More quantitatively, if we look at the ratio of the remaining error:

$$std.\,dev.\left[\frac{\left(b_x - \hat{b}_x\right) - \epsilon}{b_x}\right] \approx 15\%. \tag{4.85}$$

The error associated with the 15% discrepancy is attributed to the stationary bias assumption; first order bias only compensation, eq. (4.15); ignoring of acceleration biases in eq. (4.51); and errors in the computation of matrix exponentials. Also note the 15% average includes possible division by zero, when the control bias estimate is zero. We therefore state the repeatability is likely much better than 85% for gyro terms, and note the gyro bias coupling is the most complicated.

Paying more attention to the sensor calibration process while estimating all sensor bias terms, we can gain insight into the overall operation of inertial odometry residual inference. Fig. 4-7 shows the accelerometer and gyroscope bias estimates for the same Microstrain IMU trajectory, as enforced by XYZ position measurements from a VICON system once every $4\ seconds$. We instantiate a new pose every $1\ second$. The x-axes show the pose number for three difference time instances in the IMU trajectory of around $1\ minute$, namely at $27\ s, 42\ s, 53\ s$. We specifically note the dashed and solid lines are sensor bias estimates for the same trajectory but at difference times. The curves show the bias estimates updating retroactively as observability changes. We also note, especially with gyro biases, that the earlier poses are better constrained with less variation as the trajectory evolves. The z gyro bias (horizontal) sees a significant update (change in observability) after $42.5\ s$, visible as the jump in the solid to dashed blue curve.

We emphasize that no VICON orientation measurements where used to estimate world frame rotations, yet we are able to recover a gyro bias estimates – which were baked into the preintegrals – with strong repeatability. Cross coupling of gravity into horizontal velocity states allow us to observe pitch and roll, while several separate reference position measurements constrain heading with respect to the world frame.

The retroactive bias estimates shown in Fig. 4-7 are interesting from an observability perspective. We note in Fig. 4-7 the estimates still vary between the $27\ s$ and $53.5\ s$ systems. In contrast to Fig. 4-7, where the x gyro bias estimates are pretty stable between the $42.5\ s$ and $53.5\ s$ systems. Note a larger change in the

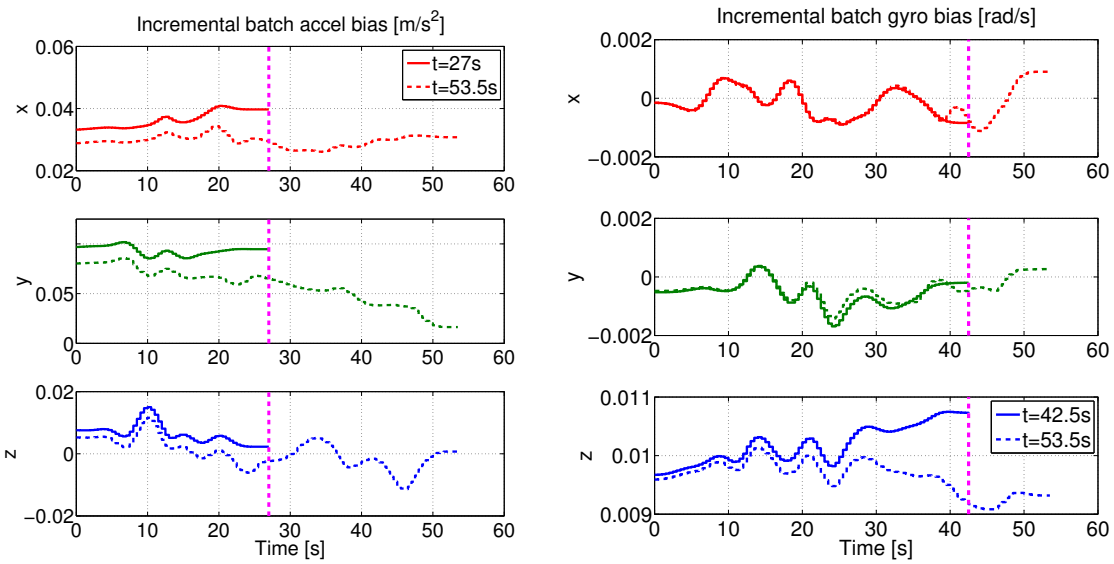


Figure 4-7: *Left*, inferred accelerometer bias estimates which are improved as more information is collected in the factor graph. The solid lines represent the incremental SLAM solution at the $27\ s$ point, the dashed line is the same trajectory but at $53.5\ s$ into the trajectory. Incremental solutions were computed with a modified version of iSAM1.7. *Right*, smoothed gyroscope bias estimates as more information is collected along with growth in the trajectory history.

z gyro bias. The stability of the bias estimates come down to observability of the different error characteristics. The x adn y gyro biases are well observed at $42.5\ s$. Improved bias observability, more sensor data or future loop closures all allow for better inertial bias estimation.

4.5.2 Gravity Estimate Validation

A last validation example involves the same Microstrain IMU trajectory discussed above, but processed to analyze the accuracy and importance of the gravity term ${}^{w}\mathrm{g}$. Using the residual eq. (4.21) together with the covariance eq. (4.67), we use a modified iSAM 1.7 implementation to perform non-linear least squares optimization. We also construct the factor graph description of the problem as given in Fig. 4-5, where inertial odometry likelihoods are regularly aided with the 3 degree of freedom (x, y, z) position priors.

We repeat the optimization multiple times, varying the gravity scale value a lit-

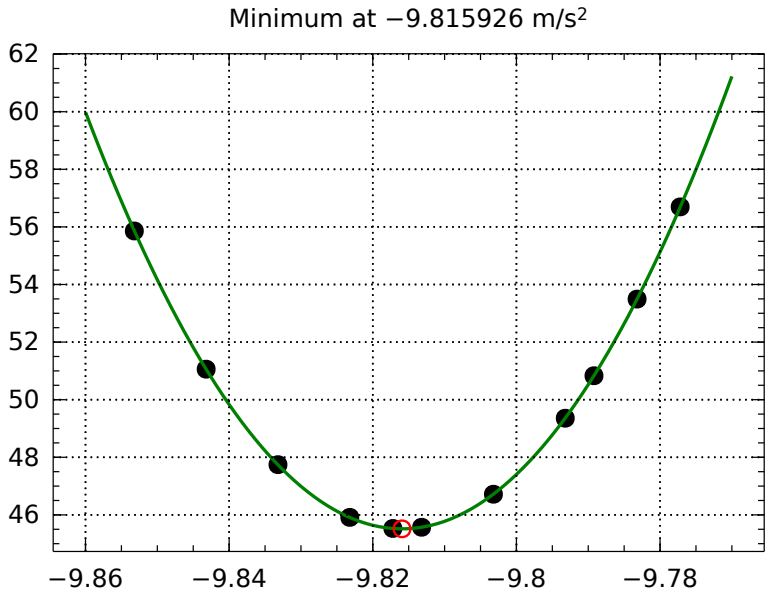


Figure 4-8: Gravity estimation with inertial odometry in a VICON room. The $x-axis$ shows the different gravity values and the $y-axis$ shows the least squares objective values (black dots) obtained from real data described in a factor graph for repeat solutions.

tle each time and recording the least squares objective value at the solution. Fig. 4-8 shows a quadratic cost bowl which is minimized at $-9.815926\ m/s^2$. After obtaining accurate local gravity information sheets, we find that our approach correctly estimates local gravity better than twice the best sensor bias stability performance.

Table 4.1: Gravity estimation accuracy using Microstrain GX-25 inertial odometry and VICON position measurements for a $\sim 1\ minute$ trajectory segment in a factor graph as shown in Fig. 4-5.

Description	Value $[m/s^2]$
Microstrain GX-25 1σ performance at $1g$	0.02500
Inertial Odometry with VICON position (3DOF)	9.81593
MIT, Cambridge reference (1970/77)	9.80399
WHOI, Woods Hole reference (1989)	9.80312
Error (doubled sensor performance)	~ 0.01200

4.6 Critical Assessment

Modeling of inertial sensor errors in pose graph form allows us to incorporate inertial measurements at a sufficiently low rate to be tractable, while capturing all motion dynamics. By analogy, a well calibrated inertial sensor solution is a very high quality motion model. Modeling of raw inertial sensor errors only allows for improved localization if sufficient aiding information is available to constrain the parameter estimates.

The value of the residual function at an optimal variable assignment should represent the accumulated measurement noise and approximation errors. Integrated gyro bias errors with gravity coupling are the dominant error source in inertial systems. Incorrect compensation of sensor errors ultimately results in a significant increase in the noise covariance $\mathbf{Q}$. Increasing process noise significantly increases the navigation system's dependence on other aiding information, thereby limiting many of the benefits an inertial navigation system can offer.

The error influences on the system vary from case to case, so it is important to keep in mind that these sensor bias estimates may very well contain more information than just sensor bias. For example, we did not yet model sensor errors such as scale factor errors and will show up in various other dimensions of the SLAM inference process.

Unlike filtering, factor graphs are more adept to asynchronous and opportunistic measurement events in the wild. The factor graph language allows us to readily add any aiding information that is available, alongside the inertial odometry constraints. The inertial solution observes gravity, producing a strong constraint on pitch and roll estimates in the SLAM solution.

We feel modeling of gyroscope and accelerometer bias terms with a succinct analytical derivation is well motivated. We argue such a derivation is not straight forward and the existence of such a sensor model will contribute to the community. Furthermore, we hope the explicit derivation will aid future investigations to derive yet more faithful models, including sensor scale factor error, temperature variation terms and more. We analytically derived continuous time, multidimensional, linear state system, outlined in eq. (4.30), which is more general than other proposed methods which have linearized and assumed zeroth order integration much earlier in the analytic development.

4.7 Conclusion

Our work promotes the use of pure, and dynamically calibrated, inertial measurements for odometry constraints in a factor graph (incremental smoothing) context. The inertial odometry residual function developed in this chapter can directly be used to construct a measurement likelihood model for least squares or nonparametric inference, which we discuss in Chapter 5.

While inertial sensors have been incorporated into smoothing style solutions, their error models have not yet fully exploited the smoothing computational structure. Most notably, gyroscope and accelerometer biases are generally not estimated in a fully-fluid manner. Fully-fluid implies the sensor bias error parameter estimates can be retroactively updated at any point in the trajectory history.

Incremental batch smoothing, along with the appropriate inertial sensor model, make retroactive sensor calibration possible and computationally tractable. The error model we present in this thesis is intended as a mechanism to naturally incorporate integrated raw inertial measurements in a factor graph formulation.

The ability to perform retroactive sensor calibration enables two desirable aspects. First, the ability to extract strong unimodal navigational aiding information from IMUs. Second, to integrate information from various sensors, as well as opportunistic *loop closures* containing powerful drift compensation information, in a succinct manner. We show that inertial odometry computations can fit well within the computational requirements for a real time system, with the caveat we only need to compute the incremental updates to a factor graph and these updates are reasonable in size.

Chapter 5

Multi-modal Inference

Chapters 3 and 4 expand our ability to model uncertainties captured in real-world measurements with new nonparametric and parametric likelihood functions for navigation-type factor graphs. This chapter avoids parametric optimization and establishes a more general non-parametric inference algorithm called *Mutli-modal iSAM* (incremental Smoothing and Mapping). Several canonical examples are presented in Chapter 6 and demonstrate that the approach is applicable beyond simultaneous localization and mapping problems.

5.1 Introduction

Many situations in robotic navigation rely on measurements which, realistically, require more powerful probabilistic representations than are currently offered by existing unimodal, parametric back-end solvers. Chapter 3 discussed how a robot front-end process assembled an arbitrary factor graph [129], as illustrated in Fig. 5-1.

This chapter develops an inference algorithm for non-Gaussian and multi-modal factor graphs, by inferring the marginals of all variables in the posterior joint probability function. The discussion follows in two parts, called the interclique and intraclique operations of the entire algorithm. Cliques emerge as groupings of variables and factors during symbolic refactoring of the factor graph into a Bayes tree [115].

Our algorithm, *Multi-modal iSAM*, exploits structure in the problem to convert a highly non-Gaussian, multi-Gaussian-hypothesis problem into a tractable and useful inference solution. Our discussion is focussed on understanding marginal density belief functions as a points in some feature space, and study how our Gibbs

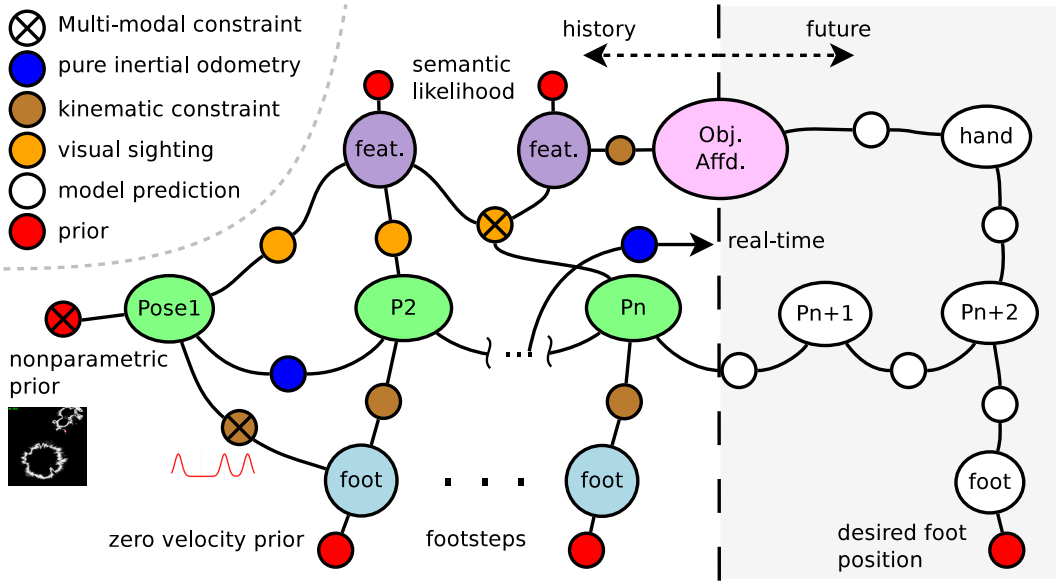


Figure 5-1: A nonparametric, multi-modal factor graph representing a joint probability density for the navigation of a humanoid robot (shown in Fig. 1-2) which is assembly by some front-end state estimation process, as discussed in detail in Chapter 3.

(Markov Chain Monte Carlo) method performs from the feature space perspective, and predict that mean map embeddings and contractive mapping proofs will play a vital role in development of non-Gaussian inference. Reversability of likelihood functions are required to establish detailed and global balance which ensures the inference algorithm will make progress. Canonical examples and real data tests are used in later chapters to empirically validate that the *sum-product* style inference algorithm performs as expected.

The algorithm is built around the idea of cliques, and discussed in two parts, namely interclique and intraclique operations. The first part of the chapter discusses interclique operations, also commonly referred to as belief propagation on the acyclic Bayes tree structure. The Bayes tree was originally developed as part of the parametric only iSAM2 algorithm [115]. Section 5.3.1 combines the *sum-product* inference methodology (using marginal densities) with the symbolic structure of the Bayes tree[1] to efficiently estimate the posterior beliefs of all variables.

[1]The Bayes tree is a derivative of the Bayes net similar Junction tree [37], but specialized with specialized controls to improve problem specific performance.

The latter half of the chapter discusses intraclique operations, which involves solving the Chapman-Kolmogorov integral equation. While the Bayes tree supplies an exact symbolic refactorization of variables and required passing structure for interclique operations, the intraclique operations are dominated by a general form expression known as the Chapman-Kolmogorov transit integral. A unique Chapman-Kolmogorov expression must be solved at each clique as part of the inference algorithm.

The intraclique operations decompose into continuous convolutions and products of infinite functions, which are discussed in Section 5.4. Our approach is inspired by the iterative sampling procedures of Tanner et al. [221] and Gelfand et al. [71], which retain asymptotic correctness of solution. Our approach makes minimal approximation error on dominant modes, but discards almost all low likelihood modes to reduce computational complexity. Convergence to a solution is based on reversability property of convolutions with conditionals and detailed balance of the nested-block-batch Gibbs sampling scheme.

The tractability of interclique and intraclique operations rests on multiple computational principles listed in Table 5.1. The combination of these principles circumvent or limit the impact of the possible exponential complexity (*curse-of-dimensionality*) associated with non-Gaussian posterior distributions. We will show that our approach is able to dynamically explore the entire *range space of possible multi-modal solutions* – defined in Section 5.2.2.

Table 5.1: Computational Principles

	Principle	*clique	Area	Accuracy	Sect.
1.	Symbolic Bayes tree [113]	inter	re-factoring	exact	5.3.1
2.	Recycling compute [115]	inter	re-factoring	exact	5.3.3
3.	Parallel compute	both	re-factoring	exact	5.3.4
4a.	Function approx. [208]	intra	prod.& conv.	asympt.	5.4.1
5.	Iterative approx. [221]	intra	prod. & conv.	asympt.	5.4.3
6.	Nested structure [125]	intra	prod.& conv.	exact	5.4.3
7.	Convolution approx.	intra	convolution	asympt.	5.4.4
8.	Modal consensus	intra	product	asympt.	5.4.6
9.	Product approx. [218]	intra	product	asympt.	5.5
4b.	Hybrid functions	intra	prod.& conv.	epsilon	5.5.5

The key insight among the nine computational principles is that one can save computation by tracking only the dominant modes in the marginal beliefs, while

only incurring only a (very minor) approximation loss. Modes are created by convolutions between likelihoods and culled by multiplications in a process of consensus. Note that multi-modal inference across all marginal beliefs is far more encompassing than a multiple hypothesis filtering approach where only a single time snapshot state estimate of plausible hypotheses is maintained.

To the best of our knowledge, this is the first larger scale *sum-product* style inference algorithm—allowing retrieval of marginal beliefs from the full joint posterior of any system variable—that can be applied to moderately large SLAM problems. Algorithm 1 presents a general overview of *Multi-modal iSAM*.

5.2 Joint Probability Distribution

Broadly speaking, inference algorithms typically fall into one (or a combination) of two main approaches: *sum-product* or *max-product*. To understand their relationship, we must start with the framework of a joint probability distribution.

From the Bayesian perspective, estimating any set of variables Θ involves trying to understand the joint probability distribution. We describe probability distributions through their densities as a function from space $\mathcal{P}$ over alphabet Ξ. Recall from Chapter 3, the joint belief $[\,\Theta \,|\, \mathbf{Z}\,] \in \mathcal{P}$ is modeled with a factor graph, as shown in Fig. 5-1. By Chain Rule, the product of independent measurements $\mathbf{Z}$, through likelihoods and variable prior potentials $[\,\mathbf{Z}_i \,|\, \Theta_i\,]$ and $[\,\Theta_j\,]$, represents the posterior joint probability density [129],

$$[\,\Theta \,|\, \mathbf{Z}\,] \propto \prod_i [\,\mathbf{Z}_i \,|\, \Theta_i\,] \prod_j [\,\Theta_j\,]\,. \tag{5.1}$$

Measurements Z_k are made from hidden states Θ_k according to an assumed measurement model. The inference task is to invert the system and estimate the belief over state variables Θ_i that likely produced the measurements seen.

To infer the belief of likely state over all variables, we focus on constructing the best possible joint probability density estimate. Joint distributions, however, are generally very complex and intricate – the details matter. While not true of every case, the computational complexity of a joint distribution can be exponential in the number of variables. That is, it is possible for each variable in the system to be affected by the state of any other variable in the system. Furthermore, these interactions are likely non-linear.

Assuming we can find the marginal posterlr beliefs for each variable, we can extract a point variable assignment according to some assumption, such as mean,

Algorithm 1: General overview for root clique marginal posterior belief—using *sum-product* style inference—over an entire nonparametric joint probability factor graph as part of upward message passing. Perform similar downward passing for recovering all marginals, and evaluate algorithm progress as discussed in Section 5.4.2.

Identify new portions of the joint probability factor graph from last iteration;
Extract the adjacency matrix of the modified portion of joint probability factor graph, and pass to CCOLAMD algorithm to get a good variable elimination order;
Starting from the root of the Bayes tree, unhook and store branches with unaffected symbolic structure by the changes to the factor graph;
Re-eliminate the modified portion of factor graph to cliques in a Bayes net (according to Algorithm 2), followed by reassembling the partial Bayes tree (according to Algorithm 3), and reattach stored branches at the correct location in the tree;
Note, belief of each variable is approximated by finite dimensional and smooth kernel density estimates, as discussed in Section 5.4.1;
foreach *(concurrently) clique j in a branch, starting from the leaves of the tree* **do**
    if *clique j is unmodified during Bayes tree update* **then**
        Recycle clique marginal beliefs $\hat{M}_{j|\mathbf{Y}}\left(\Theta_{C,j}\right)$
        Pass marginal belief over clique j separator variable to parent clique $\hat{m}_{j|\mathbf{Y}}\left(\Theta_{S,j}\right)$, as illustrated in Fig. 5-9;
    else
        Obtain the clique association matrix (CAM) for clique j, with an example shown in Table 5.2;
        Conduct Algorithm 4: Approximate clique marginal $\hat{M}_{j|\mathbf{Y}}\left(\Theta_{C,j}\right)$ (Section 5.4.3, as defined by the Chapman-Kolmogorov (CK) transit equation given by eq. (5.14)) – using nested structure seen in CAM and conceptually illustrated by Fig. 5-10. The steps are dominated by approximate convolutions (discussed in Section 5.4.4) and products of infinite belief functions (discussed in Section 5.5);
        Marginalize clique frontal variables according to CK eq. (5.15), $\hat{m}_{j|\mathbf{Y}}\left(\Theta_{S,j}\right) = \int_{\Xi} \hat{M}_{j|\mathbf{Y}}\left(\Theta_{C,j}\right) d\theta_{F,j}$;
        Pass marginal $\hat{m}_{j|\mathbf{Y}}\left(\Theta_{S,j}\right)$ upward (belief propagation) according to Fig. 5-7 and Fig. 5-6.

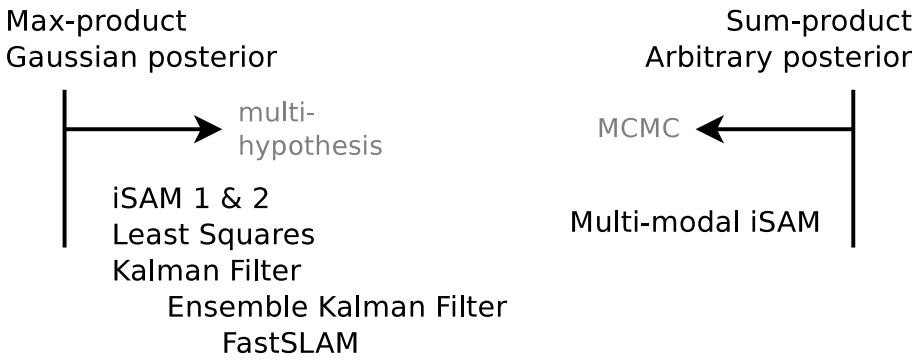


Figure 5-2: Conceptual illustration of the joint probability being estimated by *sum-product* versus *max-product* style robotic navigation inference solutions. We are promoting *Mutli-modal iSAM* as a fundamentally different algorithm that recovers a approximate function that estimates the marginal posterior distribution of a variable.

max, or fitting. The most probable point on the soution is assigned through maximization: $\theta_* = \text{argmax}_\theta \, p(\Theta|\mathbf{Z})$. For our purpose, we recover the full marginal beliefs rather than focusing on just the maximum-a-posterior estimate at this time.

The potential for exponential complexity will manifest at some point in any reliable inference procedure. One such manifestation is *the curse-of-dimensionality*, synonymous with computational complexity of multi-hypothesis type approaches. The potential for exponential complexity often leads to the use of assumptions or heuristics in any particular inference system. Common assumptions are to simplify or ignore interactions between variables, or reduce the resolution (granularity) of predicted uncertainty models of system variables.

We solidify the logic surrounding a factor graph model, shown in Fig. 5-1, as follows: Each time a platform is navigated, we are able to observe a single sample from the large joint density, which is approximated by the factor graph model. The joint probability is the distribution of countless repetitions of a similar trajectory under varying conditions. The joint distribution should summarize the uncertainty of infinitely many repeats of the same trajectory under all the remaining uncertainty from the measurement process.

In this thesis, we restrict our attention to sparse problems. By "sparse" we mean that the factor graph model is a sparse graph – this corresponds to the existence of a nontrivial factorization of the joint density which can be exploited for computation gains. Regardless of whether we use Gaussian or non-Gaussian measurement likelihoods, or factors, exploiting the sparse structure in the problem is critical for

efficient inference. Computing the full joint posterior over the entire factor graph, especially a non-Gaussian or multi-modal posterior, is extremely complex. Our only hope is to build the computations around structure within the problem.

We use the factor graph, as depicted in Fig. 5-1, to model structure in joint distribution, as shown in eq. (5.1), assuming all measurement likelihood functions have sufficient statistical independence [129]. In the next section, we adapt the existing Bayes (Junction) tree re-factorization to maximally exploit structure within the factor graph, and cast our approach as a *sum-product* type algorithm.

Before diving into the Bayes tree mechanics, we indicate the *max-product* type approach as a special case of *sum-product*, and show how the major computational simplification in *max-product* is a achieved through a significant loss of information. Thereafter, we discuss the Bayes tree as a general joint probability inference tool, and its relation to asymptotic information retention. The *sum-product* approach, discussed in the Section 5.3.1, makes the asymptotic information retention possible.

5.2.1 Current State-of-the-Art Assumptions

Mainstream solutions for high dimension joint probability problems follow from the *max-product* approach,

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \left[\Theta \,|\, \mathbf{Z} \right] = \underset{\theta}{\operatorname{argmax}} \prod_i \left[\mathbf{Z}_i \,|\, \Theta_i \right] \prod_j \left[\Theta_j \right]. \tag{5.2}$$

The nominal method, as developed by authors Legendre, Gauss [70] and Laplace from common averaging, is to assume quadratic measurement errors as a parametric Gaussian likelihood model [188]. The normally distributed likelihoods $\left[Z \,|\, \Theta_i \right] = \mathcal{N}\left(\mu = \delta\left(\theta_i \right), \Sigma \right)$ result in a common weighted least squares (maximum likelihood) optimization, Dellaert et al. [41]:

$$\begin{aligned} \theta^* &= \underset{\theta}{\operatorname{argmin}} -\log \left(\prod_i \exp\left(-\frac{1}{2} \| \, \delta\left(\theta_i, z_i \right) \|^2_\Sigma \right) \right) \\ &= \underset{\theta}{\operatorname{argmin}} \sum_i \| \ominus h\left(\theta_i \right) \oplus z_i \|^2_\Sigma. \end{aligned} \tag{5.3}$$

The optimization problems can be defined using on-manifold residual functions $\delta\left(\cdot \right)$, here depicted as $\delta = \ominus h\left(\cdot \right) \oplus z$. More generally designers utilize existing linear algebra mechanics through careful linearization, $\delta \approx \mathbf{z} - \bigtriangledown h\left(\cdot \right)$. Also note

that constant scaling factors are insignificant under the argmax_θ.

This family of approaches recovers the mean parameter, or first statistical moment, of the assumed multivariate Gaussian posterior. Marginal covariances (second moments) are computed separately. We note the posterior to be a multivariate normal distribution, as all terms are conjugate normal likelihoods and priors.

Only considering a time window of variables, in the least squares sense, is popularly known as sliding window smoothing. Information from much older measurements are marginalized together and influence the current state estimate through priors on the oldest variables in the sliding window.

The well known Kalman filtering expressions result from a further assumption: by reducing the sliding window length to a single snapshot of state, storing older information as a marginal Gaussian prior with state vector and covariance matrix as first and second statistical moments. To emphasize, parametric smoothing and filtering approaches mentioned here are special cases of the *max-product* approach, discussed above and in Section 2.4. The Kalman filter does do recursive Bayesian inference if all the Gaussian and uncorrelated assumptions are met.

5.2.2 Multi-modality: Displacing Assumptions

With modern computation, a normally distributed posterior may be overly restrictive for many real world events. We argue that SLAM type state estimation systems must contend with asynchronous and hard to predict events. The unpredictability of events makes it difficult to parse all sensor data perfectly, even with well-developed software procedures. As an extreme example, consider that even humans, albeit at much higher sophistication, are rarely able to correctly parse events when a magic trick is performed.

Given the unpredictability and potential obscurity of real world events, we should accept it as unlikely that a perfect front-end sensor processing system will always be available, regardless of the level of engineering effort spent. There will always be another corner case.

A perfect front-end would imply that perfect data association is achieved every time among vast amounts of sensory data. For example, a robot might think a camera is re-observing a previously detected 'Exit' sign, but in reality this could be either true or false. There is an inherent uncertainty associated with any such identification.

Much development effort has been spent on developing conservative front-end processes that only act when a high degree of certainty about true or false associations is available. Conservative front-end processes suffer both in loss of infor-

mation, such as significant loop closures, and lack transferability between applications. The really hard part comes when the uncertainty of multiple associations is roughly of the same scale and the front-end process is essentially 'splitting hairs' on whether or not to accept or reject potential associations.

Our discussion thus far has focused on discrete true or false decisions, a consequence of entrenched *max-product* style thinking. In the sections that follow, we will show how to efficiently deal with fully continuous uncertainty with our *Multi-modal iSAM* solution.

A multi-hypothesis approach is one avenue to mitigate the effect of binary data association uncertainties, and has been widely studied in target tracking applications, Reid, Fortman, Clark, et al. [35, 60, 195]. A multi-hypothesis system reduces data association uncertainties into decisions of yes or no, and at each fork in the road a separate *max-product* type inference procedure is started for the alternative decision. By tracking the separate likelihood uncertainty weights of each solution, such as Huang et al. [97], one could use heuristics to momentarily select a dominant solution among the exponentially many options.

We can algebraically demonstrate the exponential explosion in complexity associated with a full multi-hypothesis approach, by again looking at the joint probability distribution. Consider the product of likelihoods as shown in eq. (5.1) where two of the likelihood functions happen to be bi-modal, i.e. sum of (a, b) two equally likely normal distributions $\mathcal{N}(\mu, \Sigma)$,

$$\begin{aligned}
[\,\Theta\,|\,\mathbf{Z}\,] &\propto [\,\mathbf{Z}_1\,|\,\Theta_1\,]\,[\,\mathbf{Z}_2\,|\,\Theta_2\,]\prod_k [\,\mathbf{Z}_k\,|\,\Theta_k\,]\prod_{k'}[\,\Theta_{k'}\,] \\
&= (\mathcal{N}_{1,a} + \mathcal{N}_{1,b})\,(\mathcal{N}_{2,a} + \mathcal{N}_{2,b})\prod_k [\,\mathbf{Z}_k\,|\,\Theta_k\,]\prod_{k'}[\,\Theta_{k'}\,] \\
&= \left(\prod\cdots\right) + \left(\prod\cdots\right) + \left(\prod\cdots\right) + \left(\prod\cdots\right). \qquad (5.4)
\end{aligned}$$

These expressions show that the distributive law, which itself is an implicit convolution, produces four major terms in the joint probability density expression. Similarly, each additional multi-modal density results in an exponential increase in the number of summands in the joint probability expression.

We should immediately stress that many of the terms, while algebraically visible, have very low probability. Between all measurements, most modes collapse since the probability of the front-end actually observing those sequence of events is extremely unlikely.

Fig. 5-3 attempts to graphically illustrate the exponential complexity associated with a full multi-hypothesis approach as the upper curve. In contrast, the

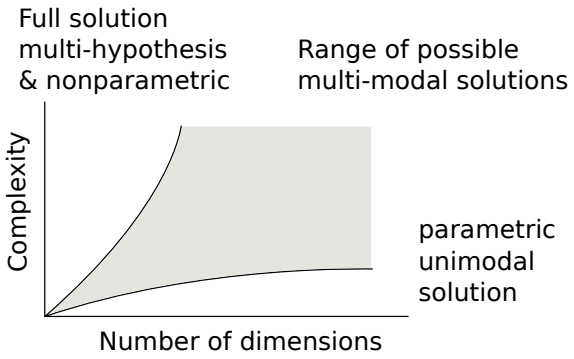


Figure 5-3: Illustration of unimodal vs. multi-hypothesis solution, with expanding space of possible multi-modal solutions.

unimodal parametric *max-product* solution (presented in Section 5.2.1) is a special case where each decision hypotheses is assumed correct and final, such that no alternatives are considered in parallel solutions. The lower curve illustrates the unimodal parametric case.

Fig. 5-3 also illustrates the enormous space between parametric unimodal and full blown multi-hypothesis solutions, which we will call the *range space of possible multi-modal solutions*. While the figure may initially seem trivial, the author believes there is a deeper observation; where we allow the front-end process to defer a reasonable amount of association uncertainties to the back-end inference procedure and let consensus happen there.

The question becomes, what do deferred association solutions look like? For example, consider a point on the lower right just above the unimodal parametric curve: (i) What is the complexity of tracking a limited number of hypotheses across the entire solution (FastSLAM [81]) versus tracking more modes in regions of the solution? (ii) Can we do better than assumed "outlier" approaches where measurements are de-weighted as null-hypotheses, such as *switch type variables* [76,206,219]? (iii) Can we select a piece of the problem where clear uncertainties exist and then treat just that portion in some special manner, leaving the rest as a conventional *max-product* parametric solution? (iv) Does a common algorithm exist where we can choose the granularity of the solution based on computational resources?

The questions boil down to a deeper question on the relation between approximated beliefs (nonparametrics) and multi-modality. We use structure within the problem to convert a highly complex multi-hypothesis, nonparametric system into

a more tractable multi-modal solution. The Bayes tree significantly reduces computation while still checking all uncertainty, as captured in the factor graph, and extracting consensus among all data.

The Bayes tree allows us to solve the complete problem while working at a per clique level and considering only local interactions, which is a direct consequence of the nontrivial factorization of the factor graph. Certain inference related operations are more efficient on the tree structure than a naive approach operating randomly over all variables in the factor graph. For example, updating all variables with a Hidden Markov Model type approach, such as a Kalman or particle filter, would terribly inefficient – as the square root smoothing and mapping algorithm [41, 42] showed over and above the inefficiencies of the earlier EKF-SLAM algorithm [11, 48].

We believe the correct avenue is to pursue an algorithm which can estimate multi-modal, non-Gaussian marginal posterior density functions, while maintaining asymptotic correctness properties. Granularity can then be varied based on available computational resources. The direct benefit of a varialbe granularity (resolution) algorithm is the ability to relax requirements on the front-end process. Given more powerful measurement models, as in Chapter 3, the user can be more optimistic when introducing measurement information. We stress that changing to non-Gaussian posteriors does not preclude use of the Bayes (Junction) tree [113], but instead actually casts the Bayes tree as a general framework for reducing computational complexity, regardless of the form and shape of likelihood functions used to assemble the factor graph.

Our purpose is therefore to reevaluate the set of operating assumptions from what has previously been used, such as in Section 5.2.1, to enable dedicated solutions that explore the *range space of possible multi-modal solutions*.

The next section describes the exact interclique operations on the Bayes tree, followed by a discussion on the asymptotically correct intraclique operations in Section 5.4.

5.3 Interclique: The Bayes Tree and *Sum-Product*

This section investigates the first of nine computation principles: Exploiting the symbolic structure in the joint probability density, which is critical for computational expedience. A general factor graph, as described by eq. (3.1), describes the inherent structure associated with the complicated joint probability.

Furthermore, we know the complexity in the factor graph is in part associated

with cycles in the graph [127]. The cycles create implicit algebraic loops which are hard to analytically compute. In the next section, we follow the Bayes tree algorithm [115] which successfully handles cycles in a near optimal manner.

5.3.1 The Bayes (Junction) Tree

We can convert a cyclic factor graph (through a variable elimination algorithm) into cliques in a Bayes network, and from there construct an acyclic tree representation which is algebraically equivalent to the original factor graph system.

Variable elimination games have been developed to re-factor a graphical model, like the factor graph, into an equivalent Bayes network, Pearl [187]. The variable elimination process can be viewed as an algebraic re-factoring of the same system. To illustrate, consider reorganizing factors in the joint probability into special groupings. We will refer to these groupings of likelihood factors as cliques.

An example joint probability that is randomly re-factored might look as follows:

$$[\,\Theta \,|\, \mathbf{Z}\,] \propto (\ldots)\,(\,[\,\mathbf{Z}_7 \,|\, \Theta_7, \Theta_2\,]\,[\,\mathbf{Z}_2 \,|\, \Theta_2\,]\,[\,\Theta_2\,]\,) \;\ldots\; (\ldots)\,. \tag{5.5}$$

Note, the number of possible factor grouping permutations is large. This space of permutations is related to the potential exponential complexity mentioned earlier in Section 5.2. A "bad" re-factoring will result in higher computational complexity.

By analogy, the original iSAM algorithm [116] connects poor re-factorings to a previously known problem, *fill-in* (in the case of Gaussian distributions). *Fill-in* is also a term familiar in the pivoting of rows and columns during of Cholesky factorization. The correspondence of refactoring permutations and *fill-in* is no coincidence, and formal methods (such as variable elimination games) have been developed to carefully find good re-factorizations, Koller et al. [127] and Kjearulff [125].

Algorithm 2: Extracting the Bayes network from factor graph

1 **foreach** $\theta_j \leftarrow$ *Variable Order* **do**
2 Remove from the factor graph all factors $[\,\mathbf{Z}_i \,|\, \Theta_i\,]$ that are adjacent to θ_j;
3 Form the local joint density $[\,\Theta_{F,j}, \Theta_{S,j} \,|\, \mathbf{Z}_j\,] \propto \prod_i [\,\mathbf{Z}_i \,|\, \Theta_i\,] \prod_{i'} [\,\Theta_{i'}\,]$;
4 Using the Chain Rule, separate the frontal and separator variables $[\,\Theta_{F,j}, \Theta_{S,j} \,|\, \mathbf{Z}_j\,] = [\,\Theta_{F,j} \,|\, \Theta_{S,j}, \mathbf{Z}_j\,]\,[\,\Theta_{S,j} \,|\, \mathbf{Z}_{j'}\,]$.

The elimination process algorithm, which produces the special groupings of

likelihood factors, requires a variable ordering as input. A good variable elimination order is one that maximizes the number of minimum dimension cliques, and is analogous in the parametric case to minimizing *fill-in*. The algorithm, modified from [115], is presented in Algorithm 2. The resulting Bayes network is chordal, meaning there are no large cycles in the re-factored model.

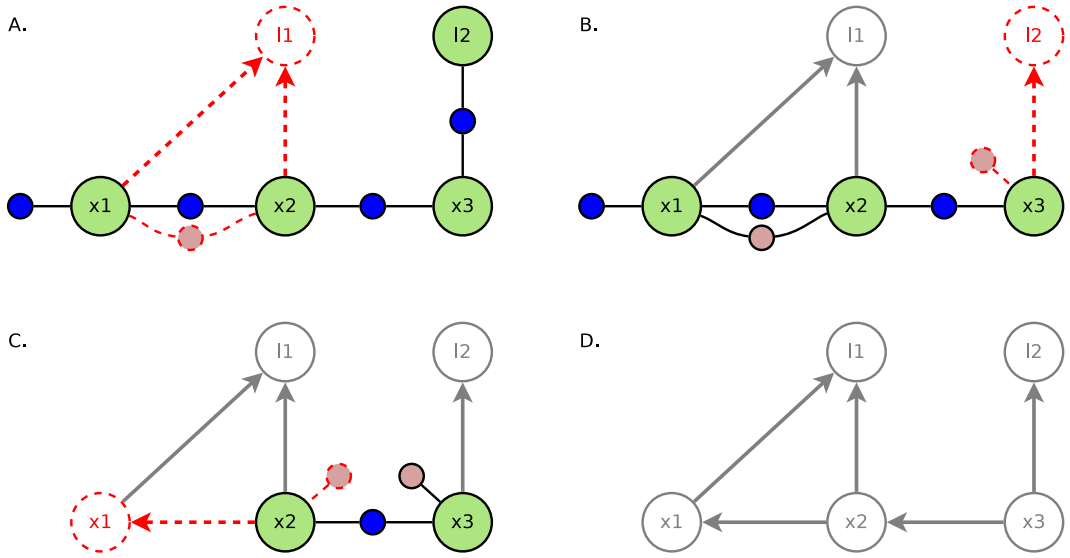


Figure 5-4: Elimination procedure to convert a factor graph into an equivalent Bayes network re-factoring of the underlying joint probability distribution. Green and blue circles show variables and factors, respectively. Condensed example from [115], using variable ordering l_1, l_2, x_1, x_2, x_3, and working along the rows from the factor graph top left to the equivalent Bayes network bottom right. The elimination process is illustrated with red dashed lines.

Following from iSAM1 and iSAM2 [115, 116], we use the Constrained-CHOLMOD heuristics [39] to obtain good variable orderings. Fig. 5-4 illustrates how the elimination process converts a factor graph segment into an equivalent Bayes network. The factor graph represents the product of six likelihood factors:

$$[\,\Theta \,|\, \mathbf{Z}\,] \propto [\,X_1\,]\,[\,Z_1 \,|\, X_1, X_2\,]\,[\,Z_2 \,|\, X_1, L_1\,] \times \\ [\,Z_3 \,|\, X_2, X_3\,]\,[\,Z_4 \,|\, X_2, L_1\,]\,[\,Z_5 \,|\, X_3, L_2\,]\,. \tag{5.6}$$

We illustrate the elimination procedure, as shown in Fig. 5-4. The variable L_1 is eliminated first, which involve connected, and thus far unclaimed, factors $[\,Z_2 \,|\, X_1, L_1\,], [\,Z_4 \,|\, X_2, L_1\,]$. We combine the factors in a local product,

Algorithm 3: Extracting Bayes tree from Bayes network, Kaess et al. [115].

```
foreach θ_j ← Reverse Variable Order do
    if no parent (Θ_{S,j} = {}) then
        start a new root clique Θ_{FR} with Θ_j;
    else
        identify parent clique C_p so that it contains the first eliminated
        variable of Θ_{S,j} as a frontal variable;
        if Θ_{F,p} ∪ Θ_{S,p} of parent C_p is equal to separator Θ_{S,j} conditional then
            insert conditional into clique C_p;
        else
            start new clique C' as child of C_p containing θ_j;
```

$[\, X_1, X_2, L_1 \,|\, Z_2, Z_4 \,] \propto [\, Z_2 \,|\, X_1, L_1 \,] [\, Z_4 \,|\, X_2, L_1 \,]$, and then use the Chain Rule to obtain a conditional and marginal, $[\, L_1 \,|\, X_1, X_2, Z_2, Z_4 \,]$ and $[\, X_1, X_2 \,]$ respectively.

Variables $\{X_1, X_2\}$ are the only ones connecting L_1 to the rest of the factor graph, and is called the separator set for L_1. We denote the separator as Θ_{S,L_1}. We note that the new marginal $[\, X_1, X_2 \,]$ is added to the factor graph, as indicated in Fig. 5-4.

The illustration does not directly show how all cycles in the graph have now been concentrated into new groupings of likelihood functions, which are called cliques. More than that, cliques are a maximally-connected proper subset [222].

We extract the cliques from the Bayes network by reading off the Bayes tree, with the maximum cardinality search algorithm [222] and the reversed variable ordering from before. The process of extracting cliques is summarized in Algorithm 3, and the resulting tree from our example is show in Fig. 5-5. For example the left leaf clique j contains $\Theta_j = \{X_1, L_1, X_2\}$.

Each clique j represents a conditional belief over a subset of variables, $[\, \mathbf{Z}_j \,|\, \Theta_{F,j}, \Theta_{S,j} \,]$. Note, the associated measurement likelihoods are carried with the respective cliques as $\mathbf{Z}_j$. Each clique is associated with a non-overlapping set of likelihood factors. We can now use all cliques to obtain the original joint probability density, which is the product of these r cliques:

$$[\Theta|\mathbf{Z}] \propto \prod_{j=1}^{r} [\, \mathbf{Z}_j \,|\, \Theta_{C,j} \,]\,. \tag{5.7}$$

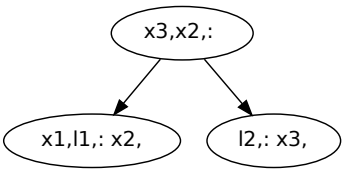


Figure 5-5: Bayes tree refactoring of example factor graph shown in Fig. 5-4.

For convenience, we take all variables of clique j as the union of frontal F and separator S variables, $\Theta_{C,j} = \Theta_{F,j} \cup \Theta_{S,j}$. In eq. (5.12) we will swap the clique measurement variables $\mathbf{Z}_{-}$ with a more meaningful definition of $\mathbf{Y}_{\cdot}$.

The Bayes tree re-factoring has a number of special properties [114]. Clique variables are a strict subset of all variables $\Theta_{C,j} \subseteq \Theta$. Frontal and separator variables in each clique are always disjoint, $\Theta_{F,j} \cap \Theta_{S,j} = \emptyset$, and frontal variables between any cliques are also disjoint, $\Theta_{F,j} \cap \Theta_{F,i} = \emptyset,\ i \neq j$. Along any branch with the root having no parents, cliques do not contain variables already in frontal variables of their children, $\Theta_{C,j} \cap \Theta_{F,child} = \emptyset$. Cliques are only related through their separator variables $\Theta_{S,child} \subseteq \Theta_{C,j}$.

The conditional independence represented by the Bayes tree factorization, following from the Bayes network chordal property, allows us to consider only local interactions at each clique. Since each clique in the tree represents a partial posterior over the local frontal and separator variables, we can exploit the local structure in a *sum-product* methodology, similar to [37]. This local encoding has much lower dimension than the full posterior distribution.

5.3.2 *Sum-Product* Perspective (Belief Propagation)

We now proceed to develop belief propagation along the symbolic structure of the Bayes tree rather than the original factor graph. By using the conditional independence structure of the acyclic Bayes tree, i.e. its chordal property, we only need use local interactions at each clique to focus computational effort and find the system solution.

The original iSAM2 algorithm assumes the *max-product* approach along with Gaussian models, as outlined in Section 5.2.1. We will instead follow the *sum-product* approach and simultaneously allow non-Gaussian likelihoods.

We focus computation, first to clique level, by marginalizing out all variables

but those in $\{i\}$ from the joint probability distribution,

$$\int_{\Xi} [\,\Theta \,|\, \mathbf{Z}\,]\; d\Theta_{\setminus\{i\}} = [\,\Theta_i \,|\, \mathbf{Z}\,], \tag{5.8}$$

to produce the full marginal posterior distribution of Θ_i given all measurements in the factor graph $\mathbf{Z}$. By using the Bayes tree re-factored joint probability, we can marginalize to any of the cliques

$$\begin{aligned}[\,\Theta_{F,j} \,|\, \mathbf{Z}\,] &= \int_{\Xi} [\,\Theta \,|\, \mathbf{Z}\,]\; d\Theta_{\setminus\{F,j\}} \\ &\propto \int_{\Xi} \prod_{k=1}^{r} [\,\mathbf{Z}_k \,|\, \Theta_{C,k}\,]\; d\Theta_{\setminus\{F,j\}}\end{aligned} \tag{5.9}$$

By considering the root clique, $j = 1$, as somewhat special (no separator variables), we realize that marginalizing all but the root frontal variables $\Theta_{F,1}$ produces a special cascading result:

$$\begin{aligned}[\,\Theta_{F,1} \,|\, \mathbf{Z}\,] \propto \int_{\Xi_2} [\,\mathbf{Z}_2 \,|\, \Theta_{C,2}\,] \int_{\Xi_3} [\,\mathbf{Z}_3 \,|\, \Theta_{C,3}\,] \,\ldots\, d\Theta_{F,3} \times \\ \int_{\Xi_j} [\,\mathbf{Z}_j \,|\, \Theta_{C,j}\,] \,\ldots\, d\Theta_{F,j} d\Theta_{F,2}.\end{aligned} \tag{5.10}$$

The integral operations of variables are over their respective complete alphabets $\theta_{F,j} \in \Xi_j$, and can be separated based on the structure of frontal and separate variables of cliques lower down in the tree. In our example $\Theta_{F,2}$ must be marginalized last but other variables, such as $\Theta_{F,3}$, can be marginalized sooner.

We recognize that the leaves of the tree have no children and we can readily marginalize out their frontal variables first,

$$\int_{\Xi_j} [\,\mathbf{Z}_j \,|\, \Theta_{C,j}\,]\; d\Theta_{F,j}, \tag{5.11}$$

and calculate the expression by using the likelihood potentials and priors associ-

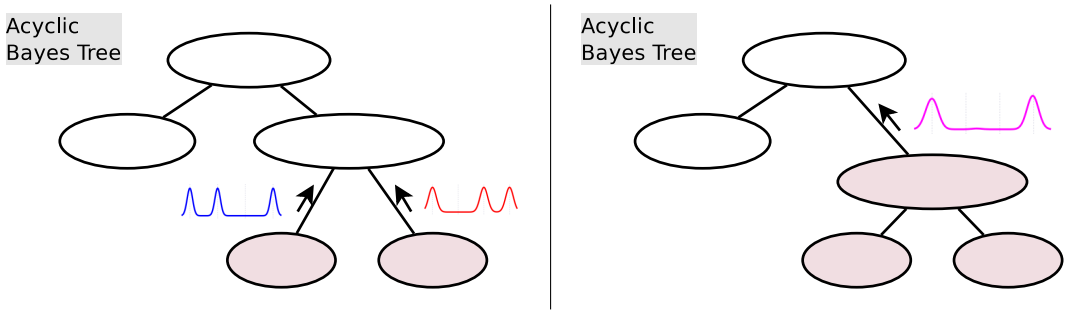


Figure 5-6: Conceptual Bayes tree, showing multi-modal beliefs being passed up from leave cliques towards the root clique. The shaded leaf cliques illustrate the Chapman-Kolmogorov transit integral operation has been completed, after which up-message passing occurs and the process repeats in the parent clique. Fig. 5-9 continues the discussion on intraclique operations.

ated with each clique j,

$$\begin{aligned}
[\,\Theta_{S,j}\,|\,\mathbf{Y}_j\,]\,|_{j=L} &= \int_{\Xi_j} [\,\Theta_{C,j}\,|\,\mathbf{Y}_j\,]\; d\Theta_{F,j} \\
&\propto \int_{\Xi_j} [\,\mathbf{Y}_j\,|\,\Theta_{C,j}\,]\; d\Theta_{F,j} \\
&= \int_{\Xi_j} \prod_{k}^{C_j} [\,\mathbf{Z}_k\,|\,\Theta_k\,] \prod_{k'}^{C_j} [\,\Theta_{k'}\,] \times 1\, d\Theta_{F,j}. \qquad (5.12)
\end{aligned}$$

We collect measurement variables $\mathbf{Z}_j$, starting from the leaves L towards the root, in a new variable $\mathbf{Y}_j = \{\bigcup_k^{L\to j} \mathbf{Z}_k\}$. In the initial case of the leaf clique itself, we have $\mathbf{Y}_L = \mathbf{Z}_L$, and at the root we have all measurements $\mathbf{Y}_1 = \mathbf{Z}$. Section 5.4.4 discusses the mechanics of working with the likelihood potential functions.

We view each clique as a conditional density, or a partial posterior, over the separator variables. The parent clique relies on the separator variables $\Theta_{S,j}$ from the child cliques. For convenience, the marginal belief $[\,\Theta_{S,j}\,|\,\mathbf{Y}_j\,]$ of each clique is dubbed a *belief message*. The beliefs contained in the outgoing message from the clique is written as

$$m_{j|\mathbf{Y}}\,(\Theta_{S,j}) = [\,\Theta_{S,j}\,|\,\mathbf{Y}_j\,]\,. \qquad (5.13)$$

Fig. 5-6 illustrates multi-modal belief messages being propagated up from the leaves towards the root of the Bayes tree.

Again, we compute the partial posterior over the clique variables using the local likelihoods, as well as the incoming belief messages from the child cliques, and introduce a message-marginal shorthand $M_{j|\mathbf{Y}} \in \mathcal{P}$:

$$M_{j|\mathbf{Y}}\left(\Theta_{C,j}\right) \doteq \left[\, \Theta_{C,j} \,|\, \mathbf{Y}_j \,\right] \propto \prod_{k}^{C_j} \left[\, \mathbf{Z}_k \,|\, \Theta_k \,\right] \prod_{k'}^{C_j} \left[\, \Theta_{k'} \,\right] \prod_{u} m_{u|\mathbf{Y}}\left(\Theta_{S,u}\right). \tag{5.14}$$

As before, we marginalize out the frontal variables which are not present in the parent clique, leaving a belief function over clique separator variables $\Theta_{S,j}$

$$m_{j|\mathbf{Y}}\left(\Theta_{S,j}\right) \doteq \left[\, \Theta_{S,j} \,|\, \mathbf{Y}_j \,\right] = \int_{\Xi_j} M_{j|\mathbf{Y}}\left(\Theta_{C,j}\right)\, d\Theta_{F,j}. \tag{5.15}$$

The density obtained from this marginalized product is passed up the tree as another belief message to the next parent clique. The parent clique then repeats the process, multiplying all incoming densities with local potentials and marginalizing out all frontal variables, as illustrated in Fig. 5-7.

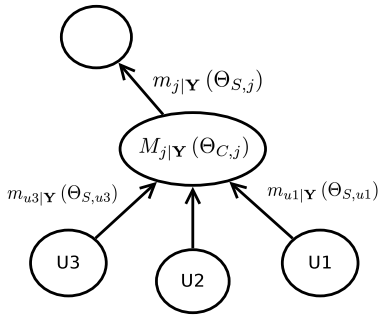


Figure 5-7: Illustration of belief propagation for clique j with marginal partial posterior $M_{j|\mathbf{Y}}$, during the upward pass with multiple children $u_k,\ k = 1, 2, 3$ and a single parent clique. Belief propagation messages $m_{k|\mathbf{Y}}\left(\Theta_{S,k}\right),\ k = j, u_1, u_2, u_3$ are illustrated.

The product and marginalization of potentials continue up the tree until the root node is reached. Since the root has no parent clique, a marginal of the full system posterior density has been obtained. These steps compute all the integral and product steps listed in the marginal joint probability density shown in eq. (5.10).

We summarize the process described by eqs. (5.14), (5.15) in Fig. 5-7. This process is equivalent to the Chapman-Kolmogorov transit integral operation. Practical computation of eq. (5.15) requires convolutions and products of infinite functions

between functions, as shown by eq. (5.14). We discuss these processes separately in Section 5.4.

Once all belief messages up the tree and marginalization has completed, we recover the first complete marginal beliefs of the frontal variables in the root, $[\,\Theta_{F,1}\,|\,\mathbf{Z}\,]$. If we require all variable marginals, we can pass similar belief messages back down the tree to all the child cliques and compute the complete marginal posteriors for each variable.

We stress eqs. (5.14) and (5.15) represent a *sum-product* inference computation. The messages traveling on the tree encapsulate multi-modal and non-Gaussian beliefs for each of the variables available at that stage in the computation. We also stress that the interclique messages are marginals and not inter-variable conditionals from leaf L to j, $[\,\Theta_j\,|\,\mathbf{Y}_j\,] \neq [\,\Theta_j\,|\,\mathbf{Y}_j, \Theta_{L\to j}\,]$.

To illustrate the benefit of the Bayes tree, we contrast belief propagation on the tree against loopy belief propagation directly on the factor graph [194], which must repeatedly iterate belief propagation in a seemingly random order across the entire network until a stationary solution is found. Loopy belief propagation may not converge to a stationary solution since the order of variable propagations can change, and as a result there is a lack of guarantee for overall algorithm performance.

5.3.3 Incremental Belief Propagation

This section completes our discussion of exact interclique operations with the second of nine computational principles, namely incremental recycling of computations. Asymptotic solution to the required Chapman-Kolmogorov transit integral operation is discussed next, in Section 5.4.

Modification of the factor graph is common in state estimation and simultaneous localization and mapping problems. As more information is accrued by the front-end process, major computational savings are available through basic symbolic operations on the tree.

Consider a modification to our example factor graph shown in Fig. 5-8 where the number of system variables is increased by $\Theta_{\Delta k}$, as well as more measurement information $\mathbf{Z}_{\Delta t}$. The updated Bayes tree with one recycled clique is also shown in Fig. 5-8.

$$[\,\Theta_t\,|\,\mathbf{Z}_t\,] \to [\,\Theta_t, \Theta_{\Delta t}\,|\,\mathbf{Z}_t, \mathbf{Z}_{\Delta t}\,]\,. \tag{5.16}$$

We emphasize the set of continually changing variables with the time subscript t.

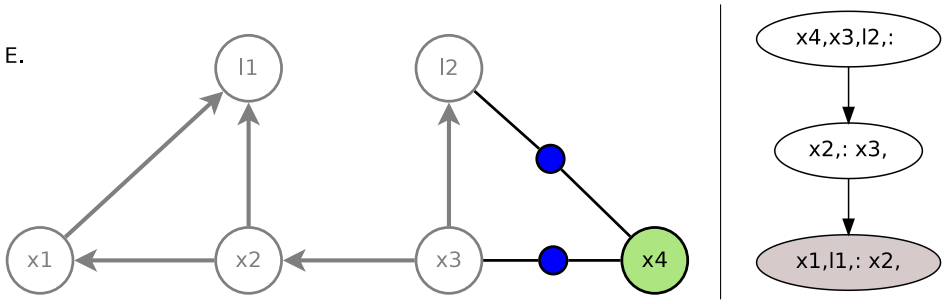


Figure 5-8: *Left*, modified factor graph (E.), and Bayes network from Fig. 5-4, showing new variable X_4 with two new factors to X_3 and L_2. *Right*, updated Bayes tree previously shown in Fig. 5-5 – notice how the shaded clique can be recycled using new variable ordering $l_1, l_2, x_1, x_2, x_3, x_4$. Note, this illustration is not an optimal elimination order.

Accordingly, the joint probability density is changed, but this results in limited changes to the tree. If we are careful about how the variable ordering is updated we need not recompute the entire tree. By paying close attention to the message passing operations on the trees, we see that full branches below an unchanged clique can be recycled.

The variable-ordering heuristics used to construct the Bayes tree can be modified to specifically minimize both clique dimension as well as modifications to the tree when the underlying factor graph is modified. By controling the variable elimination order such that much of the previous variable ordering remains unchanged, the required changes to the tree are reduced. The regions for symbolic modification are either at the root or leaves of the tree, where the upward messages flow from leaves to root along a particular branch.

Variable Ordering: Optimizations for Exploration

By keeping modifications localized near the root of the Bayes tree, we can *unhook* large branches, re-factor the modified factor graph section, and reattach previous branches onto the new tree's near root segment. Reattaching previous branches at the correct location in the tree keeps intact all their internal computations. This allows us to keep track of large amounts of sensor measurements without sacrificing inference performance or repeating computations, as illustrated in Fig. 5-9.

Using CCOLAMD, the most likely changes are forced to the top of the tree by placing the most recent variables at the end of the variable ordering. Recall the tree is constructed with the reverse variable ordering. Maximizing the number of

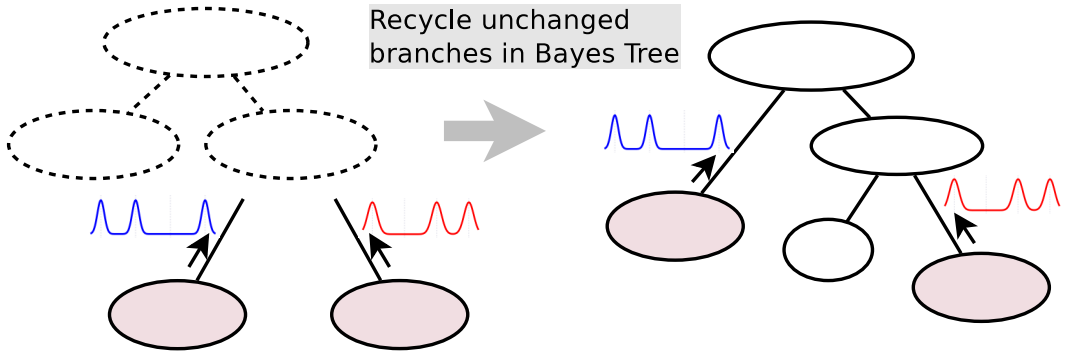


Figure 5-9: Illustration of incremental belief propagation by recycling computations associated with older and stable branches in the Bayes tree. The two shaded cliques and associated upward belief messages are recycled given limited changes to the factor graph.

unchanged variables earlier in the ordering results in larger branches lower down on the tree which results in the best turn-on-turn recycling of computations.

The down side of constantly updating only the back of the variable ordering is that the global variable ordering quality decreases. We control this parasitic effect by balancing the update size and global variable ordering quality. Very large graph applications might benefit from periodic global variable reordering, thereby updating the entire tree and requiring fresh intraclique computations.

Changes to the near root region of the tree minimize computations for the initial upward message passing operations, but new information influences all variable beliefs during the downward message passing phase. Computations can be limited by stopping downward propagation when changes to beliefs become insignificant. Kaess et al. [115] refer to this procedure as the *wildfire* algorithm.

Variable Ordering: Fixed-lead/lag Operation (Marginalization)

Fine grain control of variable ordering can also be used to semi-permanently marginalize older branches into a fixed upward belief message at some new defined leaf boundary clique. This is equivalent to marginalization in a filtering context where previous states are marginalized into the previous posterior model estimate.

By forcing older factor graph information downwards in the tree, a boundary can be found where no changes happen below that point in the branch. By limiting the upward propagation of information from that boundary, all information

lower down is effectively marginalized to a fixed state. Sliding window operation is achieved by considering only upward belief messages from the boundary cliques as the limits of computation. Practically this involves the user marking variables for marginalization and then forcing those variables to the back of the elimination order. Stale portions of the tree below the boundary cliques can be deleted or archived.

5.3.4 Parallel Computations

The third computational principle is parallel computation, and is a benefit from the statistical independence structure of the Bayes tree. As with eq. (5.10), we identified that multiple branches hang from the root clique of the tree. Each of these branches are independent and can be computed separately.

Consider the upward message passing phase, as shown in Fig. 5-6. We start by computing each of the leaf intraclique operations in parallel, as given by eq. (5.12), producing belief messages to be passed up to each of the parents. The process repeat then repeats in the parent clique as messages propagate up the tree.

There are fewer parent cliques towards the root, and finally a sigle intraclique operation at the root where the first complete posterior beliefs are computed. The downward pass increases the number of messages and child cliques allowinf for greater parallel computation. There is further structure and room for parallel computation within a clique operation, but the discussion is deferred to upcoming sections.

5.4 Intraclique: Chapman-Kolmogorov Integral

In the previous sections, we discussed the how information flows on the Bayes tree during interclique operations, as illustrated by Fig. 5-6. To complete the *sum-product* methodology, in this section, we discuss the intraclique operations, also shown by Fig. 5-10. A convenient algorithm summary is presented in Section 5.6.

To complete the interclique message passing process, we require the marginal joint density for each clique, as defined by eq. (5.14). Fig. 5-7 with eq. (5.14) illustrate how incoming messages, $m_{j|\mathbf{Y}} \in \mathcal{P}$, and local potentials are multiplied together. Fig. 5-10 previews the product between incoming messages, which contains three modes each but only two significant modes in the product result. Note, the discussion on products between approximated functions is discussed in detail in Section 5.5 hereafter, while this section discusses where and how the product of infinite functions is used in the inference procedure.

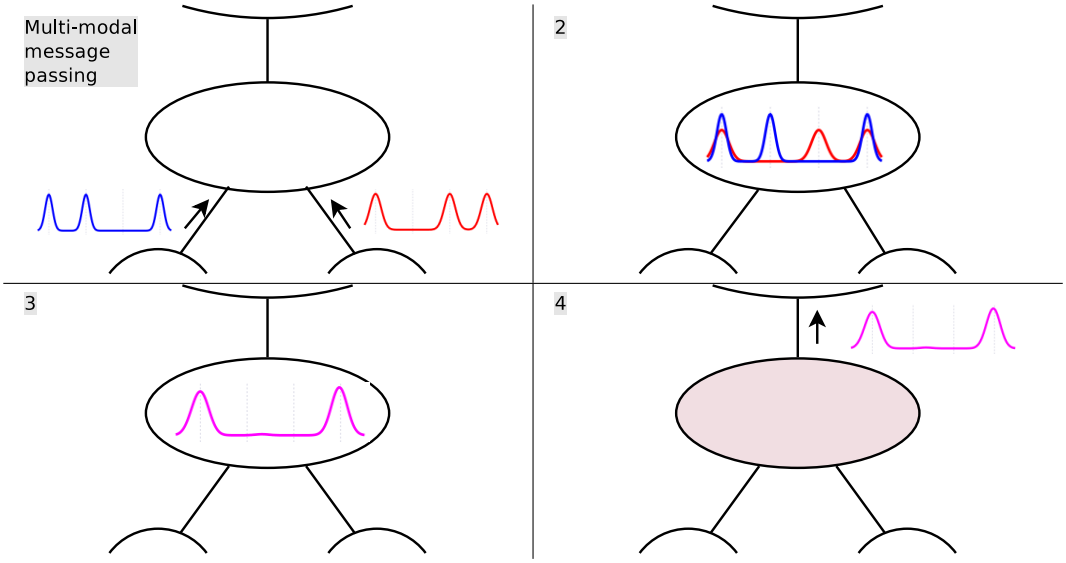


Figure 5-10: Illustration of the Chapman-Kolmogorov transit integral operation on a clique, shown during the upward pass on the Bayes Tree.

We further discuss the computational principles associated with intraclique operations (listed in Table 5.1). All the computational principles (approximations) associated with the intraclique operations maintain the exactness property, maintaining focus on fewer dominant modes as computational resources are restricted. Computational load is scheduled by computing the dominant (most likely) modes and stochastically pruning lower likelihood modes.

5.4.1 Functional Interpretation

From eq. (5.1), we seek a solution to the joint probability density on a function space $\mathcal{P}$ over the domain Ξ_j of each clique

$$[\,\Theta_{C,j}\,|\,\mathbf{Y}_j\,] \in \mathcal{P}, \qquad\qquad \Xi_j \;:\; \mathbb{R}^n \times \mathcal{X}^d \tag{5.17}$$

with variables over some combined continuous and discrete space $\Theta_{C,j} \in \Xi_j$. Continuous and discrete variable dimensions are n and d respectively. A probability density function is taken as:

$$p : \Xi \to \mathbb{R}, \;\; \text{and} \;\; \int_{\Xi} p\,(\theta)\; d\theta = 1, \qquad \forall p \in \mathcal{P} \tag{5.18}$$

where true beliefs p are assumed to already exist in a bounded Banach space, and may have infinite degrees of freedom.

A measure of similarity $d(p_1, p_2)$ between belief functions $p_1, p_2 \in \mathcal{P}$ is useful for analysis,

$$d : \mathcal{P} \times \mathcal{P} \to \mathbb{R}. \tag{5.19}$$

A common measure of dissimilarity, $d(\cdot, \cdot)$, is the KL-divergence $D_{KL}(\cdot \| \cdot)$

$$D_{KL}(p_1(\Theta) \| p_2(\Theta)) = \int_{\Xi} p(s) \log \frac{p(s)}{q(s)} \, ds, \tag{5.20}$$

which is not symmetric and does not obey the triangle inequality, but does satisfy the Pythagoras theorem discussed later at eq. (5.27). Ideally we would like to work with a true metric distance which does satisfy the identity element, symmetry and triangle inequality properties. We can also consider the L_2-metric:

$$d(p_1, p_2) = \int_{\Xi} (p_1(\theta) - p_2(\theta))^2 \, d\theta. \tag{5.21}$$

The Wasserstein probabilistic metric [75] is considered a robust option should the analysis require. We may also consider the distances between mean embedded features spaces,

$$d(p, q) = \| \mathbb{E}_p[\phi(\Theta)] - \mathbb{E}_q[\phi(\Theta)] \|, \tag{5.22}$$

which is discussed more in Section 5.4.1.

The exact clique marginal belief $[\Theta_{C,j} \mid \mathbf{Y}_j] \in \mathcal{P}$ has potential exponential complexity in the dimension of the variables directly influenced by $\mathbf{Y}_j \subseteq \mathbf{Z}$. A practical algorithm to estimate this belief must therefore approximate the belief

$$[\Theta_{C,j} \mid \mathbf{Y}_j] = \mathbf{M}_{j|\mathbf{Y}} \approx \left[\hat{\Theta}_{C,j} \mid \mathbf{Y}_j\right]^{(i)} = \hat{\mathbf{M}}_{j|\mathbf{Y}}{}^{(i)} \tag{5.23}$$

at the i^{th} iteration of some inference process. Note the shorthand for clique marginal $\mathbf{M}_{j|\mathbf{Y}}$ from eq. (5.14). We use the hat in $\hat{M} \in P$ to indicate that an approximation of the belief function is made. We discuss the function approximation space $P \in \mathcal{P}$ in the next section.

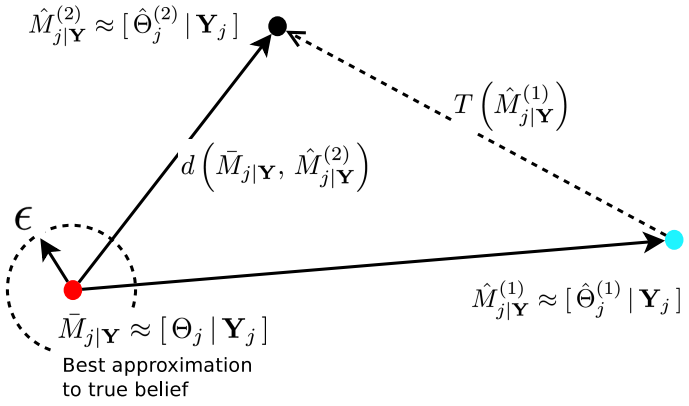


Figure 5-11: Conceptual description of manipulating an approximated belief function $\left[\hat{\Theta}\,|\,\mathbf{Z}\right]$ as a point from cyan to black with and operator T. The best approximation to the true belief is shown as the red dot inside an error tolerance $\epsilon-$ball.

Consider operator T

$$T:\ \ \mathcal{P} \to \mathcal{P}, \qquad\qquad \hat{\mathbf{M}}_{j|\mathbf{Y}}{}^{(i+1)} = T\,\hat{\mathbf{M}}_{j|\mathbf{Y}}{}^{(i)} \tag{5.24}$$

the mechanism by which the current clique marginal belief $\hat{M}_{j|\mathbf{Y}}{}^{(i)}$ is updated to a more accurate belief $\hat{M}_{j|\mathbf{Y}}{}^{(i+1)}$. Fig. 5-11 illustrates how an approximate function is modified to a point which is "closer" to the actual marginal belief, through the action of some operator T. We are careful about the term "closer", since our choice of belief function significantly influences how close we can get to the true marginal belief.

We want the accuracy of the approximation process to improve based on some tunable computational load parameter $\mathcal{Q}$, as indicated by the measure dissimilarity $d\,(\cdot,\cdot)$,

$$d\left(\mathbf{M}_{j|\mathbf{Y}},\, {}_{\mathcal{Q}}\hat{\mathbf{M}}_{j|\mathbf{Y}}\right) < d\left(\mathbf{M}_{j|\mathbf{Y}},\, {}_{\mathcal{Q}'}\hat{\mathbf{M}}_{j|\mathbf{Y}}\right) \qquad \mathcal{Q}' \prec \mathcal{Q}. \tag{5.25}$$

Therefore, we seek a transformation operator T to modify the belief approximation towards the best approximation of the true belief conditioned on the information $\mathbf{Y}_j$ at clique j, i.e. a contraction mapping

$$d\left(\mathbf{M}_{j|\mathbf{Y}},\, \hat{\mathbf{M}}_{j|\mathbf{Y}}{}^{(i+1)}\right) < d\left(\mathbf{M}_{j|\mathbf{Y}},\, \hat{\mathbf{M}}_{j|\mathbf{Y}}{}^{(i)}\right). \tag{5.26}$$

After obtaining a sufficiently accurate estimate of $d\left(\mathbf{M}_{j|\mathbf{Y}}, \hat{\mathbf{M}}_{j|\mathbf{Y}}\right) \leq \epsilon$, eq. (5.15) marginalizes out frontal variables $\Theta_{F,j}$ as the new message to be passed further up the tree. We note that frontal variables $\Theta_{F,j} \subseteq \Theta_{C,j}$ only contain information from $\mathbf{Y}_j$, which denote all measurements from the current and lower cliques of this branch in the tree.

The statistics and artificial intelligence communities continue to conduct research into posterior estimation, and we mention a few works which have influenced ours [29,71,166,228]. We choose to follow an approximate belief propagation approach with the intention of tracking the dominant modes across the entire solution, but without permanently discarding dormant modes which may become active as more information is gathered.

Propagation of summarizing statistics beyond Gaussian-only parametrics, such as expectation propagation [152], are popular avenues,but statistical moments alone are not well suited for multi-modal densities. Approximating belief functions directly (such as Kernel density estimators, or KDEs) is another more direct approach that we will follow.

The next sections discuss how we choose to approximate the marginal beliefs and how, for algorithm analysis, to project functions onto a point in some high dimensional space. We also present how to benchmark the inference using the Bellman optimality equation. Section 5.4.3 continues with a substitution based sampling algorithm to perform multi-modal inference.

Function Approximation

The fourth computational principle, Table 5.1, involves approximating belief functions to finite accuracy. An ϵ approximation error tolerance from the true marginal density allows the algorithm stochastic freedom along with associated computational benefits. The computational benefits express in two forms: limited dominant modes and approximation of nonparametric belief functions.

A true belief function from space $\mathcal{P}$ may not be continuous and have infinite degrees of freedom. We limit ourselves to approximate belief functions as finite dimensional and continuous subset of all possible belief functions $P \in \mathcal{P}$. The approximated probability density functions in Θ_j as $\left[\hat{\Theta}_j^{(i)} \mid \cdot\right] \in P$ are over the alphabet of real numbers $\mathcal{M}(\Xi)$, where we use $\mathcal{M}$ to indicate all discrete variables are marginalized into the continuous belief functions. We additionally employ a shorthand notation $\hat{p}^{(i)}(\theta_j) = \left[\hat{\Theta}_j^{(i)} \mid \cdot\right]$. All approximate belief functions vanish at infinity and are at least L_2 bounded, which limits the belief that can be expressed

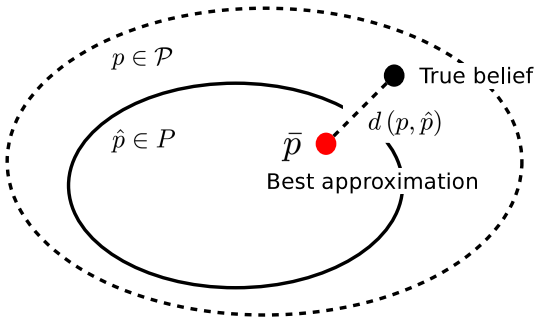


Figure 5-12: Illustration of best function approximation $\bar{p} \in P$ to the true probability density from the considered family of approximating functions $\hat{p} \in P$.

(such as Cauchy distributions).

We use the Pythagoras property of the dissimilarity distance $d(\cdot, \cdot)$,

$$\begin{aligned} d(\hat{p}, \hat{q}) &\geq d(\hat{p}, \bar{p}) + d(\bar{p}, \hat{q}), \qquad \forall \hat{p}, \hat{q} \in P \\ \text{s.t.} \quad \bar{p} &= \underset{\hat{p} \in P}{\operatorname{argmin}}\ d(p, \hat{p}), \end{aligned} \tag{5.27}$$

to identify the best function approximation of the belief $\bar{p} \in P$. Fig. 5-12 illustrates the family of approximating functions and best approximation $\bar{p}$ as the red dot.

In the multi-modal case, we observe that each hypothesis can still be represented as a separate Gaussian term. Furthermore, for non-Gaussian beliefs we can use kernel density estimation [208] to approximate the belief function. We choose to use kernel density estimates for function approximation using a sum of Gaussians

$$\left[\hat{\Theta}_j{}^{(i)} \mid \cdot \right] = \sum_{k=1}^{N} w_{j,k}{}^{(i)} \mathcal{N}\left(\theta_j;\, \mu = \mathbf{x}_{j,k}{}^{(i)},\, \Lambda_{j,k}{}^{(i)}\right) = kde\left(\{x_{j,k}{}^{(i)}\}_{k=1}^{N}\right) \tag{5.28}$$

at the (i)'th iteration of the inference procedure; $x_{k,j}{}^{(i)}$ is the k'th kernel sample position, along with even weighting factor and common bandwidth parameter.

The N limited number of particles, along with evenly weighted particles $w_{j,k}{}^{(i)} = w_j{}^{(i)}$, and common bandwidth parameter $\Lambda_{j,k}{}^{(i)} = \Lambda_j{}^{(i)}$ define our function approximation family of choice. We do however maintain separate bandwidth parameters on each of the individual scalar dimensions $\Lambda_j = [\lambda_{j,1}, \ldots]$. Fig. 5-13 shows a target belief function and two approximations, in this case a mixture of half Beta and Uniform distributions.

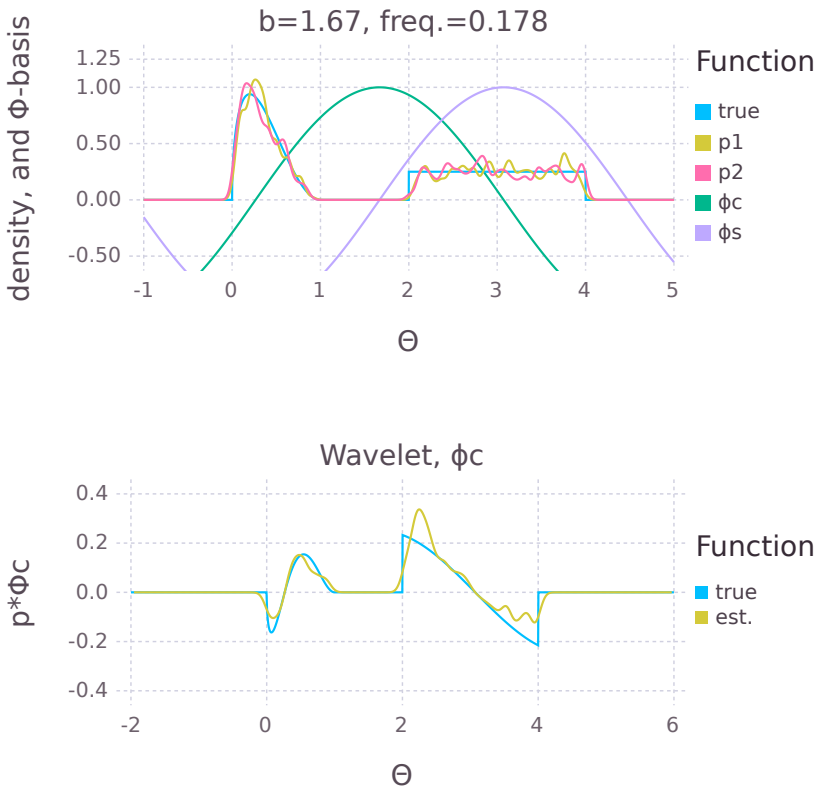


Figure 5-13: *Top*, illustration of approximate probability density function using kernel density estimation, showing a target density and two approximate functions. Two basis functions $\phi_{cos/sin}$ are also shown. *Bottom* shows arbitrary wavelet obtained by modulating and translating the true and approximated belief functions with the basis function ϕ_c.

We use leave-one-out likelihood cross validation (LOOCV), Silverman [208], to compute an appropriate bandwidth Λ for the KDE. We have found that in the multi-modal case many of the heuristic methods drastically over-smooth the bandwidth estimates. To limit the length of discussion here, we refer the reader to Silverman [208] for more details on kernel density estimation and bandwidth selection.

The function approximation introduces a ϵ loss of information relative to the true underlying belief, $d\left(p, \bar{p}\right) \propto \epsilon$. The approximation error diminishes asymptot-

ically as the N number of samples are increased. If required, more accuracy can be obtained by releasing the common bandwidth, or even weighting constraints.

The sum of Gaussian kernels results in an exponential increase of terms as multi-modal likelihood potentials are multiplied together – similar to the discussion surrounding eq. (5.4). However, we can limit the number of kernels in the resulting product and the inference process can instead focus on estimating dominant modes in each marginal. The infinite function product discussion is deferred to Section 5.5.

Hilbert Embeddings with Explicit Basis (Wavelets)

In this next two sections we will take an unusual step, in order to better understand our upcoming inference algorithm. In the previous section we describe how an infinite probabilistic belief function $p \in \mathcal{P}$ is instead approximated by a smooth finite dimensional approximation function $\hat{p} \in P$. Generally speaking we have little insight into how the inference procedure influences the approximated belief function, other than variations in the density distribution.

Instead we will look at feature space projections of the approximated belief functions and evaluate how the inference procedure is progressing in that space. We use the Hilbert space of inner products on infinite functions to project functions onto some basis and then study the projection points. We note that if approximated functions have N degrees of freedom, then projecting the function onto N independent basis functions provides a complete picture of the function in the original space.

We would like a robust method to study the influence of an inference operator T on approximate functions belief $\hat{p} \in P$, which have many degrees of freedom. We can project, with an inner product, over a function's entire domain with a set of orthogonal basis functions $\phi_\cdot$ onto a point in some high dimensional (Hilbert) feature space $\mu_\Theta \in \mathcal{H}_\Xi$:

$$\begin{aligned} \mu_\Theta = \mathbb{E}_p\left[\phi\left(\Theta\right)\right] &= \left\langle\, p\left(\Theta\right),\, \phi_j\left(\Theta\right)\,\right\rangle \\ &= \int_\Xi p\left(\theta\right)\, \overline{\phi_j\left(\theta\right)}\, d\theta = \int_\Xi \overline{\phi_j\left(\theta\right)}\, d\rho\left(\theta\right) \end{aligned} \tag{5.29}$$

where $\overline{\phi}$ represents complex conjugation. A more direct representation of using a probability measure $d\rho\left(\cdot\right)$ could also be used. A Monte Carlo approximation of the embedding from samples $\theta_{j,k} \sim p\left(\Theta\right),\ k = 1, 2, .., N$ weighted by α_i is used in

practical application,

$$\mu_\Theta \approx \hat{\mu}_\Theta = \sum_{k=1}^{N} \alpha_i \, \phi\left(\theta_{j,k}\right). \tag{5.30}$$

This has the effect of reducing the dimensionality of some true density function from space $\mathcal{P}$ down to a limited family of approximations $P \subset \mathcal{P}$ with finite degrees of freedom. We can then represent an approximated function $\hat{p} \in P$ according to a set of independent basis functions ϕ up to the same degree of freedoms as the original approximation. Many operations during the inference process are easier to understand in this feature space, than in the original approximated function space.

The Fourier series is built from a convenient dictionary of infinitely many orthogonal basis functions $\phi_f\left(\theta_i\right) = e^{i\pi f \theta_j}$. We can project our belief functions using a composition of modulation $E_{\omega=2\pi f}$ and centering translation D_b operators, with $b_{j'} = \mathbb{E}_{\hat{p}}\left[\Theta\right]$, and in inner product as

$$\mu_{\theta,j,f} = E_\omega \, D_b \, \hat{p}\left(\Theta_j\right) = \left\langle \, \hat{p}\left(\Theta_j - b_{j'}\right), \, e^{i2\pi f \Theta_j} \, \right\rangle. \tag{5.31}$$

Where individual quadrature frequency coefficients $\mu_{\theta,j,f} \in \mathbb{C}$ are complex. We can view the product $\hat{p} \times \phi$ as producing a wavelet function, since our belief functions vanish at infinity, as shown Fig. 5-13.

The expansion coefficients represent a point in frequency feature space, where low frequencies are less influenced, and high frequencies alias or "oscillate" for subtle changes in the approximated function.

The embedding allows us to consider belief functions as a point moving around in some space, as shown in Fig. 5-11. We draw the analogy to conventional parametric optimization, where variable assignment effectively moves a point around on an objective function surface. Note, we could also reparameterize the high dimensional point to a continuous curve along three dimensions, with sine and cosine quadrature on two and frequency along the third dimension.

For our analysis, we can select frequencies $f \in F$ as a multiple κ according to the standard deviation of the initial belief

$$f = \frac{1}{\kappa \sqrt{\mathbb{E}_{\hat{p}}\left[\left(\Theta_j - \mu_j\right)^2\right]}}, \tag{5.32}$$

while limiting high frequencies to a user-defined alias tolerance. This interval of so

called mid-frequencies F can be used to select a usable set of N' orthogonal basis vectors, $\{\phi_{f,k}\}_{k=1}^{N'}$.

We briefly illustrate point projections of frequency coefficients of our approximating family of functions P in Fig. 5-13. Two orthogonal basis vectors $\phi_{c,f,b}(\theta) = \cos 2\pi f(\theta - b)$ and $\phi_{s,f,b}(\theta) = \cos 2\pi f(\theta - b)$ are also shown in Fig. 5-13, for a particular frequency and mean value such that $\kappa = 4$.

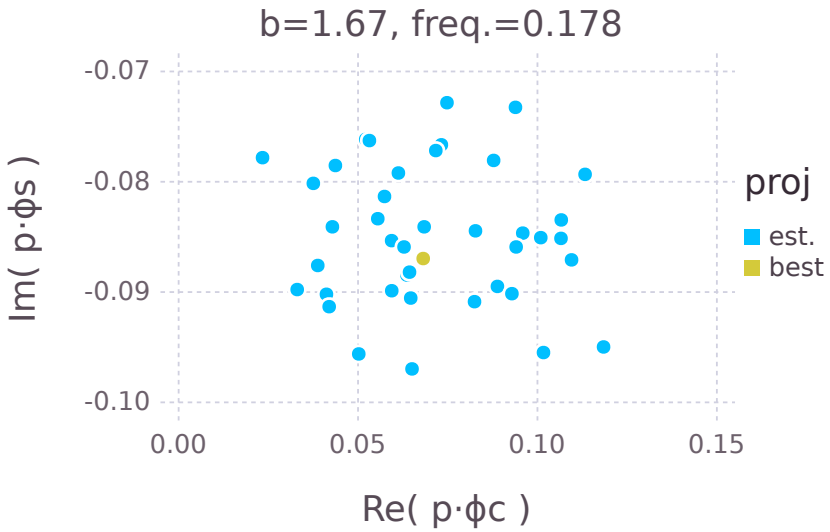


Figure 5-14: Two dimensional projection of fifty approximated belief functions onto orthogonal Fourier series coefficient such that $\kappa = 4$. Basis vectors are shown in Fig. 5-13 as $\phi_.$.

Example projections of many approximated functions are shown in Fig. 5-14, with one special "best" approximation. The best point was calculated from a very high accuracy approximation of the true belief function, i.e. very large number of kernels N.

An illustration of our algorithm, considering functions as high dimensional points, is given in Chapter 6 in Section 6.2. We wish to not limit ourselves with any particular choice of basis functions and rather defer the choice to a fit-for-purpose decision as the situation requires. In the next section, we discuss an extension with specific family of implicit basis functions ϕ (feature maps) known as reproducing kernel Hilbert spaces.

Reproducing Kernel Hilbert Spaces and Implicit Basis

The Reproducing Kernel Hilbert Space (RKHS) method is a different approach to function approximation with mechanics for convolution and product approximations, which are discussed in Sections 5.4.4 and 5.5.4 respectively. In this section we connect function approximation and Hilbert space embedding to using a special class of reproducing kernels, and emphasize the difference to smoothing kernels used for kernel density estimation above.

A reproducing kernel Hilbert space (RKHS) describes functions $f : \Xi \rightarrow \mathbb{R}$ in inner product space $\mathcal{F}$ over domain Ξ as formed by a kernel $k(\theta_i, \theta_j)$, [69,77,214]. The inner product projection in eq. (5.29) is combined with an implicit feature map to form

$$k(\theta, \theta') = \langle \phi(\theta), \phi(\theta') \rangle_{\mathcal{F}}. \tag{5.33}$$

Many kernels $k(\cdot, \cdot)$ satisfy the reproducing property, which we shall call Mercer kernels, owing to Mercer's theorem [77] which states that continuous, symmetric and non-negative definite kernels have an infinite orthonormal basis (eigen functions $\mathbf{e}_k$ and eigen values λ_k):

$$k(\theta_i, \theta_j) = \sum_{k=1}^{\infty} \lambda_k \mathbf{e}_k(\theta_i) \mathbf{e}(\theta_j) \tag{5.34}$$

A kernel satisfying the reproducing property allows

$$\langle f(\cdot), k(x, \cdot) \rangle_{\mathcal{F}} = f(x), \tag{5.35}$$

and implies evaluation of a function f at point x can be replaced with an inner product. As a side note, the frequency series decomposition discussed above does have a Mercer kernel and can therefore be interpreted as a Hilbert embedding [77], and is an example where a basis is selected and an accompanying kernel is found.

The alternative is to choose a convenient kernel, and leave the basis functions as implicit definitions. The squared difference kernel is a common choice

$$k(\theta, \theta') = exp\left(-\sigma \|\theta - \theta'\|^2\right) \tag{5.36}$$

While computing the complete and infinite inner product with some kernel can be very difficult, we can instead use a Monte Carlo approximation of the embed-

ding a mean map (representer theorem), similar to eq. (5.30)

$$\hat{\mu}_{\Theta}(x) = \sum_{i=1}^{m} \alpha_i k(\theta_i, x). \tag{5.37}$$

We defer further discussion of RKHS at this point, and will return when computing convolutions and products of infinite function approximations in Sections 5.4.4 and 5.5.4. With embedding logic in hand, we turn our attention to inference procedure optimality.

Consider the frequency projection of some marginal belief where low frequency coefficients are mostly unchanged, while high frequency coefficients show repeated cycles as the belief moves towards the target distribution. The bulk work therefore seems to be contained in the mid-frequency region where coefficients are moved one point to another. This would suggest we can better focus computation for efficient inference, and return to this argument with multi-scale computations in Section 5.5.1.

5.4.2 Bellman Optimality (Momentum)

We proceed in using functional projections to better understand optimality in our inference procedure. Later, we will show how Bellman optimality arguments can be used to show inefficiencies in our inference procedure. We expect to use the Bellman optimality as a starting point for developing more efficient algorithms in future developments of the general inference procedure.

Fig. 5-15 illustrates how an inference procedure could evolve the marginal belief over some variable in multiple steps. An algorithm could either take a big step towards the desired belief, if the required direction was known, or explore the space with several shorter steps. The figure also illustrates a convergence boundary which may result in chaotic behavior in the inference procedure.

The Bellman equation can be used to benchmark the performance of an inference procedure, which in turn can be used to better design the inference algorithms. The Bellman equation is defined as

$$\begin{aligned} V\left(\hat{p}^{(0)}\right) = \max_{\{a^{(i)}\}_{t=0}^{\infty}} & \sum_{t=0}^{\infty} \beta^i \mathbf{F}\left(\hat{p}^{(i)}, a^{(i)}\right) \\ s.t. \quad & a^{(i)} \in \Gamma(\hat{p}v), \quad \forall t = 1, 2, \ldots, \text{ and } \beta \in [0, 1] \\ & \hat{p}^{(i+1)} = T\left(\hat{p}^{(i)}, a^{(i)}\right), \end{aligned} \tag{5.38}$$

where cost-to-go $\mathbf{F}\left(\hat{p}^{(i)}, a^{(i)}\right)$ is attributed by the current belief state $\hat{p}^{(i)}$ and operator T as influenced by action $a^{(i)}$ from plausible policies $\Gamma\left(\hat{p}^{(i)}\right)$.

Readers familiar with trajectory planning and path planning should recognize the correspondence in the future segment of the factor graph in Fig. 5-1 with the Bellman equation analysis of the inference process here. Multiple policies can be interpreted as multi-modal beliefs in predicted future states.

When ground truth is available, we can use a cost-to-go function which incorporates the discrepancy of the current and target beliefs, $\mathbf{F}\left(\hat{p}^{(i)}, \bar{p}, a^{(i)}\right)$. In general when ground truth is not available, we might consider using direction information from the function point motion. The gradients of mid-frequency coefficients might be useful, and can be considered some form of momentum preservation.

A more direct approach could be to "regularize" the direction of belief function modification, again using the Pythagoras property $\bar{p} = \operatorname{argmin}_{p_\varrho \in P_\varrho} d\left(p, q\right)$, from the family of functions $P_\varrho = \{p_\varrho : \varrho p_0 + (1-\varrho)\, p_1,\ \varrho \in [0,1]\}$. We can then define

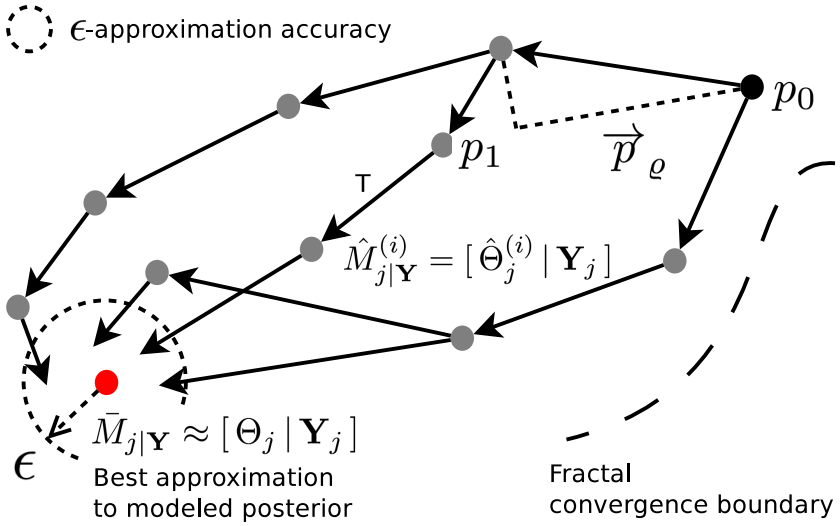


Figure 5-15: Illustration of how a functional point projection can iteratively be moved from an initial state towards the desired belief state as the red dot, and can interpret the process as a cost-to-go requirement for Bellman optimality.

the cost-to-go function as

$$\mathbf{F} = \frac{\alpha}{\mathcal{Q}\left(a_t\right)} \left| d\left(\hat{p}^{(i)}, T\right) \right| - \left| d\left(\overrightarrow{p_\varrho}, \hat{p}^{(i)}\right) \right|$$
$$\text{s.t. } \overrightarrow{p_\varrho} = \underset{p_\varrho \in \mathcal{P}_\varrho}{\operatorname{argmin}} \; d\left(p_\varrho, \hat{p}^{(i)}\right). \tag{5.39}$$

By choosing $p_0 = \hat{p}^{(0)}$ as the initial state and $p_1 = T\left(\hat{p}^{(i)}, a^{(i)}\right)$ as the next state, the inference procedure T is rewarded for taking big steps in the same direction. The computational quality parameter $\mathcal{Q}\left(a^{(i)}\right)$ is also considered, and can be simply linked to the expected computation time to reward faster algorithms.

While our analysis is limited to discrete steps in the functional feature space, it is worth noting that some continuous modification of a belief function would result in a continuous change of the embedding point. It is therefore possible to collapse the Bellman cost-to-go steps into a continuous domain. This insight is important as it allows us to consider smooth gradients of modifications to functional embeddings in the feature space.

Note, the Bellman equations give us a relative benchmarking technique, but no convergence guarantees. We can use the benchmark to evaluate algorithm efficiency on any or all marginals in the Bayes tree structure. We return to the use of the Bellman equation in Section 5.4.5.

5.4.3 Functional Fixed Points and Detailed Balance

In the previous sections, we identified a continuous probability density function approximation scheme, using a restricted form of kernel density estimation. We defined the inference procedure under a function operator T, which would modify approximated beliefs over variables in each clique of the Bayes tree.

In this section we make the connection between stable functional fixed points of the embedded feature space and detailed balance equations from Markov chain Monte Carlo (MCMC) sampling, discussed in the next section. Our intraclique inference operation $T_\mathcal{Q}$ implements nested, block, batch Gibbs sampling as a special case of the Metropolis-Hastings algorithm which satisfies the detailed balance equations [242],

$$\left[\hat{\Theta}_i\right] \left[\Theta_i' \,|\, \Theta_i, \cdot\right] = \left[\hat{\Theta}_i'\right] \left[\Theta_i \,|\, \Theta_i', \cdot\right]. \tag{5.40}$$

The conditional proposal beliefs $\left[\Theta_i' \,|\, \Theta_i, \cdot\right]$ and $\left[\Theta_i \,|\, \Theta_i', \cdot\right]$ are in fact the user

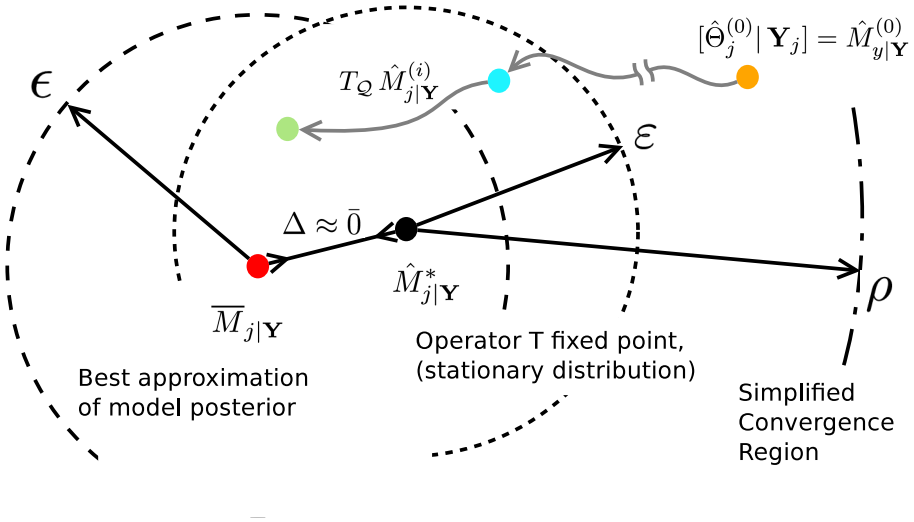


Figure 5-16: Illustration of stochastic functional fixed point for some marginal belief over a variable in Θ_j of true clique posterior $M_{j|\mathbf{Y}}$.

supplied likelihood factors from the factor graph, and $\left[\hat{\Theta}_i\right]$ is the approximated marginal belief over variable Θ_i. By ensuring reversability of these proposal beliefs (a requirement which is pass on to the user), implies that global balance can be found by marginalizing out θ_i' on either side.

Convergence guarantees are subject to the probabilistic model provided by the user via the factor graph, and guarantees are therefore problem specific. We restrict our attention to consistent problem descriptions onlu – consistent and inconsistent models are discussed in more detail in Section 3.6.

We are working our way through the definition of operator $T_{\mathcal{Q}}$ for solving the intraclique Chapman-Kolmogorov transit integral operation, and now need to define how our technique corresponds to a high dimensional (infinite function) fixed point.

Stochastic Fixed Points in Feature Space

We believe there is a connection between stochastic fixed points in an embedding space, that corresponds with stationary distributions in Markov Chain Monte Carlo methods. With the methods developed here, we will in Chapter 6 empirically show how embedded mean mappings converge to a region of the embedding space, an argument that is trivially satisfied by the feature map embedding of a

MCMC chain's stationary distribution itself.

By operator fixed point we mean a value to which some mapping converges. For example, consider the contractive mapping g which counts the number of written letters of the incoming word, and converges to $four$:

$$g\left(ten\right) \rightarrow g\left(three\right) \rightarrow g\left(five\right) \rightarrow g\left(four\right) \rightarrow g\left(four\right) \ldots \tag{5.41}$$

We take the expanded fixed point concept as an infinite function of the inference operator $T_{\mathcal{Q}} : P \rightarrow P$, following from eq. (5.24). If we could prove operator $T_{\mathcal{Q}}$ to be a contraction mapping, see eq. (5.25), we will know that the process will converge to a fixed point in the high dimensional function space. We already know that the true joint posterior marginals have some embedding in the feature space, but we need to understand when the inference operator T will contract to a fixed point which is close to or on the true belief embedding point.

Again, the problem is we do not know ahead of time what the user likelihood functions might be, but we can restrict the user to reversible likelihood models that are only dependent on their current state. Then we can use results for MCMC as a necessary condition for $T_{\mathcal{Q}}$ to be a contractive mapping. We can further define that our approach only uses Gibbs sampling, which is well known to satisfy detailed balance with reversible likelihoods.

We define a stochastically stable functional fixed point $\hat{M}_{j|\mathbf{Y}}{}^{*}\left(\Theta_j\right)$ as

$$\begin{aligned} \hat{M}_{j|\mathbf{Y}}{}^{*}\left(\Theta_j\right) := \mu^{*}_{j|\mathbf{Y}} &= \lim_{i \rightarrow \infty} \mathbb{E}_{M_{j|\mathbf{Y}}{}^{(i)}}\left[\phi\left(\Theta_j\right)\right] \\ &\approx \frac{1}{\Delta I} \sum_{i=I_1}^{I_2} \mathbb{E}_{M_{j|\mathbf{Y}}{}^{(i)}}\left[\phi\left(\Theta_j\right)\right],\ 1 \ll I_1 < I_2 \end{aligned} \tag{5.42}$$

with bounded approximation error ε around the expected value of the marginal belief embedding, i.e.

$$d\left(M_{j|\mathbf{Y}},\ \left(T_{\mathcal{Q}}{}^{i} \hat{M}_{j|\mathbf{Y}}{}^{*}\right)\right) \prec \varepsilon, \qquad \forall i \in \mathbb{N}_{+}. \tag{5.43}$$

Strictly speaking, infinitely many approximations are concentrated around the fixed point and not explicitly less than some fixed bound ε. We intend ε as a sigma multiple of covariance, or high quantile, limit for infinitely many approximations around the fixed point. As before, the quality parameter $\mathcal{Q}$ represents increased granularity/resolution for higher computational availability.

We can visualize the function in the region around the current state using an

orthogonal basis projection $\hat{\mu} = \langle \hat{p}, \phi \rangle$. Fig. 5-16 illustrates a stochastic fixed point $\hat{M}_{j|\mathbf{Y}}{}^{*} \in P$, the best approximation $\overline{M}_{j|\mathbf{Y}} \in P$ and operator $T_{\mathcal{Q}}$'s region of attraction as a simplified ball ρ. An example of function approximation accuracy ϵ is practically shown in Fig. 5-14.

Fig. 5-16 also shows the separation Δ

$$\Delta = d\left(\bar{M}_{j|\mathbf{Y}}, \hat{M}_{j|\mathbf{Y}}{}^{*}\right), \tag{5.44}$$

and is defined as the distance between the clique's best posterior belief approximate $\overline{M}_{j|\mathbf{Y}} = [\,\Theta_j \,|\, \mathbf{Y}_j\,]$ and functional fixed point of $M_{j|\mathbf{Y}}{}^{*}$. We must show that there is no separation between the best approximation and the fixed point, i.e. $\Delta \rightarrow 0$. We turn our attention to understanding the contractive operator $T_{\mathcal{Q}}$, as given in eq. (5.24), and the relationship between the fixed point and best function approximation.

The lingering question is: how do we know that stochastic fixed point $M_{j|\mathbf{Y}}{}^{*}$ actually exists? The question is further complicated by not knowing what the factor graph and likelihood models are. We do know that some joint probability density must exist (maybe not in closed form) since there is physical action to the variables of interest. We can accept that the joint probability density can be embedded as a point in the feature space. Our discussion will now connect with existing Markov Chain Monte Carlo theory to establish at least one operator $T_{\mathcal{Q}}$ that will contract and stabilize around some point in the feature space. The details as to whether that point is actually close to the true joint posterior embedding is highly problem specific.

Our approach will restricting the user into staying as close as possible as possible to the requirements for a Markov random field, where convolution with the user likelihood functions can be performed on current state varialbes and independently sampled noise parameters. By ensuring the likelihood functions are only dependent on their current state and that the likelihood models are reversible, can we satisfy the Kolmogorov criterion and thereby detailed balance. Keeping to this Markov property will let us establish detailed balance. With detailed balance, as discussed in the next section, we are assured that a stationary distribution to each clique level operation exists. As a marginal belief estimated is updated with the progress of the inference algorithm, we expect to see the embedding point estimate to migrate and settle around a stable region in the feature space. Indeed, we will empirically show for various different likelihood models that the mean embedding point does converge to a stable region.

Intraclique Gibbs Sampling (Stationary Distributions)

By example, consider the *Wikipedia: Getting to Philosophy* [240] operator T_W over the space of non-orphaned *Wikipedia* pages, which proceeds to follow one of the first hyperlinks in the article body. Inevitably this process leads to either the *Philosophy*, *Knowledge*, *Science* or *Experience* pages, and stays in an orbit nearby these articles. To quote the article directly (for first link only visiting),

> "As of February 2016, 97% of all articles in *Wikipedia* eventually lead to the article *Philosophy*".

The *Wikipedia* operator T_W establishes a Markov chain, since the probability of visiting any next page, $\left[X^{(i+1)} \,|\, \cdot \right]$ only depends on the current page $X^{(i)} = x^{(i)}$ and local probability of which link will be visited:

$$\left[X^{(i+1)} = x^{(i+1)} \,|\, x^{(0)}, x^{(1)}, \ldots, x^{(i)} \right] = \left[X^{(i+1)} \,|\, x^{(i)} \right] \tag{5.45}$$

By stationary distribution we mean a posterior probability distribution over states visited in state space, given some Markov chain operation defined by operator T.

Our approach for general *sum-product* inference operator T similarly employs a Markov chain inspired by Gelfand [71] and Tanner [221]. The Gelfand and Tanner approaches iteratively approximate marginalized beliefs over each variable. In our case, variables of of a clique $\Theta_{C,j}$.

Successive approximation procedure follows a Gibbs sequence, popularized by [72], and relies on access to all conditional distributions of each variable in clique j. The conditionals act as proposal distributions for transition between states over space Ξ_j.

Consider a clique with frontal variables θ_1, θ_2 and separator variable θ_3. We iterate over all variables, and at each iteration, we wish to draw N samples from the approximated conditional distributions:

$$\begin{aligned}
\{\theta_{j,1,1}, \theta_{j,1,2}, \ldots, \theta_{j,1,N}\}^{(i+1)} &\sim \left[\hat{\Theta}_{j,1} \,|\, \hat{\Theta}_{j,2}^{(i)}, \hat{\Theta}_{j,3}^{(i)}, \mathbf{Y}_j \right] \\
\{\theta_{j,2,1}, \theta_{j,2,2}, \ldots, \theta_{j,2,N}\}^{(i+1)} &\sim \left[\hat{\Theta}_{j,2} \,|\, \hat{\Theta}_{j,1}^{(i+1)}, \hat{\Theta}_{j,3}^{(i)}, \mathbf{Y}_j \right] \\
\{\theta_{j,3,1}, \theta_{j,3,2}, \ldots, \theta_{j,3,N}\}^{(i+1)} &\sim \left[\hat{\Theta}_{j,3} \,|\, \hat{\Theta}_{j,1}^{(i+1)}, \hat{\Theta}_{j,2}^{(i+1)}, \mathbf{Y}_j \right].
\end{aligned} \tag{5.46}$$

By iterating the chain expressed in eqs. (5.46) and (5.54), we obtain a series of samples from the marginals of the approximated clique partial posterior $[\, \Theta_{C,j} \,|\, \mathbf{Y}_j \,]$. After iterating through all clique variables, the process is repeated and

the marginal belief estimates for each of the variables in the clique are updated. One iteration of the batch sampling produces new samples which are then used to approximate a new belief for each variable. The marginal estimates are themselves improved by enforcing all local likelihood factors and incoming message information (per clique) as the Markov chain progresses.

In order to implement the imputation (successive approximation) block, batch Gibbs sampling approach, we need to construct each conditional distribution based on user supplied likelihood functions within each clique of the Bayes tree. Each conditional distribution, as illustrated in eq. (5.46), is formed by taking the required product between local likelihoods, priors and incoming belief message functions involving that variable. Examples of how to construct the different conditional distributions required is presented in the next section.

We will soon discuss how to practically generate the conditional probability density functions from the user-defined likelihoods, but first discuss how to relate the Markov chain to the contractive mapping requirement. We can show the Gibbs style Markov chain has a stationary distribution by satisfying the stronger condition of detailed balance among the conditional proposal distributions. Recall that the user is defining the likelihood factors at run-time.

The detailed balance arguments ensure that the successive approximation Gibbs sampling operator $T_{\mathcal{Q}}$ results in an operation that contracts the mean embedding point toward a stable point in the embedding space. Reversibility of the conditional transition probabilities ensures detailed balance (Kolmogorov's criterion, Kelly Chapter 1 [119]), which ensures global balance and the existence of a stationary distribution.

At this point we do not know if there is just one unique fixed point in some set of possible solutions, however, we expect multiple fixed points to exist given practical compute limitations. Previous work in operator fixed point analysis and random operator theory indicates methods who approximate complicated posteriors should exist. Most notably we cite Bharucha-Reid [20] and Itoh [104] as examples of intensive study in belief approximation approaches.

An empirical convergence test involves monitoring the incremental updates to belief of each variable in the clique. The Markov chain is assumed to have approached the stationary distribution when the dissimilarity between consecutive approximated marginal beliefs drops below the operator quality $\mathcal{Q}$ bound: $d\left(\hat{M}_{j|\mathbf{Y}}{}^{(i+1)}, \hat{M}_{j|\mathbf{Y}}{}^{(i)}\right) \prec \varepsilon_{\mathcal{Q}}$. We do not present a derivation of the $\mathcal{Q} : \varepsilon$ bound relation to here, and refer the interested reader to the analysis done by Tanner and Wong [221].

Regarding Bellman optimality in eq. (5.38), the speed of convergence for $T_{\mathcal{Q}}$

depends on the combination of the cost of one iteration of $T_{\mathcal{Q}}$, and the number of iterations required to satisfy $d\left(M_{j|\mathbf{Y}}, \hat{M}_{j|\mathbf{Y}}{}^{(i)}\right) \prec \varepsilon$.

Optimizations for Nested Intraclique Structure

Previous work on nested junction tree computation [125] is applicable and is the fifth computational principle we address. Each clique partial posterior $M_{j|\mathbf{Y}}(\Theta_{C,j}) = \left[\Theta_{C_j} \mid \mathbf{Y}_j\right]$ retains a certain amount of internal structure. The structure in the joint belief $M_{j|\mathbf{Y}}(\Theta_{C,j}) \in \mathcal{P}$ results from local priors and incoming message densities which are independent.

The independence of some potential functions allows us to improve the intraclique operations and exploit the independence structure. The structure within a clique can be represented with a graphical model, but we will instead use an association matrix to optimize the Gibbs sampling sequence.

An example clique association matrix for clique j with partial posterior $M_{C,j}(\cdot)$ is shown in Table 5.2, including three binary measurement likelihood potentials $\varphi_k(\cdot)$, two priors $\psi_k(\cdot)$, and an incoming density messages $\hat{m}_{u|\mathbf{Y}}(\Theta_3) = \left[\hat{\Theta}_3 \mid \mathbf{Y}_u\right]$. Clique j defines an upward partial posterior belief message $m_{j,|\mathbf{Y}}(\Theta_5)$ over the only separator variable $\theta_5 \in \Xi_5$. Pairwise or singleton (unary) potential functions φ and ψ are described in Section 5.4.4.

To sample from any conditional in a clique, as shown in eq. (5.46), we first inspect the relevant column of the clique association matrix, which relates incoming messages and local potentials to variables of the clique. Each column guides runtime assembly of conditionals. Table 5.2 is carefully chosen to illustrate specific symbolic structure which can be used for expedited clique marginal belief estimation.

For example, message belief $\hat{m}_{3|\mathbf{Y}}(\Theta_6)$ is a separator with no interaction with other clique variables, and can be passed directly as part of the next upward belief message. Variable Θ_4 is computed as the direct product between two incoming messages

$$\left[\hat{\Theta}_4 \mid \mathbf{Y}_j\right] \propto \hat{m}_{1|\mathbf{Y}}(\Theta_4)\ \hat{m}_{4|\mathbf{Y}}(\Theta_4). \tag{5.47}$$

Variables $\Theta_k,\ k = 1, 2, 3, 5$ must be estimated through the iterative Gibbs procedure as illustrated by eq. (5.46), since they are interconnected via binary measurement likelihood functions $\varphi_k(\cdot),\ k = 1, 2, 3$. Conditional transition beliefs for variables $\Theta_k,\ k = 1, 2, 5$ are assimilated from the product of two beliefs each, while

Table 5.2: Example clique association matrix over frontal variables $\theta_k,\ k \in [1..4]$ and separator $k \in [5,6]$; with local binary potentials $\varphi_{j,k},\ k \in [1..3]$, and unary potentials $\psi_k,\ k \in [1,2]$ and incoming messages beliefs $\hat{m}_{u|\mathbf{Y}}(\cdot),\ u \in [1..4]$. Arrows on Θ_2 indicate correspondence with the process outlined in Fig. 5-17. We note potentials can have n-ary dependency.

	$\downarrow$					
	$\Theta_{j,1}$	$\Theta_{j,2}$	$\Theta_{j,3}$	$\Theta_{j,4}$	$\Theta_{j,5}$	$\Theta_{j,6}$
$\varphi_{j,1}$	x	x				
$\varphi_{j,2}$		x	x			
$\varphi_{j,2}$			x		x	
$\psi_{j,1}$	x					
$\psi_{j,2}$			x			
$\hat{m}_{1\|\mathbf{Y}}$				x		
$\hat{m}_{2\|\mathbf{Y}}$					x	
$\hat{m}_{3\|\mathbf{Y}}$						x
$\hat{m}_{4\|\mathbf{Y}}$				x		
		$\uparrow$				

Θ_3 depends on three potentials,

$$\left[\hat{\Theta}_3{}^{(i+1)} \,|\, \hat{\Theta}_{j,2}, \hat{\Theta}_5, Z_j\right] \propto \left[\hat{\Theta}_3 \,|\, \hat{\Theta}_2{}^{(i')}, Z_{23}\right] \left[\hat{\Theta}_3 \,|\, \hat{\Theta}_5{}^{(i')}, Z_{25}\right] [\,\Theta_3\,], \tag{5.48}$$

where $Z = z$ indicates actual measurements made, along with user supplied likelihood potentials φ and ψ. The prime superscript implies possible variation $(i') \to (i)$ or $(i') \to (i+1)$ in Gibbs sampling order.

In order to compute the conditional belief for Gibbs sampling (Chapman-Kolmogorov transit integral), we need to perform two major steps: convolutions with individual likelihood conditionals, and taking the product between infinite belief function approximations. The sections that follow cover each of these aspects.

5.4.4 Conditionals: Approximating Convolutions

In the previous sections we identified how conditional probability densities are used to construct a operator $T_{\mathbf{Q}}$ for estimating the marginal clique partial posterior density distribution function, $\hat{M}_{j|\mathbf{Y}}$. In this section we continue the intraclique op-

eration description with the penultimate part: how to incorporate user likelihood factors as conditional probability functions for computing convolutions between dimensions. Section 5.5 will cover the last part of the inference process, taking the product between infinite functions.

To solve the Chapman-Kolmogorov transit integral using Gibbs eq. (5.46), consider sampling a clique conditional $\theta_{j,2,k} \sim \left[\hat{\Theta}_{j,2} \,|\, \cdot \right],\ k \in [1..N]$, we inspect the Θ_2 column of the clique association matrix, see arrows in Table 5.2, and see associations to beliefs on Θ_1 and Θ_2 via two pairwise likelihoods φ_1 and φ_2, respectively:

$$\left[\hat{\Theta}_{j,2} \,|\, \hat{\Theta}_{j,1}{}^{(i')}, \hat{\Theta}_{j,3}{}^{(i')}, \mathbf{Y}_j, Z_j \right] \propto \left[\hat{\Theta}_{j,2} \,|\, \hat{\Theta}_{j,1}{}^{(i')}, \mathbf{Y}_j, Z_{12} \right] \times \left[\hat{\Theta}_{j,2} \,|\, \hat{\Theta}_{j,3}{}^{(i')}, \mathbf{Y}_j, Z_{23} \right]. \tag{5.49}$$

The expression implies we need to bring existing beliefs $\left[\Theta_{j,k}{}^{(i')} \,|\, \cdot \right],\ k \in [1,3]$ into the space $\theta_{j,2} \in \Xi_{j,2}$. Let's focus on approximating the first term, $\left[\hat{\Theta}_{j,2} \,|\, \hat{\Theta}_{j,1}{}^{(i')}, \mathbf{Y}_j, Z_{12} \right]$, which involves an implicit convolution with the underlying factor graph likelihood potential $[\, Z_{12} \,|\, \Theta_{j,1}, \Theta_{j,2} \,]$.

For generic convolution with a user-defined measurement likelihood model (parametric, or nonparametric, and non-linear), we take a *coordinate relative prior* $\pi(\cdot)$ which will be available during the later belief products process

$$\begin{aligned} [\, Z_{12} \,|\, \Theta_{j,1}, \Theta_{j,2} \,] &\propto [\, \Theta_{j,2} \,|\, \Theta_{j,1}, Z_{12} \,]\ \pi(\Theta_{j,1}) \\ &= \varphi_{j,2|1}(\Theta_{j,1}, Z_{12})\ \pi(\Theta_{j,1}), \end{aligned} \tag{5.50}$$

where $\varphi_{j,2|1}$ indicates the conditional potential function from the factor graph, belonging to clique j. We use a Monte Carlo approximation to the convolution

$$\begin{aligned} \int_{\Xi} [\, \Theta_{j,2} \,|\, \Theta_{j,1}, Z_{12} \,] \left[\hat{\Theta}_{j,1}{}^{(i')} \,|\, \cdot \right] d\theta_1 &\approx \left[\hat{\Theta}_{j,2} \,|\, \hat{\Theta}_{j,1}{}^{(i')}, \cdot \right] \\ &= kde\left(\{\theta_{j,2,k}{}^{(i')}\}_{k=1}^{N \times k'} \right), \end{aligned} \tag{5.51}$$

where samples $\theta_{j,2,k}{}^{(i')} \sim \left[\hat{\Theta}_{j,2} \,|\, \hat{\Theta}_{j,1}{}^{(i')}, \mathbf{Y}_j, Z_{j,12} \right]$ are drawn from (computed

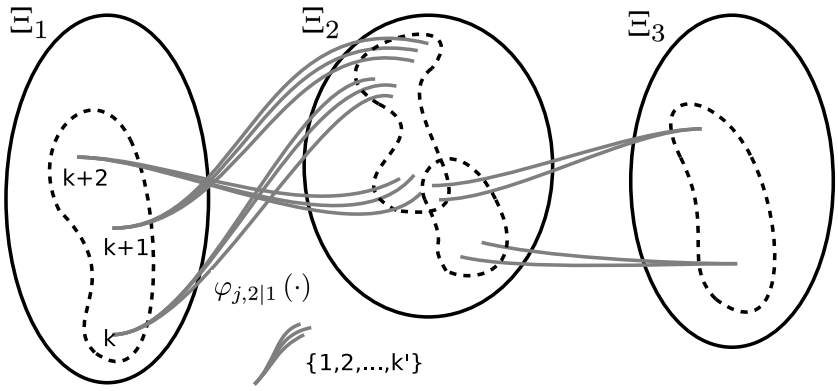


Figure 5-17: Monte Carlo approximation of a convolution between sample approximated belief and conditional likelihood φ, by transmitting points in spaces Ξ_1 and Ξ_3 to Ξ_2, while including i.i.d. sampled points from the measurement noise model. The illustration corresponds to arrows indicating column $\Theta_{j,2}$ in Table 5.2.

with),

$$\begin{aligned} {\theta_{j,2,k}}^{(i')} &= \varphi_{j,2|1}\left(\Theta_{j,1} = {\theta_{j,1,k}}^{(i')}, Z_{12} = \eta_{12,k}\right) \\ &= \varphi_{j,2|1}\left({\theta_{j,1,k}}^{(i')}, \eta_{j,12,k}\right), \end{aligned} \tag{5.52}$$

and $\eta_{12,k}$ is sampled i.i.d. from the noise distribution associated with that measurement likelihood, as discussed in Chapter 3. We note that up sampling to $N \times k'$ points in the convolution result is possible, hence the k' index. This increases the granularity of the convolution approximation process, and is illustrated in Fig. 5-17.

We find $N \times k'$ values of ${\theta_{j,2}}^{(i')}$ by enforcing the user-defined residual function $\delta(\cdot)$, which describes the measurement likelihood model. Note also that the use of limited samples constitutes a further approximation information loss. Larger values of $N \times k'$ will improve estimation accuracy, but also increase computational load.

The most desirable formulation is through a root finding framework which al-

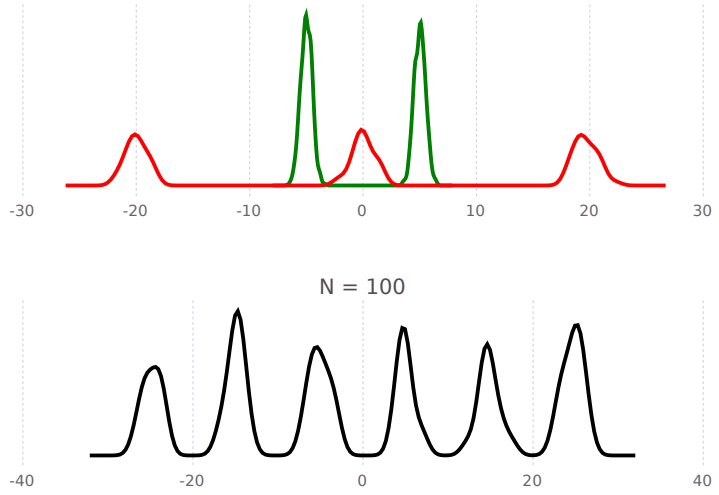


Figure 5-18: Example Monte Carlo convolution approximation between multimodal likelihoods: *Top*, prior belief $\pi(\Theta_1)$ with three modes in red and user-defined bimodal likelihood $[\Theta_2 \,|\, \Theta_1]$ in green. *Bottom*, approximate convolution result showing six modes, using eq. (5.54) and a simple linear dependence $\delta(\theta_2, \eta(\theta_1)) = \theta_2 - \eta,\ \eta \sim \mathcal{N}(\mu = -1^{\gamma} \times 5 + \theta_1, \sigma = 0.5)\,,\ \gamma \sim Cat([0.5, 0.5])$.

lows user-defined non-linear and implicit residuals:

$$\begin{aligned} {\theta_{j,2,k}}^{(i')} &= \varphi_{j,2|1}\left({\theta_{j,1,k}}^{(i')}, \eta_{j,12,k}\right) \\ &= \underset{\theta_{j,2}}{\text{solve}} \quad \left[\delta_{j,12}\left({\theta_{j,1,k}}^{(i')}, \theta_{j,2}, \eta_{j,12,k'}\right) = \mathbf{0}\right]. \end{aligned} \tag{5.53}$$

We generalize the convolution using user likelihood potential expression with index placeholder $_{\mathcal{J}}$

$$\varphi_{\mathcal{J}}\left({\Theta_{\mathcal{J}\backslash k}}^{(i)}, Z_{\mathcal{J}}\right) \doteq \underset{\theta_k \subset \Theta_{\mathcal{J}}}{\text{solve}} \quad \left[\delta_{\mathcal{J}}\left({\theta_{\mathcal{J}}}^{(i)}, Z_{\mathcal{J}}\right) = \mathbf{0}\right]. \tag{5.54}$$

Fig. 5-18 shows an example where two beliefs with three and two modes each are convolved to produce a six-mode belief result. Convolutions with likelihood functions are a source of new modes and is part of the mechanism to promote all measurement modes in the proposal distributions.

With low-dimensional residual functions, we opt for a minimization approach

where user residual is composed with an optimization objective $\mathcal{O}(\delta)$

$$\varphi_{\mathcal{J}}\left(\Theta_{\mathcal{J}\backslash k}{}^{(i)}, Z_{\mathcal{J}}\right) \doteq \underset{\theta_k \subset \Theta_{\mathcal{J}}}{\operatorname{argmin}} \left[\mathcal{O}\left(\delta_{\mathcal{J}}\left(\theta_{\mathcal{J}}{}^{(i)}, Z_{\mathcal{J}}\right)\right)\right]. \tag{5.55}$$

Low-dimensional factors mean that not all dimensions can be transited through that specific factor conditional. Our solution is twofold: Firstly, we try combine factors into fewer factors of higher dimension, which is algebraically equivalent but better focuses computations (less MCMC thinning is required).

In the event unconstrained dimensions cannot be avoided, we randomize which variables are optimized by eq. (5.54) or (5.55), and introduce uniform noise when applicable. This approach increases the correlation between successive Gibbs iterations, which is overcome with simply increasing the iterations of operator $T_{\mathcal{Q}}$.

Unary factors (priors) are considered a special case and assumed to introduce information though direct sampling of the underlying distribution function $\psi_j(\Theta_{j,k}; Z_{j,k})$.

Conditionals with Kernel Hilbert Embeddings

The convolution operation is closely related to a series of inner products in some Hilbert space. This section follows from the reproducing kernel Hilbert spaces (RKHS) discussion in Section 5.4.1, as an alternative method to approximate arbitrary functional convolution operations which are associated with conditional probability densities, such as eq. (5.48).

In the previous section we discussed how to approximate a convolution operation with run-time defined user likelihood functions, and remains an important method for transmitting belief to unknown parts of the range space. Recent work by Song et al. [214] on conditional kernel embeddings claim a method of approximating the convolution at much higher computational speed than competing methods.

It is important to understand the covariance embedding between any two variables. We briefly reduce notational burden to just variables Θ_i, Θ_j. Fukumizu [69] defines a covariance operator $\mathcal{C}_{\Theta_i,\Theta_j} : \mathcal{F} \to \mathcal{F}$ for functional embeddings in kernel Hilbert space $\mathcal{F}$. The covariance operator allows us to compute the expected value of a product between functions directly as a linear matrix operation:

$$\mathbb{E}_{\Theta_i,\Theta_j}\left[f(\Theta_i)\, g(\Theta_j)\right] = \langle f, \mathcal{C}_{\Theta_i,\Theta_j}\, g \rangle_{\mathcal{F}} \tag{5.56}$$

The covariance operator is computed by tensor product $\otimes$, of the feature map $\phi(\cdot)$ $\mathcal{C}_{\Theta_i,\Theta_j} = \mathbb{E}_{X_s X_t}[\phi(\Theta_i) \otimes \phi(\Theta_j)]$. As an extension of the direct mean embedding from eq. (5.30), using evenly weighted samples $\{(\theta_{i,1}, \theta_{j,1}), ..., (\theta_{i,m}, \theta_{j,m})\}$, we can approximate the covariance embedding with:

$$\mathcal{C}_{\Theta_i,\Theta_j} \approx \hat{\mathcal{C}}_{\Theta_i,\Theta_j} = \frac{1}{m} \sum_{k=1}^{m} \phi(\theta_{i,k}) \otimes \phi(\theta_{j,k}) \tag{5.57}$$

Based on the Schur complement [69], these expressions allow us to compute the expectation of a conditional density:

$$\begin{aligned} \mathbb{E}_{\Theta_i|\theta_j}[f(\Theta_i)] &= \langle f, \mathbb{E}_{\Theta_i|\theta_j}[\phi(\Theta_i)] \rangle_{\mathcal{F}} \\ &= \langle f, \mathcal{C}_{\Theta_i,\Theta_j} \mathcal{C}^{-1}_{\Theta_j,\Theta_j} \phi(\theta_j) \rangle_{\mathcal{F}} \\ &= \langle f, \mathcal{C}_{\Theta_i|\Theta_j} \phi(\theta_j) \rangle_{\mathcal{F}} \\ &= \langle f, \mu_{\Theta_i|\theta_j} \rangle_{\mathcal{F}}, \end{aligned} \tag{5.58}$$

where we have used the shorthand $\mathcal{C}_{\Theta_i|\theta_j} = \mathcal{C}_{\Theta_i\Theta_j}(\mathcal{C}_{\Theta_j\Theta_j})^{-1}$.

The significance of eq. (5.58) is an implicit convolution operation, $[\Theta_i^{\pi} | \cdot] \propto \int [\Theta_i | \Theta_j] \left[\Theta_j^{\pi}\right] d\theta_j$. Focusing on the conditional mean mapping of an entire functional embedding

$$\begin{aligned} \mu^{\pi}_{\Theta_i|\Theta_j} = \mathbb{E}_{\Theta_j}\left[\mathcal{C}_{\Theta_i|\Theta_j}\phi(\Theta_j)\right] &= \mathcal{C}_{\Theta_i|\Theta_j}\mathbb{E}_{\Theta_j}[\phi(\Theta_j)] \\ &\approx \mathcal{C}_{\Theta_i|\Theta_j}\, \hat{\mu}^{\pi}_{\Theta_j}, \end{aligned} \tag{5.59}$$

where we construct the "input" embedding using existing samples from the distribution π. By expanding the tensor products and using a proper kernel $k(\theta_i, \theta_j) = \langle \phi(\theta_i), \phi(\theta_j) \rangle$, such as the square exponential $k(,) = \exp(-\sigma\|\theta_i - \theta_j\|^2)$, a matrix expression is found [214]

$$\hat{\mu}^{\pi}_{\Theta_i|\Theta_j}(x) = \mathcal{C}_{\Theta_i|\Theta_j}\, \hat{\mu}^{\pi}_{\Theta_j} = {\Phi_i}^T (\mathbf{G}_{ij} + \lambda \mathbf{I})^{-1}\, \mathbf{G}^{\pi}_{ij} \mathbf{a}^{\pi}, \tag{5.60}$$

where the Gram matrix elements are based on training samples $\{(\theta_{i,1}, \theta_{j,1}), ..., (\theta_{i,m}, \theta_{j,m})\}$ is $G_{ij} = k(\theta_{i,k}, \theta_{j,k})$. The second Gram matrix $G^{\pi}_{ij} = k\left(\theta_{i,k}, \theta^{pi}_{j,k}\right)$ uses samples from "input" embedding $\hat{\mu}^{\pi}_{\Theta_j}$, and a weighting vector from "input" samples $\mathbf{a}^{\pi} = [\alpha_1, .., \alpha_k]$ (see eq. (5.37)). The transposed leading vector is $\Phi_i^T(x) = \left[k(\theta_{i,1}, x) \;\; \ldots \;\; k(\theta_{i,m}, x)\right]$. Finally, from experience [69, 214] a regularization (hyper-) parameter λ is used to prevent overfitting and produce a smooth estimate. We direct the reader to Song et al. [214] for more

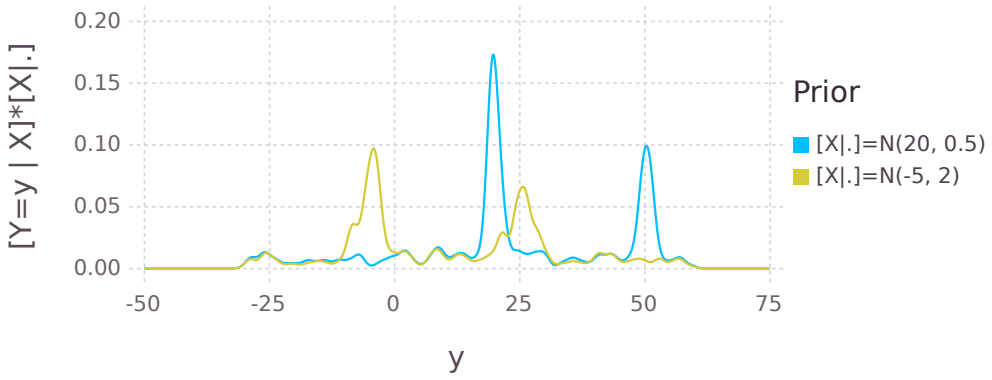


Figure 5-19: Reproducing kernel conditional density embedding demonstration. Reproducing kernel conditional density embedding demonstration of $[\,Y \,|\, X\,]$, given sample points from a bi-modal joint density which transmit belief from a a normally distributed prior belief $[\,X \,|\, \cdot\,]$. Transmitted beliefs for two priors is shown, and conditional embedding support is visible from $(-30, 60)$.

many more details on reproducing kernel Hilbert space embeddings.

For the purpose of our further discussion, an example conditional embedded convolution is shown in Fig. 5-19. A normally distributed prior belief $[\,X^{\pi} \,|\, \cdot\,]$ is used with a conditional belief $[\,Y \,|\, X\,]$ which introduces a new bi-modal uncertainty. The conditional belief is trained with samples from a user-defined function which either does or does not introduce a bias of 30, $y = x + \gamma \times 30,\ \gamma \sim Cat\,([0.5, 0.5])$. Notice, however, that the region of support is visible from around $[-30, 60]$, outside which the convolution will be inaccurate. The region of support is dictated by the training samples used for the conditional embedding.

In robotic navigation, the user defines the structure of the joint distribution at run-time, resulting in a completely unknown posterior distribution, see eq. (5.1). We do not have access to the posterior joint distribution prior to inference, which is critically assumed and used in [214]. The conditional embedding approach involves approximating conditional beliefs based on available samples over the likelihood model.

The successive approximation inference procedure described above migrates to a small region of the state space, where the conditional kernel embedding can be approximated for useful convolution approximation, as illustrated in Fig. 5-17. We can use a direct numerical method in the vicinity of interest around the current

state, to produce training samples for calculating the conditional embeddings.

5.4.5 Common-Mode $\epsilon/\varepsilon/\Delta$-Tolerance in Bayes Tree Depth

Thus far, we have focused on approximating the stationary distribution of each clique $M_{j|\mathbf{Y}}$ with some error tolerances ϵ_j and ε_j, as illustrated in Fig. 5-16. Understanding how the error tolerances accumulate across the Bayes tree during belief propagation is important, and we briefly consider the propagation of error across the Bayes tree in this section.

Consider the error tolerance surrounding the target distribution of some clique marginal and propagate the $\epsilon/\varepsilon/\Delta$-tolerance up the tree, as though a message. Each parent clique introduces additional $\epsilon/\varepsilon/\Delta$ slop in the next marginal estimate based on incoming message beliefs.

The increase in $\epsilon/\varepsilon/\Delta$ error is counter-acted by parallel branches in the tree. A parent clique with multiple children will have a stationary distribution as the aggregate mean between incoming messages. Differential mode errors between the children will cancel out to some extent, while common-mode errors among all cliques will become more entrenched.

The downward belief message pass would work to remove differential mode errors from all marginal beliefs. This point has further importance, in that multiple up and down passes on the tree would be required to remove differential-mode errors from all marginal belief estimates. Common-mode errors among cliques would go unchanged beyond the first upward-downward message passing operations.

Using the Bellman equation (5.38), we see that global benefit is had when constraining the leaves of the tree to higher accuracy, while reducing to lower computational accuracy at the root. This would reduce computation time with few branches near the root, but maximize accuracy in wide parallel computation near the leaves. The Bellman equations allow us to therefore effectively aid in decision making during algorithm design.

5.4.6 Importance of Consensus

The seventh computational principle is in consensus, where decay of unlikely modes occur during the product operation between proposal beliefs, see eq. (5.14). The convolution operation, illustrated graphically in Fig. 5-18, introduces new modes to the proposal distributions, while the product operation between different proposal distributions produce only limited consensus among the beliefs, as

illustrated in Fig. 5-17. We call this *consensus in belief*. Computations are better spent on consensus modes, while lower likelihood modes are only recovered at higher computational cost $\mathcal{Q}$.

The traces in Fig. 5-20 represent proposal beliefs from sensor measurements over some variable Θ in the same space Ξ, in accordance with the illustration in Fig. 5-17. Now we take the product between proposal beliefs in Θ. Their true product posterior, shown in black, represents the joint probability over the given likelihood functions. Notice that six modes are possible in the product between the green and red traces, however, the product only has two meaningful modes. The other modes are unlikely and can be reasonable ignored, which reduces the computational requirements.

The sky blue trace in Fig. 5-20 is an approximated version of the true posterior probability density shown in black. In Section 5.5 we discuss the next major computational principle (see Table 5.1), which involves reducing computational complexity associated with taking products between infinite functions.

The idea is that measurement likelihoods have some consensus about real world events which translates into concentrated belief on only a few modes in each of the marginals. This approach is a cornerstone of our method and allows us explore the *range space of possible multi-modal solutions* with limited practical computation resources. The number of modes recovered by the inference procedure should only be limited by computational resources.

Consensus repeats during belief propagation over the Bayes tree from clique to clique. Iterations of inference operator $T_{\mathcal{Q}}$ may see different modes as influenced by the available information. The concept extends to multiple up and down message passing iterations. Modes initially ignored during the upward message passing may well be re-introduced from other cliques during the downward message passing phase.

5.5 Methods for approximate density products

The last major required component in computing the Chapman-Kolmogorov integral, eqs. (5.14) & (5.15), is the ability to take the product between approximate belief functions in the same function space. In our case we will be taking the product between run-time discovered proposal beliefs for batch Gibbs sampling, as shown in eq. (5.46). In this section we briefly review existing function space product approximation methods, and note several communities that have been developing various techniques in this area.

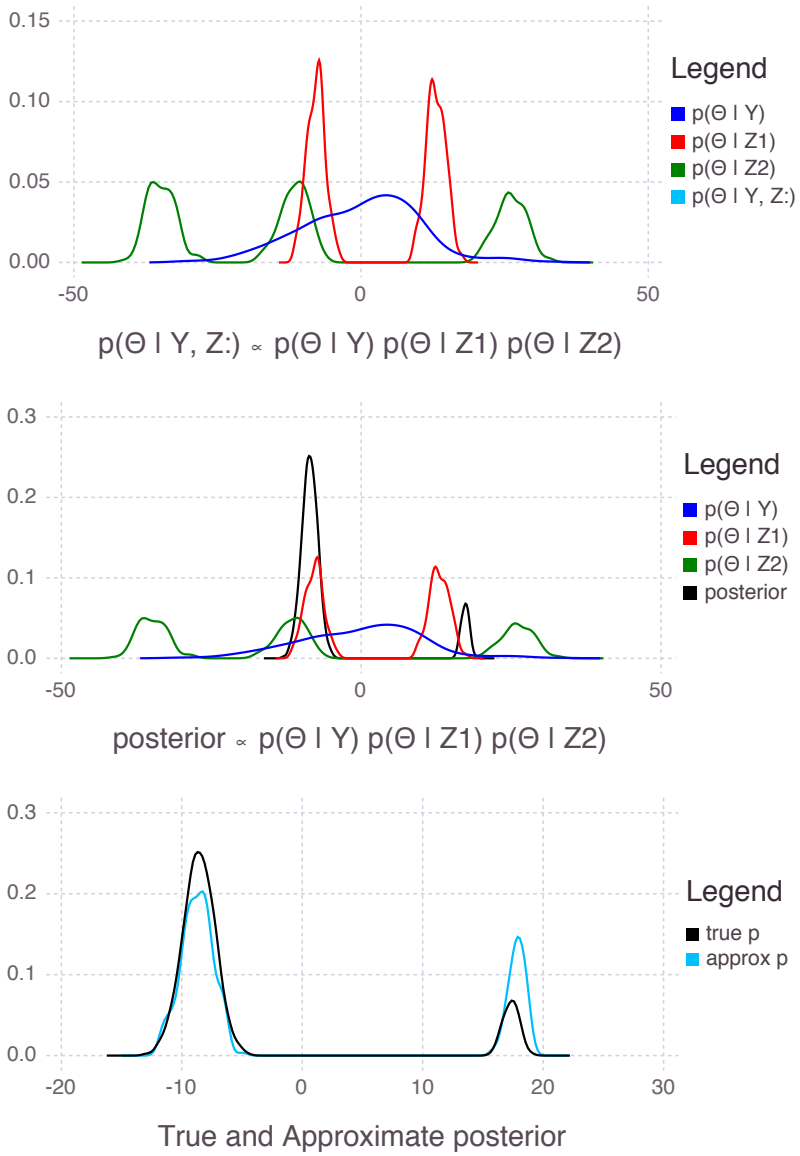


Figure 5-20: Consensus in approximated posteriors when multiplying several infinite belief functions together. True posterior is shown in black, while an approximated multi-modal posterior is shown in sky blue. Only two of the six possible modes are likely to occur in the product, which we call consensus. Approximate posterior product computed with product method from [218].

To reiterate, we need to compute the product between multiple multi-modal (nonparametric) proposal beliefs following a convolution operation, where the low likelihood modes are dropped according to computational resource restrictions. The clique frontal variables are marginalized out of the partial posterior result, and new messages are formed to propagate on the Bayes tree, see eq. (5.15). In this section we discuss various product approximation strategies to complete the Chapman-Kolmogorov intraclique operation.

We will briefly reference the major known techniques, namely: (i.) Importance weighting with resampling (a form or rejection sampling); (ii.) Nonparametric Belief Propagation (NBP), [217]; (iii.) Particle Belief Propagation (PBP) [100]); (iv.) Progressive Bayes: a homotopy type ODE product [204]; and (v.) Kernel Belief Propagation (KBP) [214, 216]).

Importance weighting with resampling, as used in particle filtering, is the simplest method of approximating the product between two infinite functions. Consider evaluating the product at a set of points $\{x_k\}$, such that $f(x_k) \times g(x_k) = \alpha_k \times g(x_k)$, where $\alpha_k = f(x_k)$. We can similarly approximate the product of many terms through importance weighting [45], however, as dimensionality and terms increase we need increasing numbers of sample points to accurately represent the product space. We can use resampling to reduce the number of samples, but would have to reject many small weight samples for an accurate representation. As a result importance sampling over-simplifies the product operation, resulting in random guessing of important regions in the product space. We will pursue a different avenue instead.

5.5.1 Nonparametric Belief Propagation

Sudderth and Ihler [101, 217, 218] developed an efficient method of directly computing the product between multiple kernel density estimates (KDEs) through multi-scale Gibbs sampling. Their method employs an internal KD-tree representation for kernel density approximation and multi-scale sampling. The product between approximate densities is computed using Gibbs sampling, but iterations are staged across increasingly finer and finer scales to promote sampling from all modes.

NBP product operation assumes incoming densities $\left[\hat{\Theta}_i \mid \cdot\right]$ are kernel density estimates, Section 5.4.1, and each has the same number N of Gaussian mixture components, eq. (5.28). The goal is to draw a set number of samples from the

product of d mixtures,

$$\left[\hat{\Theta}_j \,|\, \cdot\right] \propto \prod_{i=1}^{d} \left[\hat{\Theta}_j \,|\, \cdot\right]_i . \tag{5.61}$$

For comparison, if sampling is done from the exact product with N^d Gaussian mixture components, we would need to compute the parameters for each component $L = [l_1, .., l_d]$ similar to an earlier discussion at eq. (5.4). We compute the product between Gaussian mixture components with

$$w_L = \frac{\prod_{i=1}^{d} w_{l_i} \mathcal{N}\left(\theta_j; \mu_{l_i}, \Lambda_i^{-1}\right)}{\mathcal{N}\left(\theta_j; \mu_{l_L}, \Lambda_L^{-1}\right)}, \qquad \Lambda_L = \sum_{i=1}^{d} \Lambda_i, \qquad \Lambda_L \mu_L = \sum_{i=1}^{d} \Lambda_i \mu_{l_i} \tag{5.62}$$

where μ_L, Λ_L^{-1} are the mean and variances of one component L of the product.

A Gibbs sampling strategy over the indices of incoming mixture components is used to select individual components in the product to sample from. Working backwards, to draw a single sample from an individual mixture component in the exact product of N^d mixtures, we understand the product mixture originates from the specific selection of d components from each incoming mixture of N components. We use Gibbs sampling to stochastically select the combination of d incoming mixture components, and then compute them according to eq. (5.62). This individual component of the product mixture is used to sample a new independent variable θ_j sample from the posterior product. This process asymptotically draws samples from the product of incoming densities.

The label Gibbs sampling process starts (so that we can pick the next label l_j) by randomly selecting $d-1$ labels of the incoming mixture densities $\{l_i\}_{i \neq j}$:

$$\begin{aligned} l_1^{(1)} &\sim \left[l_1 \,|\, l_2^{(0)}, l_3^{(0)}, ..., l_d^{(0)} \right] = Cat\left([\rho_1, \rho_2, ..., \rho_N]_{k=0}\right) \\ l_2^{(1)} &\sim \left[l_2 \,|\, l_1^{(1)}, l_3^{(0)}, ..., l_d^{(0)} \right] = Cat\left([\rho_1, \rho_2, ..., \rho_N]_{k=1}\right). \\ &\vdots \end{aligned} \tag{5.63}$$

Each of the sampling weights $\{\rho_i\}_{i=1}^{N}$ is computed by normalized weighting of each mean of the N components in mixture j against the quasi-product

$$\rho_{i,k} = \mathcal{N}_{p_j}^{(k)}\left(\mu_{l_i}; \mu_{L_j}, \Lambda_{L_j}^{-1}\right) \tag{5.64}$$

of input densities defined by $L_j = \{l_i\}_{i \neq j}$. The quasi-product is computed with the $d-1$ conditional components, see eq. (5.62):

$$\mathcal{N}_{p_j}^{(k)}\left(\mu_{L_j}, \Lambda_{L_j}^{-1}\right) = \prod_{i \neq j} \mathcal{N}\left(\mu_{l_i}, \Lambda_i^{-1}\right). \tag{5.65}$$

Lastly note, incoming mixtures are assumed to have equal weight components $w_{l_i} = \frac{1}{N}$, $\forall i \in [1, N]$.

The approach described above requires a large number of Gibbs iterations to thoroughly explore multiple mode beliefs – precisely the type of beliefs we are interested in. Ihler et al. [101] extends the Gibbs method to multiple scales using a KD-tree representation, where component means of a kernel density estimate are split based on their direction of largest variance. The binary tree structure develops a depth of $\log_2 N$, where the leaves represent the all the samples used to construct the kernel density estimate, eq. (5.28).

Accurately approximating the true product density is accelerated by starting with coarser scale approximate models at the root of the KD tree, and then allowing the Gibbs process to more rapidly explore the concentrated regions of the state space moving towards the leaves in the tree. Kernel density estimation with Gaussian models simplifies the coarse to fine transition approach, since all beliefs are conjugate. We invite the reader to follow Ihler et al. [101] for more details.

Lastly, sufficient samples from the posterior product multi-scale Gibbs process are used to assemble a smooth kernel density estimate belief approximation, eq. (5.28). The bandwidth parameter is recovered using leave-one-out-likelihood cross validation [208], as other heuristics tend to oversmooth multi-modal beliefs, see Section 5.4.1 for more details.

Density Products on Manifolds

We can extend the NBP approach to include density products on manifolds. We focus on the 3D rotation manifold extension of eq. (5.62), where rotation $\mathbf{R} \in SO(3)$ is a Lie group element [32]. The process is illustrated in Fig. 5-21.

Consider taking the product between two bodies in world rotation components $[\,{}_b^w\mathbf{R}_1 \,|\, \cdot\,] \times [\,{}_b^w\mathbf{R}_2 \,|\, \cdot\,]$, resulting in a single element $\left[\,{}_b^w\hat{\mathbf{R}} \,|\, \cdot\,\right]$ given the inputs are conjugate pairs. We can use the exponential parameterization of rotation elements, i.e. Lie algebra [32], to define the product space. We implicitly take the exponential

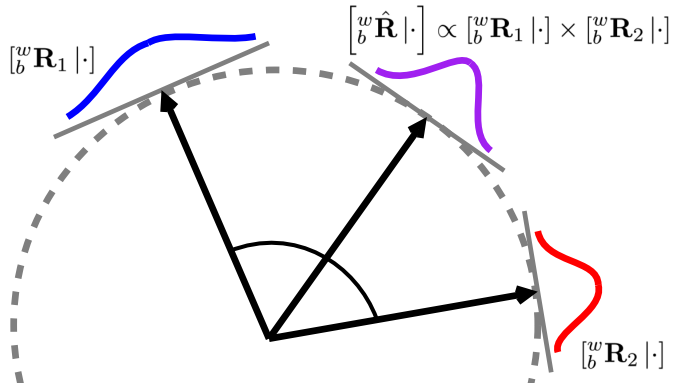


Figure 5-21: Illustration of density product on rotation manifold of parametric Gaussian product in eq. (5.62).

expansion point about the resulting product $\left[{}^{w}_{b}\hat{\mathbf{R}}\,|\cdot\right]$, which is initially unknown

$$ {}^{w}_{b}\hat{\mathbf{R}} = \exp\left[{}^{w}_{b}\hat{\psi}_{\times}\right], \tag{5.66} $$

where $\left[{}^{w}_{b}\hat{\psi}_{\times}\right] \in so\,(3)$ is the corresponding Lie algebra element. We compute the product density (eq. (5.62)) iteratively from a midway a starting point

$$ \Lambda_L \, {}^{w}_{b}\hat{\psi}^{(k+1)} = \sum_{i=1}^{d} \Lambda_i \, {}^{w}_{b}\Delta\psi_i^{(k)}, \tag{5.67} $$

where ${}^{w}_{b}\Delta\psi_i^{(k)} = \log_{SO(3)}\left({}^{w}_{b}\mathbf{R}_i{}^{T}\,{}^{w}_{b}\hat{\mathbf{R}}\right)$, and Λ_i^{-1} is the covariance of the exponential expansion about each individual $\left[{}^{w}_{b}\mathbf{R}_i\,|\cdot\right]$ component. The logmap is defined in eq. (4.23)

5.5.2 Particle Belief Propagation

Ihler et al. [100] later proposed a direct particle belief propagation (PBP) scheme which is slightly more efficient than NBP at solving the Chapman-Kolmogorov transit integral equations give in eqs: (5.14) and (5.15). Ihler's proposal is an importance sampling based particle belief propagation step, where the convolution

step (Section 5.4.4) is used as a proposal distribution $W(\cdot)$:

$$m_{j|\mathbf{Y}}(\Theta_{S,j}) = \mathbb{E}_{\theta_j \sim W_j}\left[\varphi_j(\Theta_{C,j}) \frac{\psi_j(\Theta_{C,j})}{W_j(\Theta_{C,j})} \prod_u m_{u|\mathbf{Y}}(\Theta_{S,u})\right], \tag{5.68}$$

which is approximated as

$$\hat{m}_{j|\mathbf{Y}}{}^{(i)} = \frac{1}{n}\sum_{k=1}^{n} \varphi_j\left(\Theta_{C,j}{}^{(k)}, \Theta_{C,j}{}^{(i)}\right) \frac{\psi_j\left(\Theta_{C,j}{}^{(k)}\right)}{W_j\left(\Theta_{C,j}{}^{(k)}\right)} \prod_u \hat{m}_{u|\mathbf{Y}}\left(\theta_j^{(k)}\right), \tag{5.69}$$

where samples $\Theta_{C,j}^{(i)}$ are drawn from a sampling (weighting or proposal) distribution at node s.

The highest computational expense comes from sampling the product of messages. To reduce computational load, Ihler suggests the *belief-based* sampling approach:

$$m_{j|\mathbf{Y}}(\Theta_{S,j}) = \sum \psi_j(\Theta_{C,j}) \frac{\varphi(\Theta_{C,j}) \prod_u m_{u|\mathbf{Y}}(\Theta_{C,j})}{m_{j|\mathbf{Y}}(\Theta_{C,j})}, \tag{5.70}$$

where computations are reduced by storing and updating belief values (the numerator). In contrast, NBP draws new samples to represent the product density, while PBP uses existing sample points and avoids the need to smooth the sample set. We refer the reader to Ihler et al.'s work [100] for more in depth details about particle belief propagation.

5.5.3 Progressive Bayes: a homotopy type ODE product

Another method to take the product between KDEs, presented by Schrempf and Hanebeck equation [204], is Optimal Mixture Approximation of the product of Mixtures whereby the product is converted to a vectorized first order ordinary differential. A homotopy method is used to parameterize the product, using scalar variable γ, and modifying all local potentials except one:

$$M_{j|\mathbf{Y}}(\Theta_{C,j}, \mathbf{Z}, \gamma) \propto \psi_1(\Theta_{C,j,1}, Z_{j,1}) \prod_{i=2}^{C_j} \psi_i(\Theta_{C,j,i}, Z_{j,i}, \gamma) \prod_{j}^{\cup C_U} m_{u|\mathbf{Y}}(\Theta_{S,j}, \mathbf{Y}_u, \gamma) \tag{5.71}$$

such that, where ϵ is a small number, $\psi(\gamma = 0) = 1$ and $\psi(\gamma = 1) = \psi$:

$$\psi = \exp\left(-\frac{1}{2}\frac{(\theta-\mu)^2}{\left(\sigma\frac{1+\epsilon}{\gamma+\epsilon}\right)^2}\right) \tag{5.72}$$

The progressive Bayes technique then calls for a distance measure between the exact product and the approximation:

$$G = \int_{\mathcal{R}} \left(M_{j|\mathbf{Y}}(\Theta_{C,j}, \mathbf{Y}_j, \gamma) - \hat{M}_{j|\mathbf{Y}}(\Theta_{C,j}, \mathbf{Y}_j, \eta)\right)^2 dv, \tag{5.73}$$

and an optimal solution against η, which is analytically obtained by solving $\frac{\partial G}{\partial \eta} = 0$. Surprisingly, the resulting equations resemble that of the Kalman filter covariance updates and the product operation is transformed into

$$\mathbf{P}(\eta, \gamma)\frac{\partial \eta}{\partial \gamma} = \mathbf{b}(\eta, \gamma). \tag{5.74}$$

Through algebra we can arrive at a mechanism to numerically solve this differential equation, with built in hooks by which error bounds can be maintained. If the resulting product density exceeds a user-defined error tolerance, more kernels are introduced to the product and the integration process is continued. If low probability kernels are found they are discarded from the solution maintaining a lower kernel count in the resulting product density.

The derivation by [204] maintains the nonlinear nature of the product, where the initial Progressive Bayes model called for a Taylor expansion. The derivation is presented for the scalar case, but naturally lend itself to higher dimensional analysis. We refer the reader there for more details.

5.5.4 Kernel Belief Propagation

In Section 5.4.1, we introduced the reproducing kernel Hilbert space approach for approximating marginal beliefs, and in Section 5.4.4 we described the conditional embedding approach that allows matrix operation based approximation of convolution operations. The final part for the *sum-product* inference technique is how to take products of infinite functions with, which can be done with kernel embeddings in multiple ways. In this section we specifically look at work from Song et al. [214] on kernel belief propagation (KBP).

As explained in the convolution section, the conditional kernel embedding requires training data to "support" operations in a region of the state space. The method cannot be blindly used for estimating the posterior in unknown region of the state space. However, research into multi-linear algebra, as used by RKHS, is likely a good avenue for future research. The methods we present surrounding RKHS can potentially be used to speed up computation in known regions of the state space, once our base line method has converged to a stable region of the state space.

The KBP method does not sample new points in the represented product density. Further work, such as [164] and [17] have looked at constructing posterior sampling methods which operate directly in the feature space. A combination of feature space sampling and KBP would result a new and powerful sampling scheme.

The key idea behind Song et al.'s algorithm [214,216] is that reproducing kernel Hilbert space (RKHS) functions are used to express the messages between pairs of nodes. Outgoing belief messages are computed from incoming messages and local potentials, as shown in eq. (5.14), but use linear matrix operations instead of a nested Gibbs type sampling approach we presented in Sections 5.4.3 and 5.5.1. Furthermore, the KBP method operates directly in the embedded feature space, exploiting the reproducing kernel property, $\langle f(\theta), k(x,\theta) \rangle = f(x)$.

We can compute the Chapman-Kolmogorov transit integral, eqs. (5.14) and (5.15), for local clique variables Θ_j and parent clique variables Θ_t:

$$m_{j|\mathbf{Y}}(\Theta_{S,j}) = \mathbb{E}_{\Theta_t|\theta_j}\left[\prod_u m_{u|\mathbf{Y}}(\Theta_u) \prod_k \psi_k(\Theta_j)\right] \tag{5.75}$$

with child clique variables Θ_u and local singleton potentials ψ_j. To reduce notation, we fold local potentials in as though a message and the product of children and local potentials $\prod_{u,k}$. The product of singletons and messages are computed as

$$\begin{aligned}\prod_{i=\{u,k\}} m_{i|\mathbf{Y}}(\Theta_i) &= \prod_{i=\{u,k\}} \langle m_{i|\mathbf{Y}}, \phi(\Theta_i) \rangle_{\mathcal{F}} \\ &= \langle \bigotimes_{i=\{u,k\}} m_{i|\mathbf{Y}}, \bigotimes_{i=\{u,k\}} \phi(\Theta_i) \rangle_{\mathcal{F}}\end{aligned} \tag{5.76}$$

with tensor product $\bigotimes$ for multiple dimensions. The message update then be-

comes, from [214, 215],

$$m_{j|\mathbf{Y}}\left(\Theta_{S,j}\right) = \langle \bigotimes\nolimits_{i=\{u,k\}} m_{i|\mathbf{Y}},\, \mathbb{E}_{\Theta_t|\theta_j}\left[\bigotimes\nolimits_{i=\{u,k\}} \phi\left(\Theta_t\right)\right] \rangle_{\mathcal{F}}. \tag{5.77}$$

Notation is again reduced for multi-message expectation,

$$\mu_{\otimes|\theta_j} = \mathbb{E}_{\Theta_t|\theta_j}\left[\bigotimes\nolimits_{i=\{u,k\}} \phi\left(\Theta_t\right)\right] \tag{5.78}$$

$$= \mathcal{C}_{\Theta_t^{\otimes}|\theta_j} \phi\left(\theta_j\right). \tag{5.79}$$

The final computation is performed through multilinear matrix operations. Please see references in this section for more details.

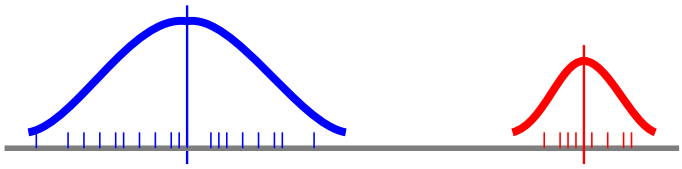
5.5.5 Hybrid Mixture Models (Dirichlet allocation)

Previous sections consider approximate convolutions and products of infinite functions for highly flexible intraclique computations and truly *sum-product* style inference. Section 5.4.6 shows how various potentials and belief messages constrain a solution into a concentrated region of the state space. Here then follows an alternative computational principle, $4b$ in Table 5.1, for approximating and multiplying belief estimates by using mixture models of parametric functions. By mixture model we mean the sum combination m of various parametric models p:

$$\left[\hat{\Theta} \,|\, \cdot\right] \approx \sum_{i=1}^{m} \alpha_i \, p_i\left(\Theta\right). \tag{5.80}$$

Fig. 5-22 illustrates two clusters of sample points which are assumed to be a combination of two normal density distributions. Assuming for the moment clustering of sample points to parametric models is handled correctly, we can build the inference process using the parametric models.

Clustering and assignment of individual samples is a large area of study. Methods such expectation maximization [156] require the number of clusters to be known, and latent Dirichlet allocation [23] for unknown cluster concentrations are corner stone examples. Discussing clustering, assignment and approximation tolerance are beyond the scope of this thesis, but is part of our proposed future work. For example the consensus example in Fig. 5-20 shows possible dimensionality reduction when separated multi-modal normal components can be reduced by pruning or culling low likelihood modes, given some maximum tolerance approx-



$$[\hat{\Theta}|\,\cdot] \approx \alpha_1 \mathcal{N}(\Theta; \mu_1, \Sigma_1) + \alpha_2 \mathcal{N}(\Theta; \mu_2, \Sigma_2)$$

Figure 5-22: Illustration of clustering and fitting low dimension parametric models to nonparametric estimate sample points, in this case two normal densities are assumed.

imation error ϵ.

We limit our discussion to conjugate normal density distributions, although the principle extends to all conjugate products and does not preclude resorting to the nonparametric inference method already discussed. Note that the hybrid methods can offer various optimizations at some permanent approximation error. The discussion in the previous sections all retain asymptotic exactness when computing the Chapman-Kolmogorov integral. The mechanics of conjugate normal densities are discussed in Section 5.5.6 hereafter.

To take the product between nonparametric functions and Gaussian mixture models, we can almost directly use the nonparametric belief propagation method with only slight modifications, see Section 5.5.1. Consider iterating through labels according to eq. (5.64), where a lower dimension Gaussian mixture model can be directly substituted for a kernel density estimate, although the output will again be a nonparametric kernel density estimate of higher dimension.

More complicated mixture densities could likely be incorporated through importance weighting, however, we prefer using the discussed function approximation methods with robust convolution and the product of infinite function approximation instead.

5.5.6 Special Case of Conjugates

At the start of this chapter, in Section 5.2.1, we mentioned state-of-the-art Gaussian methods. We can now come back full circle, considering each marginal belief as potentially being either approximated by a higher dimension nonparametric function or a lower dimension mixture model. The product of multiple normal distributions can be replaced with a single normal parametric model, according Gaussian product in eq. (5.61). In the upcoming sections, we will briefly indi-

cate two major methods to perform the intraclique operations with normal density parametric models, namely the triples method and gradient method.

The conjugate nature is also significant for the required convolution operations between normal density functions representing conditional likelihoods. Convolutions for parametric models can be computed using the sigma-point transform (unscented) [109].

In the extreme case we reduce the approximation down to a single normal density distribution, resulting in the well known nonlinear least squares. The Gaussian product and convolutions operations associated with the Chapman-Kolmogorov transit integral become equivalent to the Gram-Schmidt orthonormalization process.

Furthermore, we again emphasize the issue of consistency, which is no different from the requirements for nonparametric inference: A valid posterior distribution must encapsulate all measurement and computation errors. We assume that the user has assigned sufficient uncertainty (covariance) in each of the measurement likelihood potentials, encapsulating errors which may have been introduced into the joint probability distribution.

The triples method

By rewriting purely Gaussian potentials with linearized measurement residuals $\delta(\Theta; \mathbf{z}) = \mu(\mathbf{z}) - \Theta$, Cowell [37] presents the triples method $G = (g, \mathbf{h}, \mathbf{K})$,

$$\mathcal{N}(\Theta; \mu, \Sigma) = \exp\left(-\frac{1}{2}\|\delta(\Theta; \mathbf{z})\|_{\Sigma}^{2}\right) = \exp\left(g + \Theta^{T}\mathbf{h} + \Theta^{T}\mathbf{K}\Theta\right), \tag{5.81}$$

for multiplying or dividing beliefs

$$\begin{aligned} \mathcal{N}_1(\cdot) \times \mathcal{N}_2(\cdot) &= G_1 \times G_2 = (g_1 + g_2, \mathbf{h}_1 + \mathbf{h}_2, \mathbf{K}_1 + \mathbf{K}_2) \\ \mathcal{N}_1(\cdot) / \mathcal{N}_2(\cdot) &= G_1 / G_2 = (g_1 - g_2, \mathbf{h}_1 - \mathbf{h}_2, \mathbf{K}_1 - \mathbf{K}_2). \end{aligned} \tag{5.82}$$

We could likely define the triples method on manifolds, such as the Lie algebra, but this is currently beyond the scope of this thesis.

The Schur complement again defines the marginalization operation. Consider an organized block structure

$$\mathcal{N}(\Theta_1, \Theta_2) = \exp\left(g + \begin{bmatrix}\Theta_1 \\ \Theta_2\end{bmatrix}^{T} \begin{bmatrix}h_1 \\ h_2\end{bmatrix} + \begin{bmatrix}\Theta_1 \\ \Theta_2\end{bmatrix}^{T} \begin{bmatrix}K_{1,1} & K_{1,2} \\ K_{2,1} & K_{2,2}\end{bmatrix} \begin{bmatrix}\Theta_1 \\ \Theta_2\end{bmatrix}\right) \tag{5.83}$$

and integrating out variables $\int_{\mathbb{R}} \mathcal{N}(\Theta_1, \Theta_2)\ d\theta_1$,

$$h = h_2 - K_{2,1}K_{1,1}^{-1}h_1, K \qquad = K_{2,2} - K_{2,1}K_{1,1}^{-1}K_{1,2}. \tag{5.84}$$

Therefore eqs. (5.14) and (5.15) can be computed with the triples method for algebraic products and the sigma-point transform for convolutions of conjugate Gaussian functions.

Gradient methods (Gram-Schmidt/Elimination methods)

Gaussian parametric functions are defined by their first and second statistical moments, mean and covariance, which may be directly computed. The mean can be computed under a localized *max-product* assumption, see Section 5.2.1. The Laplace approximation can be used to recover a covariance estimate [112].

The iSAM2 algorithm [115], which inspired our approach, is a highly optimized version of the single component Gaussian marginal belief mixture (unimodal). The iSAM2 algorithm assumes all measurement likelihood residual functions δ_i are continuous and convex, whereby quasi-Newton methods are used to perform the intraclique inference process. Newton methods are popular given rapid convergence rates near the final solution.

High performance of Newton methods originates from being able to modify the parametric state estimate independently along each degree of freedom, [198]. If we consider the clique association matrix again, for example Table 5.2, we see columns as degrees of freedom and rows as interactions between them. The ability to isolate each of the dimensions us usually implicitly achieved through an ortho-normalization process. Cholesky [116] or QR-decomposition are the current mainstream techniques, enabled by Givens rotations [73] or the Householder transformation.

5.6 Algorithm Summary: Intraclique Operations

We present a summary of the operations discussed in this chapter. Given a particular factor graph constructed at time t, we perform incremental updates to the Bayes tree structure (as shown in Fig. 5-9) using an elimination procedure described in Algorithms 2 and 3. Unclaimed likelihood functions are then associated with cliques in the tree, according to frontal variables and by starting consumption at the leaves of the tree. Each clique will have some subset of factors related to the frontal variables and loosely related to the separator variables of that clique.

Interclique message passing occurs in accordance with illustrations in Fig. 5-6, first from leaves to root for the most recent variable marginal posteriors and then back down to leaves to recover all complete marginals. Recall that the factor graph and associated measurement likelihood functions are only assembled at run-time. The computer software algorithms are written **before** the user defines a likelihood model, therefore preventing any knowledge about what the posterior distribution might be.

Interclique message passing hinges on a general procedure for calculating the marginal beliefs over clique variables, referred to as the intraclique process. The steps in Algorithm 4 summarize our solution for the intraclique operations, which computes the Chapman-Kolmogorov transit integral. Please see Appendix C for a tutorial type example of the intraclique operations.

Algorithm 4: Intraclique operations for solving Chapman-Kolmogorov during upward message passing for clique j with marginal partial posterior $M_{j|\mathbf{Y}}$, as shown in eq. (5.14). Several steps can be precomputed at construction.

Non-leaf cliques receive incoming messages of approximated belief functions from child cliques $\bigcup_u \hat{m}_{u|\mathbf{Y}}\left(\Theta_{S,u}\right)$ according to separator variables in cliques $\{u\}$;
Determine all local potentials not claimed by frontal variables lower down in the tree;
Form the clique association matrix, as illustrated in Table 5.2;
foreach *(concurrently)* $\Theta_{j,k}$ *columns of only singletons* **do**
    Compute independent product marginals, as illustrated with eq. (5.47):
    if *all singleton beliefs are conjugate parametric* **then**
        Compute product with analytic conjugate computations, as discussed in Section 5.5.6
    else
        Compute nonparametric product of infinite functionals using any approximation technique from Section 5.5;
List $\{\Theta_{j,g}\}_{Gibbs} \subseteq \{\Theta_{j,k}\}_{k=1}^{|C_j|}$ of columns with pairwise potentials for block Gibbs scheme, eq. (5.46);
while *not stationary* $d\left(\hat{M}^{(i+1)}, \hat{M}^{(i)}\right) \prec \varepsilon$, *or user iteration limit* **do**
    # these steps describe Markov chain, i.e. $\hat{M}_{j|\mathbf{Y}}{}^{(i+1)} = T_{\mathcal{Q}}\, \hat{M}_{j|\mathbf{Y}}{}^{(i)}$.
    foreach *(sequentially)* $\Theta_{j,g}$ *in randomized or linear sweep* **do**
        foreach *(concurrently) associated pairwise likelihood model* $\varphi\left(\Theta_{j,g}, \cdot\right)$ **do**
            Approximate convolution with user likelihood model to $\left[\hat{\Theta}_{j,g}{}^{(i')} \mid \cdot\right]$, as discussed in Section 5.4.4;
        Take product of individual proposals $\left[\hat{\Theta}_{j,g}{}^{(i')} \mid \cdot\right]$, eqs. (5.48) & (5.49), according methods in Section 5.5 (nonparametric), or analytically for parametric conjugate beliefs, as illustrated in Fig. 5-20;
        Assemble new marginal function estimate $\left[\hat{\Theta}_{j,g}{}^{(i+1)} \mid \hat{\Theta}_{j\backslash g}{}^{(i')}, \mathbf{Y}_j\right]$ as required, eq. (5.49), using in Section 5.4.1;
New upward message $\hat{m}_{j|\mathbf{Y}}\left(\Theta_{S,j}\right)$ available from latest variable belief estimates, (marginalize out frontals $\Theta_{F,j}$, eq. (5.15));
Algorithm performance determined by Bellman optimality, Section 5.4.2.

5.7 Critical Analysis

In this chapter we have explored several details of the *sum-product* inference algorithm. We have argued, without hard proofs, that reversibility in the nested Gibbs type operations results in stationary, or functional fixed point, marginal density estimates of all variables in the system.

We have taken the approach of converting a cyclic factor into an acyclic Bayes tree representation and showed several optimizations specific to our robot navigation use case, namely incremental updates and specialized variable orderings. Our emphasis on the Bayes tree follows from successes had in [115,125,218], while providing, to the best of our knowledge, the first nonparametric *sum-product* inference algorith to operate on thousands of varialbes simultaneously. Previous nonparametric methods have operated at much lower dimension, or resorted to randomized ordering in loopy belief propagation. Previous tree based techniques [127] suffered excessively from large clique dimensions, but we are able mitigate this problem due to the CCOLAMD [39] variable ordering technique.

Refactoring of the full joint density is absolutely fundamental in finding a computationally tractable and robust SLAM solution. The heuristics used to construct the Bayes tree are special in that they minimize the modifications to the tree when small modifications are made to the underlying factor graph. Modifications to the factor graph are common in SLAM but have not been yet been considered in other main stream Bayesian inference literature. We maintain that the Bayes tree representation is a key computational aspect going forward.

In Section 5.2.2 we introduced the *range space of possible multi-modal solutions*, and discussed at length how the *sum-product* approach allows us to search to the full solution space. We are not disputing the *range space of possible multi-modal solutions* as very large or intricate, but rather that we have a practical solution beyond unimodal parametric inference which asymptotically approximates the *sum-product* solution. In Chapter 6 we will show how the algorithm presented here does in fact find valid multi-modal solutions.

One of the major concerning factors is computational cost and how the *curse of dimensionality* factors into our approach. We note that an exact nonparametric solution to the naive full state factor graph definition is indeed exponentially complex. Instead our premise rests upon nine central computational principles to make the inference process tractable. The computational principles used are all either exact or asymptotically correct, and are listed in Table 5.1. Furthermore, asymptotic principles in the algorithm are defined relative to some quality parameter $\mathcal{Q}$ which allows us to improve granularity of the solution for increased computational cost.

The Bayes tree is a symbolic factoring of the conditional independence structure of the full joint probability distribution. The Bayes tree algorithm minimizes the dimensionality of cliques through a provably good variable orderings [39]. We only ever need consider one clique at a time, and therefore only need to deal with the exponential complexity of the dimension of that clique. For increased speed, we exploit further nested simplifications within a clique, similar to [125], while simultaneously leveraging improved parallel computation.

Consensus (information loss I) plays a important part in drastically reducing the computational complexity. As discussed in Section 5.4.6 we expect fewer real world events likely, as more information is gathered from one reality. Algebraically this concentration of belief in the state space reduces the potential exponential number of modes, since most product operation culls most of the very low probability modes. By ignoring low likelihood modes and focusing on most likely modes, we argue it possible to arrive at feasible solutions which are able to track the few remaining consensus modes accurately.

Belief approximation tolerance (information loss II); We have chosen to approximate the marginal posterior densities of each variable in the system by approximation. By reducing an infinite belief function to a limiting accuracy approximating function, we can concentrate computation around the dominant features and modes in the system. As a give and take argument, the approximation introduces a loss of information, but maintains some flexibility in recovering modes if more information becomes available. Our intention is to use the approximation process to implicitly achieve consensus discussed above.

Iterative approximation (imputation); We have also chosen to focus on belief space computations using an iterative approximation scheme (nested, block, batch Gibbs sampling), rather than naively searching the entire state space. While our approach must start from initial condition, which introduces the risk of a local only search, we do propose all modes and uncertainties of each likelihood factor in the factor graph model.

Approximate convolution (information loss III). In Section 5.4.4 we discuss one mechanism by which conditional distributions are enforced in the factor graph model. This can be seen as approximating the convolution operation in a Bayesian approach. We selectively solve only sample test points in the belief, after which kernel density approximations are repeated. This limited testing and sampling across each of the conditionals is a further loss of information during inference, and can be thought of as the digital equivalent of a continuous analog belief.

Approximation of the product of infinite function; By approximating continuous functions with finite dimensions (nonparametrics), such as kernel density esti-

mates, we are able to compte approximate products of the underlying infinite belief functions. For example, the nonparametric belief propagation technique (Section 5.5.1) again leverages ϵ tolerance in function approximation and an internal multi-scale successive Gibbs approximation scheme. Approximating the product, as illustrated in Fig. 5-20, drastically reduces the associated computational cost further, see [218].

In Section 5.4.4 we discussed how the convolution with likelihoods creates new modes, and the product operation, in Section 5.5, culls the low likelihood modes. This creation and culling of modes approach reintroduces the prospect of full global solutions. In theory, by turning the quality parameter $\mathcal{Q}$ to maximum should allow near global solutions.

5.8 Conclusion

In conclusion, our approach allows recovery of nonparametric, multi-modal posteriors from cyclic large factor graph joint probability descriptions. This chapter and the examples that follow in Chapters 6 and 8 shows the existence of a multi-modal inference solution. Further research into convolution approximation could be better leveraging conditional kernel Hilbert embeddings, see [214]. Reproducing kernel Hilbert spaces may well play an important role in computing product of infinite function approximations too.

The key theoretical challenge is understanding that, for belief space methods, that the inference process can be considered an operator $T_{\mathcal{Q}}$ with proof that it will contract to a stochastic fixed point. The further theoretical requirement is to know that the contracted fixed point of $T_{\mathcal{Q}}$ is in fact near (or nearest) to the true functional embedding point of the true joint posterior belief. We are not directly able to prove that statement in this chapter, because the topological structure of the problem and likelihood models used is entirely unknown at this stage.

We do restrict the class of user-defined likelihood models to: (i.) marginalized all discrete varialbes into a continuous belief at function product time; and (ii.) convolutions with user-defined functions should only depend on the current state, and not depend on other outside values (Markov property).

In places, the Markov property is not kept since some user-defined likelihood functions have lower dimension that the variables they are constraining. We are able to break the correlation between consecutive values by simply adding more clique level Gibbs iterations and thinning out the samples. A further approach is to compose several factors together into a single higher dimensional factor. We do

claim that there a stationary distribution from MCMC standpoint which has an equivalent feature space embedding; and that by satisfying the Markov propery we can use composite detailed balance to ensure the solution will converge towards some fixed point.

A further avenue still could be to improve the intraclique operations by leveraging parametric mixture models. By accepting a permanent ϵ true belief approximation error, in exchange for higher speed parametric computations. We stress again, the existing iSAM2 algorithm [115] is a single Gaussian mixture, where a *max-product* is computed. The author feels there is a lot of room for development between our presented approach and the existing iSAM2 algorithm. Lastly, we note that the iSAM2 algorithm (*max-product*) is a special case of the broader *sum-product* belief propagation approach which is presented here.

Further still, in Section 5.4.2 we introduced Bellman optimality in context of multi-modal inference. We proposed a cost function and discussed how the optimality is used to influence our algorithm design decisions. We also showed how momentum can be used in future algorithm developments, to arrive at posterior estimates more directly.

Chapter 6

Canonical Examples

Chapter 5 discusses the *Multi-modal iSAM* algorithm[1] for computing the associated *sum-product* inference solution. This chapter is dedicated to canonical examples to help understand the nature of multi-modal solutions. Chapter 8 discusses the Julia[2] implementation used, along with results from real robot experiments.

6.1 Introduction

The *Multi-modal iSAM* (Smoothing and Mapping) algorithm implements a non-parametric *sum-product* solution to factor graph described problems. These factor graph descriptions do not necessarily have to be associated with robotic navigation, although that is our problem focus area. In this chapter, we look at a few generic examples to better understand the nature of multi-modal beliefs, and what the inputs and outputs of the algorithm are.

The first section explores a multi-modal square root solution, as the most basic one dimensional example to illustrate how the iterative procedure T_Q operates. Next we look at a one dimensional robot three door and four door navigation example originally inspired by [224] but reworked here for general continuous domain simultaneous localization and mapping (SLAM). . Later in the chapter we thoroughly investigate the SLAM-e-donut two dimensional range only navigation example.

[1] Implementation of *Multi-modal iSAM* and canonical examples is available at www.github.com/dehann/IncrementalInference.jl

[2] Julia programming language, "A fresh approach to numerical computing" [19]: www.julialang.org and www.juliacomputing.com

6.2 Multi-modal Square Root

Inspired by the Babylonian method to iteratively compute the square root $y = x^2$, we construct an example system such that

$$\begin{aligned} 0 &= \delta\left(y = (x^2 + \eta), \tilde{x}\right) = y - \tilde{x} \times \tilde{x} \\ z &\sim [\, Z \,|\, Y, X \,] = \mathcal{N}\left(\mu = \delta, \sigma\right), \end{aligned} \tag{6.1}$$

where η is a noise slack variable producing noisy square root $\tilde{x}$ from zeroing a residual δ, see eq. (5.54). However, our generalized iteration scheme is already built into the inference algorithm, given a continuous measurement likelihood. Consider the measurement z from the conditional measurement likelihood distribution $[\, Z \,|\, Y, X \,]$ between variable X^2 and its square root X, as modeled in a factor graph shown in Fig. 6-1. A prior belief is added to the square terms to gauge the solution $[\, X^2 \,]$

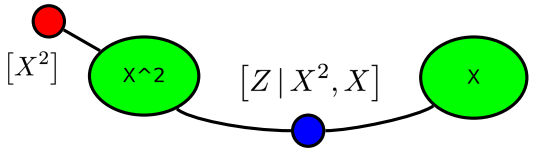


Figure 6-1: Example factor graph for square fixed point example, with two variables X and X^2. Two factors are present, namely a prior on $[\, X^2 \,]$ and a pairwise squaring conditional likelihood $[\, Z \,|\, X^2, X \,]$.

The factor graph in Fig. 6-1 is resolved into a Bayes tree on just one root clique

$$M_{1|\mathbf{Z}} = \left[\, X^2, X \,|\, Z \,\right] \propto \left[\, Z \,|\, X^2, X \,\right] \left[\, X^2 \,\right], \tag{6.2}$$

where $M_{1|Z}$ represents the marginal belief of the root clique. The root clique association matrix is:

	X^2	X
$[\, Z \,\|\, X^2, X \,]$	x	x
$[\, X^2 \,]$	x	

As defined by the normally distributed $\mathcal{N}$ prior and measurement likelihood functions

$$\begin{aligned} \left[\, X^2 \,\right] &= \mathcal{N}\left(\mu = 4, \sigma = 0.05\right) \\ \left[\, Z \,|\, X^2 \to a, X \to b \,\right] &= \mathcal{N}\left(\mu = a - b \times b, \sigma = 0.01\right). \end{aligned} \tag{6.3}$$

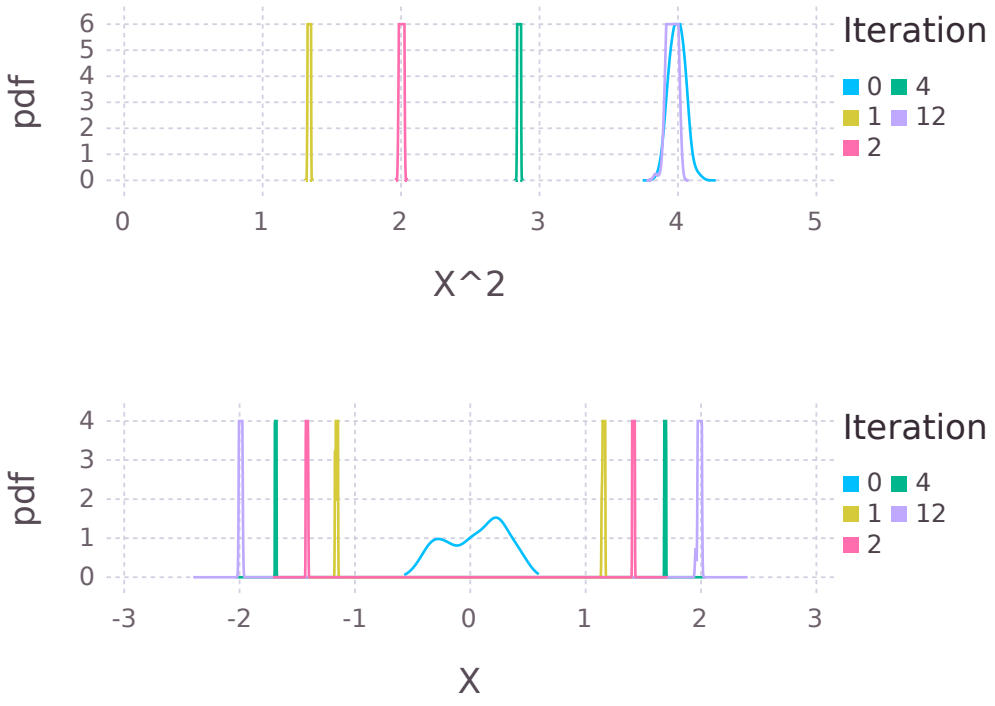


Figure 6-2: Example of square root marginal beliefs converging to the stationary distribution for factor graph shown Fig. 6-1.

We understand the intraclique successive iteration belief approximation process — Chapter 5, Section 5.4 — will iteratively estimate

$$\begin{aligned}&\left[X^2 \,|\, X, Z\right] \propto \left[X^2 \,|\, X, Z\right] \left[X^2\right] \\ &\left[X \,|\, X^2, Z\right],\end{aligned} \tag{6.4}$$

using convolution approximation eq. (5.54). Fig. 6-2 shows the marginal posterior belief estimates for both $[X^2 \,|\, \cdot], [X \,|\, \cdot]$ during the iteration cycles in the top and bottom plots respectively.

The top plot in Fig. 6-2 shows $[X^2 \,|\, \cdot]$ approximated over several iterations. We see the solution is initialized at user prior 4, but at the first iteration is pulled away to around 1.3. During the next 10 iterations we see the belief return to the prior be-

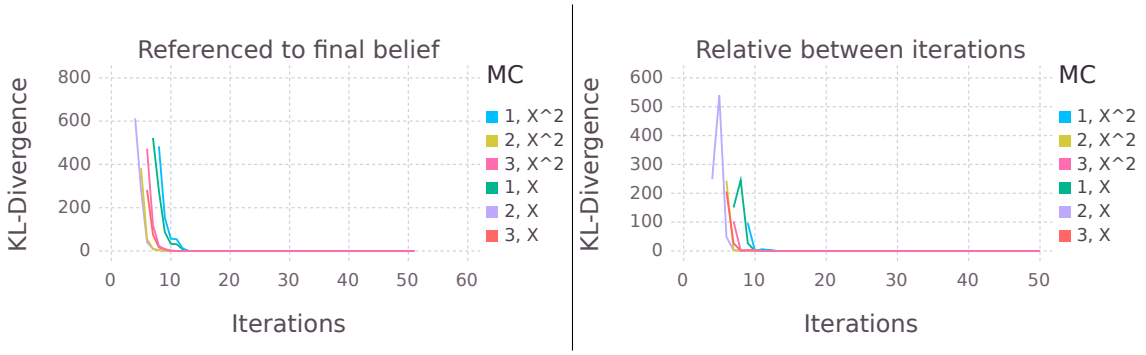


Figure 6-3: Square root fixed point KL-Divergences for factor graph shown in Fig. 6-1. *Left*, shows reference KL-Divergences against final state after 50 iterations. *Right* shows KL-Divergence between successive Gibbs substitution sampling iterations. Multiple curves indicate repeated runs to see the trend pattern.

lief position around 4. We point out that the belief moves around with little overlap between previous and next beliefs, a fact that would make importance weight type belief product approaches difficult. Instead, the nested Gibbs approach is able to maintain healthy belief updates in "new" regions of the state space. We note that repeated runs have varying convergence time. The example chosen here shows uncharacteristically slow convergence.

The bottom plot in Fig. 6-2 shows $[\,X\,|\,\cdot\,]$ approximated over the square root estimate. In particular, we note the initial condition on X, uniform $\left[\,X^{(0)}\,|\,\cdot\,\right] = U\left(\frac{-1}{2}, \frac{1}{2}\right)$, shown as iteration 0. After the first iteration we find the bimodal belief heading towards ± 2, which is indeed the square root of 4. This discrepancy between the initial guess and the true marginal belief shows $[\,X\,|\,\cdot\,]$ resulted in the initial "error" at marginal $[\,X^2\,|\,\cdot\,]$ during the first couple of iterations. Once the system approached the stationary point, we find an accurate estimate of the true solution.

Fig. 6-3 shows the KL-divergence, eq. (5.20), for the variable marginals against successive approximation iterations and three Monte Carlo runs. The left plot shows the divergence of each marginal estimate against the last belief, while the plot on the right shows the intermediate change in divergence between iterations. We see that after 12 or 13 iterations the belief function approximation has settled to nearby the stationary value.

As discussed in Section 5.4.1, we can look at the mean map projections of the beliefs for better understanding of the inference algorithm. Using the explicit quadrature frequency basis, we find the marginal belief progression shown in

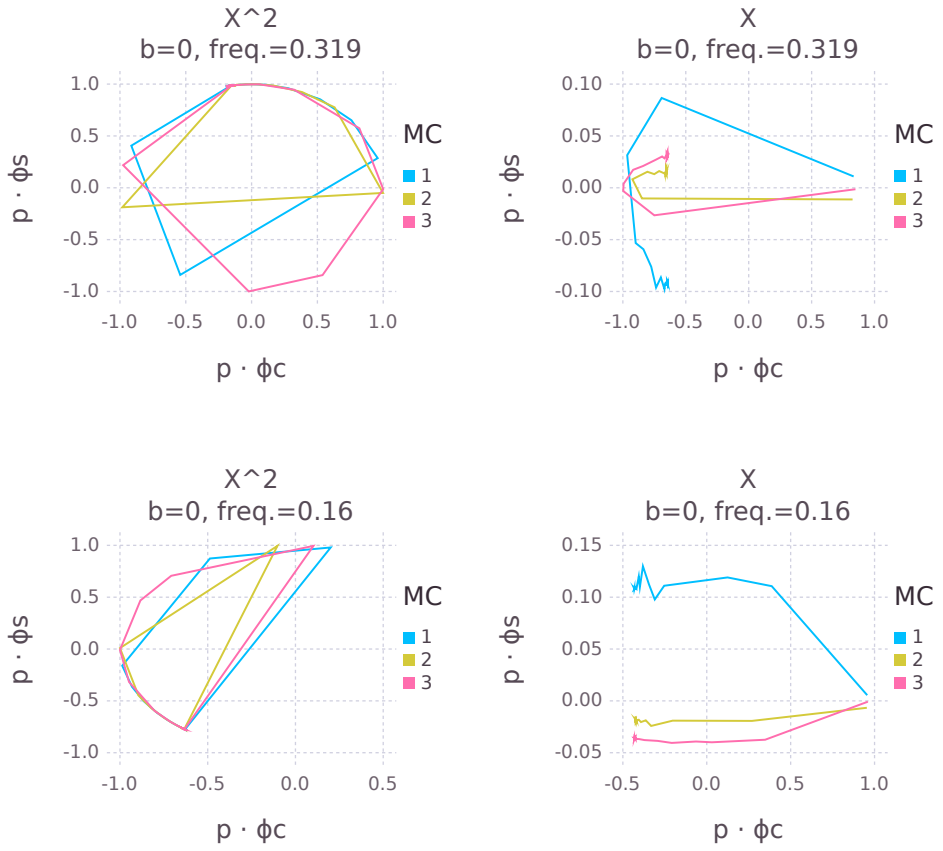


Figure 6-4: Trajectory of higher (*top*) and mid (*bottom*) frequency Fourier series coefficients from successive approximation inference algorithm over a factor graph shown in Fig. 6-1. *Left* illustrates the point projection of the X^2 marginal belief. *Right* illustrates the approximated belief trajectory for the X variable belief.

Fig. 6-4. The sinusoid projection is shown on the vertical and cosine projection on the horizontal axes for a mid (0.16) and higher (0.319) frequencies. Traces for three Monto Carlo runs are shown. We note these points corresponds to each of 50 successive approximation iterations, showing how the solution moves and converges to a region of the mean map projection space.

The right side plots in Fig. 6-4 shows how the inference algorithm moves the belief from one point in steps towards the left by a fair distance for higher (top) and mid (bottom) frequencies. In contrast, for lower frequency, the right plot in Fig. 6-5

also shows how the marginal belief $[X \mid \cdot]$ is moved, but not by the same distance – that is less information is updated in lower frequencies. The right upper plot in Fig. 6-4 starts to show periodicity (aliasing) as the three trace solutions move from right to left and then curve back around. Higher frequencies are more exaggerated.

In all higher, mid, and lower frequency cases (for the right side plots) of Fig. 6-4 and Fig. 6-5, we notice that the Monte Carlo solutions start out close together but converge to different points in the projected feature space. This separation is not visible in Fig. 6-3 and shows the value of studying the mean map projections.

The most separation is over the sinusoid basis (vertical) component, which indicates symmetry about the origin and translates into imbalance in the amount of mass between the $x = +2$ and $x = -2$ modes. We ascribe these vertical variations to the difference between the true and estimated beliefs Δ, as shown in Fig. 5-16, which lie inside the inference operator error term ε.

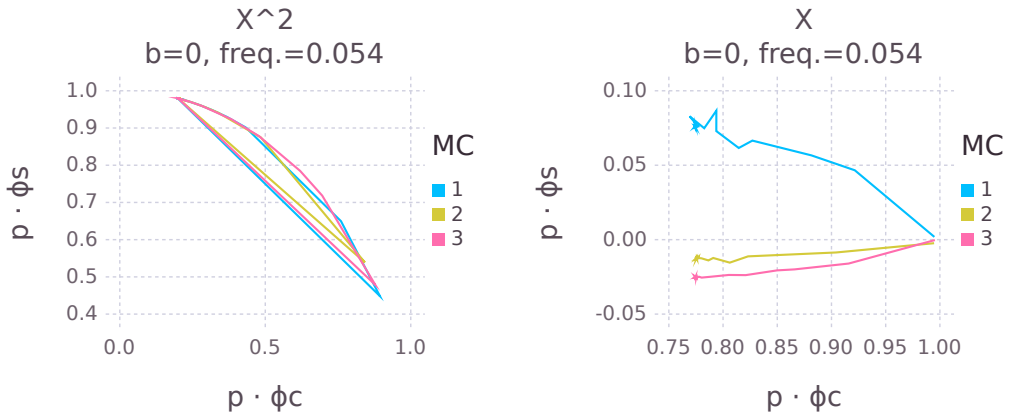


Figure 6-5: Trajectory of low frequency Fourier series coefficient for Gibbs successive substitution inference over factor graph shown in Fig. 6-1.

Still with the plots on the right side in Fig. 6-4 and Fig. 6-5, we notice that the traces quickly move from similar initialization on the right to concentrated regions on the left. The convergence regions to the left are the mean map fixed points, as discussed in Section 5.4.3. Variations around the fixed points are ascribed to approximation accuracy ϵ, with the inference operation T_Q, as shown in Fig. 5-16. In this case we immediately note that ϵ related variations are much smaller than ε related errors.

Plots on the left of Fig. 6-4 show the higher (top) and mid (bottom) frequencies

for $[\, X^2 \,|\, \cdot \,]$, which start and end at the same point. This corresponds to the belief moving away and back from the desired value of 4, shown in the top plot of Fig. 6-2.

Next, we investigate a one dimensional three mode example more closely associate with robotic navigation.

6.3 Multi-modality: 3 Doors Example

The three doors robot localization has become a classic Bayesian example in the SLAM community [224]. A robot knows the location of three doors and occasionally makes a door sighting while driving around in one dimension.

The robot starts with seeing one door and then drives through an intermediate pose location, making a second door sighting at the third pose. The robot doesn't know which of the three known doors it has seen, but does have a noisy measurement of the distance traveled between the doors. Our task is to correlate the distance traveled between sightings to the known map of door locations.

6.3.1 Factor Graph with Multi-Modal Potentials

We reformulate the three doors example to explain the tree based propagation algorithm. Rather than apply a histogram filter [224] to estimate the posterior belief, we construct the nonparametric factor graph shown in Fig. 6-6. Green nodes indicate three pose variable nodes and blue nodes are odometry measurement likelihood factors.

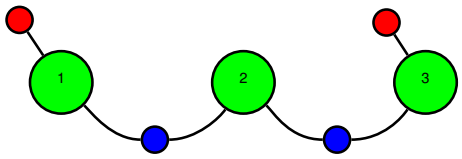


Figure 6-6: Factor graph of the "three doors" example, with three robot poses (green nodes 1, 2 & 3). Doors are seen by the robot as it drives past some of them. Blue nodes indicate unimodal constraint functions. Poses are connected through odometry constraints. Red nodes are multimodal constraint functions. The robot has seen a door at the first and last pose, but unsure which of three doors were seen.

Fig. 6-7 shows the three possible (multi-modal) door sighting locations, which

are represented as:

$$[\, Z_1 \,|\, X_1 \,] \propto \psi_{X_1}\left(x_{1,abc}\right) \tag{6.5}$$
$$[\, Z_4 \,|\, X_4 \,] \propto \psi_{X_3}\left(x_{3,abc}\right). \tag{6.6}$$

We observe that each door sighting hypothesis can be represented as a Gaussian term. The accumulated sum of Gaussians represents the measurement likelihood:

$$\psi_{X_i}\left(x_{i,abc}\right) = \exp\left(-\frac{\left(h\left(x_i\right) - z_{i,a}\right)^2}{2{\sigma_j}^2}\right) + \exp\left(-\frac{\left(h\left(x_i\right) - z_{i,b}\right)^2}{2{\sigma_j}^2}\right) + \exp\left(-\frac{\left(h\left(x_i\right) - z_{i,c}\right)^2}{2{\sigma_j}^2}\right) \tag{6.7}$$

where i represents the variable node number and j the measurement number.

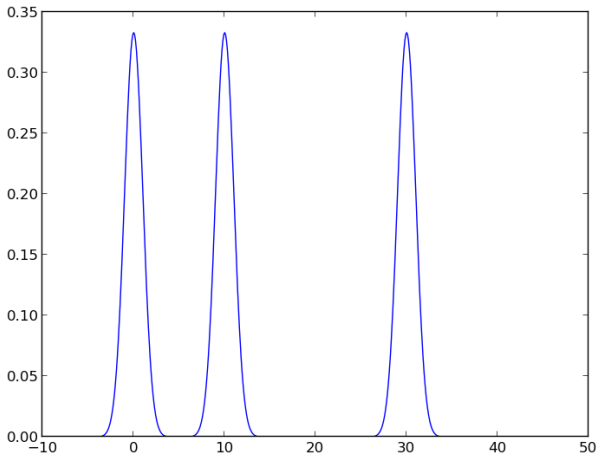


Figure 6-7: Multi-modal prior measurement probability based on a door sighting.

The unimodal odometry densities are $\left(h_{odo}\left(x_{i-1}, z_i\right) - x_i\right) \sim \mathcal{N}\left(\mathbf{0}, \sigma_i\right)$, and

$$[\, Z_2 \,|\, X_1, X_2 \,] = \varphi_{1,2} \propto \exp\left(-\frac{\left(h\left(x_1, z_2\right) - x_2\right)^2}{2{\sigma_2}^2}\right)$$
$$[\, Z_3 \,|\, X_2, X_3 \,] = \varphi_{2,3} \propto \exp\left(-\frac{\left(h\left(x_2, z_3\right) - x_3\right)^2}{2{\sigma_3}^2}\right). \tag{6.8}$$

With the measurement likelihood functions in hand, we develop the Bayes tree and analytically compute the posterior solution.

6.3.2 Bayes Tree and Analytical Beliefs

We choose to eliminate variables in the factor graph, Fig. 6-6, with variable ordering X_1, X_2, X_3. The elimination scheme arrives at a Bayes tree presented in Fig. 6-8. In this example there is only root ($j = 1$) and child ($j = 2$) cliques, where j indicates the clique number. The child is also a leaf clique in this tree.

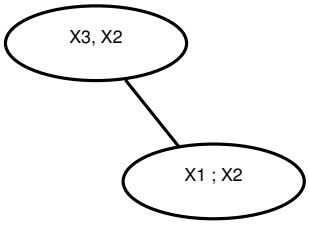


Figure 6-8: Bayes tree of "three doors" example using variable ordering X_1, X_2, X_3.

The tree shows a factorization of the root clique and one child clique. We interpret the full joint posterior of the system by product of Bayes tree cliques:

$$[\, X_1, X_2, X_3 \,|\, Z \,] \propto [\, X_3, X_2 \,|\, Z \,]\, [\, X_1 \,|\, X_2, Z \,] \tag{6.9}$$

Inference is performed through upward message passing, from leaf to root cliques. The upward message from the leaf clique is computed from the clique marginal belief $M_{2|Z}$.

Leaf Clique Marginal

To compute the marginal posterior of the leaf clique as potentials containing the frontal variable x_1, we use the associated odometry measurement and door sighting measurement likelihoods

$$\begin{aligned} M_{2|Z}\,(x_1, x_2) &\propto [\, Z \,|\, X_1, X_2 \,]\, [\, X_1 \,] \\ M_{2|Z}\,(x_1, x_2) &\propto \varphi_{1,2}\,(x_1, x_2)\, \psi_1\,(x_{1,abc})\,. \end{aligned} \tag{6.10}$$

The message density over the separator $\Theta_{S,2} = \{X_2\}$ is $m_{2|Z}\,(x_2)$ is calculated from marginalizing out x_1

$$m_{2|Z}\,(x_2) = \int_R M_{2|Z}\, dx_1, \tag{6.11}$$

and is sent from leaf to parent. Using *Mathematica*, we perform the symbolic computations to find the actual message belief

$$m_{2|Z}(x_2) = e^{-\frac{1225}{4}-\frac{1}{4}(-10+x_2)x_2}\left(e^{300}+e^{15x_2}+e^{5(50+x2)}\right)\sqrt{\pi}, \tag{6.12}$$

which is shown in Fig. 6-9.

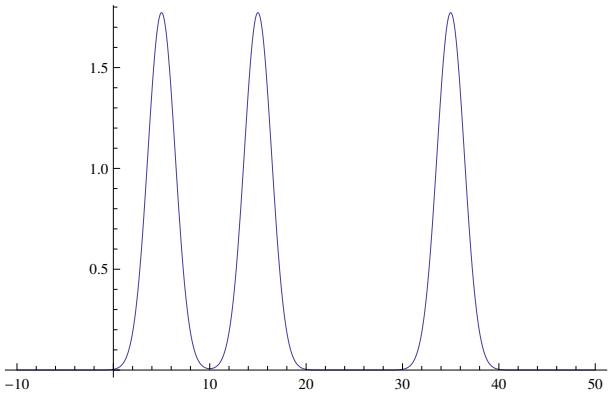


Figure 6-9: Message from leaf to root node, m_2

Root Clique Marginal

After the upward message pass from the leaf, we can calculate the marginal full posterior $M_{1|\mathbf{Y}}$ at the root, where $\mathbf{Y}$ represents all measurements from that clique down to the leaf. We take the product of the incoming message and the local potentials associated with $\Theta_{F,1} = \{X_3, X_2\}$:

$$\begin{aligned} M_{1|\mathbf{Y}}(x_2, x_3) &\propto [\, Z \,|\, X_2, X_3 \,]\, [\, X_3 \,]\; m_{2|Z}(X_2) \\ &\propto \psi_{2,3}(x_2, x_3)\, \varphi_3(x_{3,abc})\, m_{2|Z} \end{aligned} \tag{6.13}$$

which gives the marginal density over the root clique. The root clique marginal is special as it represents the a marginal from the full joint posterior distribution $M_{1|\mathbf{Y}}(x_2, x_3) = \int_{\mathbb{R}} [\, X_1, X_2, X_3 \,|\, Z \,]\; dx_1$.

The second odometry measurement likelihood $\psi_{2,3}(x_2, x_3)$ is also unimodal,

with $\sigma_3 = 1$, $z_3 = 5$:

$$\varphi_{2,3}(x_2, x_3) = \exp\left(-\frac{(h_3(x_2, x_3) - z_3)^2}{2{\sigma_3}^2}\right)$$
$$= \exp\left(-\frac{(x_3 - x_2 - 5)^2}{2}\right) \tag{6.14}$$

At pose location x_3 another door is observed and is again introduced as a three mode unary potential, eq. (6.7). Again, using *Mathematica*, the product of incoming messages (marginals) and local measurement potential is:

$$M_{1|Z} = \psi_{2,3}(x_2, x_3)\ \psi_3(x_3)\ m_2(x_2)$$
$$= e^{-\frac{3075}{4} - \frac{3x_2^2}{4} + x_2\left(-\frac{5}{2} + x_3\right) + 5x_3 - x_3^2}\left(e^{300} + e^{15x_2} + e^{5(50+x_2)}\right) \times$$
$$\left(e^{450} + e^{30x_3} + e^{10(40+x_3)}\right)\sqrt{\pi} \tag{6.15}$$

This is again a joint potential over x_2 and x_3. Marginalizing out x_2 will produce the unscaled density over the last pose location x_3, also shown in Fig. 6-10:

$$[X_3 \,|\, Z] \ \propto\ \frac{2\pi}{\sqrt{3}} \exp\left(-\frac{2(1075 + (-5 + x_3)x_3)}{3}\right) \times$$
$$(e^{700} + e^{40x_3} + e^{\frac{100(6+x_3)}{3}} + e^{20(20+x_3)} + e^{10(45+x_3)}$$
$$+ e^{\frac{40(45+x_3)}{3}} + e^{10(65+x_3)} + e^{650+\frac{10x_3}{3}} + e^{250+30x_3}) \tag{6.16}$$

The new posterior arrives as a single dominant mode with eight other much smaller modes, as shown in Fig. 6-10.

As information propagates up the tree, some potentials introduce more modes but these eventually coagulate with other modes from potentials higher up in the tree. If sufficient information exists, invalid hypotheses drop to low weight modes. These modes can essentially be ignored once we have strong confidence in a subset of other nodes.

6.3.3 Approximate Belief Propagation on Bayes Tree

In this section, after studying the analytic solution to the three doors example above, we consider a rudimentary Gibbs sampling approach to illustrate how the

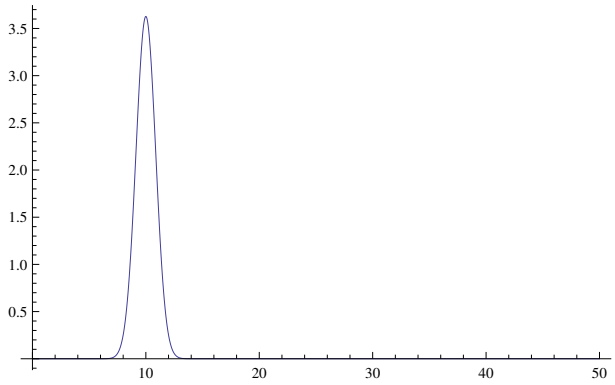


Figure 6-10: Posterior over X_3 after marginalization of potentials product at the root of the tree.

same posteriors are estimated. In the next section we will upgrade our discussion to using the *Multi-modal iSAM* solver to compute truly multi-modal posteriors of an extended four door example.

In the three door case, the door sighting measurement likelihood function is given in (6.7) and odometry likelihoods in eq. (6.8). The product of Gaussians sum results in an exponential increase of terms as all multi-modal potentials in a factor graph are multiplied together, as is shown in the previous section.

We choose to approximate the posterior densities of each clique in the Bayes tree with a set of samples. Furthermore, the number of samples will be restricted to introduce an ϵ loss of information, but simultaneously introduce a natural mechanism to cull computation of low likelihood modes.

The algorithm maintains samples representing the most dominant modes in the system at any point in time. When inference on the clique is repeated, a new set of samples is generated which may approximate different modes than before.

Repeating the three doors example computations, we first approximate the posterior of the leaf clique $M_{2|Z}$, see eq. (6.10).

Markov Chain Sampling, Leaf Clique

We can approximate the marginal density message $[\, X_2 \,|\, Z_1, Z_2 \,]$ with MCMC samples, by identifying the dependency structure in the leaf clique with the clique association matrix (see Section 5.4.3):

	X_1	X_2
$\varphi_{1,2}$	x	x
ψ_{X_1}	x	

In this case we can sample X_1 from ψ_{X_1}, eq. (6.7), and use the sample to define the conditional odometry distribution $\varphi_{1,2}$, eq. (6.8). We take the samples over X_2 as the marginal belief that represents the upward message over the leaf clique separator variable $\hat{m}_{2|Z}\,(x_2)$. A histogram plot of samples in $\hat{m}_{2|Z}$ is shown in Fig. 6-11.

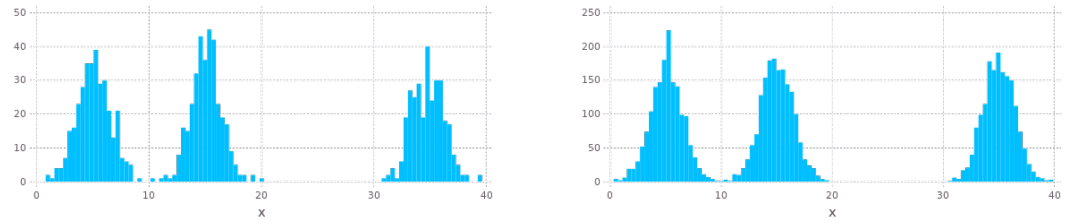


Figure 6-11: Histogram of 1000 (left) and 5000 (right) samples in message m_2, which is an approximation of the analytic version shown in Fig. 6-9

Incorporating Message $\hat{m}_{2|Z}$ at the Root Clique

Recall the root clique marginal belief $M_{1|\mathbf{Y}}$ from eq. (6.13), where $\hat{m}_{2|Z}$ is the sample approximated marginal density message from the leaf clique.

We use Monte Carlo integral approximation to marginalize out X_2 in the product between message $\hat{m}_{2|Z}$ and the root clique's local odometry $\varphi_{2,3}$ and door sighting ψ_{X_3} measurement likelihood potentials. Working towards the marginal density of the last pose over three of four measurements $[\,X_3 \,|\, Z_1, Z_2, Z_3\,]$, we take the second odometry constraint.

$$
\begin{aligned}
[\,X_3 \,|\, Z_1, Z_2, Z_3\,] &\propto \int_{-\infty}^{\infty} [\,X_3 \,|\, Z_3, X_2\,]\,[\,X_2 \,|\, Z_1, Z_2\,]\,dx_2 \\
&\approx \frac{1}{N}\sum_{k=1}^{N}\left[\,X_3 \,|\, Z_3 = z_3, X_2 = \hat{m}_2^{(k)}\,\right] \\
&\approx \frac{1}{N}\sum_{k=1}^{N}\varphi_{2,3}\left(X_3, X_2 = \hat{m}_2^{(k)}; z_3\right) \qquad (6.17)
\end{aligned}
$$

where $\hat{m}_{2|Z}^{(k)}$ represents the k^{th} sample from the approximated child clique samples. The approximated marginal posterior density over all measurement potentials is $[\, X_3 \,|\, Z_1, Z_2, Z_3 \,]$ and is shown in Fig. 6-12.

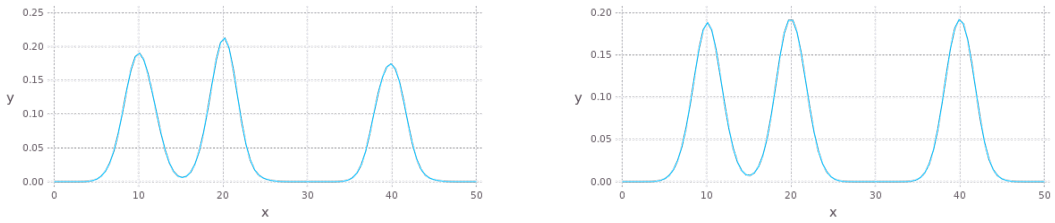


Figure 6-12: Approximated marginal of X_3 including the first three of four measurements, that is $\mathcal{P}_{X_3|Z_1Z_2Z_3}$, using of 1000 (left) or 5000 (right) samples in message m_2

Incorporating all observations

Note that the density in Fig. 6-12 still includes all three modes at pose X_3. Only after the second observation of a door, Z_4, can we discover the true mode. We can repeat the computation in eq. (6.17), but include the second observation also:

$$
\begin{aligned}
[\, X_3 \,|\, Z_1 Z_2 Z_3 Z_4 \,] &\propto \int_{-\infty}^{\infty} [\, X_3 \,|\, Z_4 \,]\; [\, X_3 \,|\, Z_3, X_2 \,]\; [\, X_2 \,|\, Z_1, Z_2 \,]\, dx_2 \\
[\, X_3 \,|\, \mathbf{Z} \,] &\approx \frac{1}{N} \sum_{k=1}^{N} [\, X_3 \,|\, z_4 \,]\; \left[X_3 \,|\, z_3, X_2 = \hat{m}_2^{(k)} \right] \\
&= \frac{1}{N} \sum_{k=1}^{N} \psi_{X_3}(X_3)\; \varphi_{2,3}\left(X_3, X_2 = \hat{m}_2^{(k)}; z_3 \right) \qquad (6.18)
\end{aligned}
$$

Fig. 6-13 shows the approximated density, $[\, X_3 \,|\, \mathbf{Z} \,]$, using Monte Carlo integration of samples all measurements as modeled in the factor graph in Fig. 6-6.

This concludes our discussion of the robot three door example. In the next section, we present an extended four door example, using our *Multi-modal iSAM* algorithm to estimate multi-modal marginals and posterior on the Bayes tree.

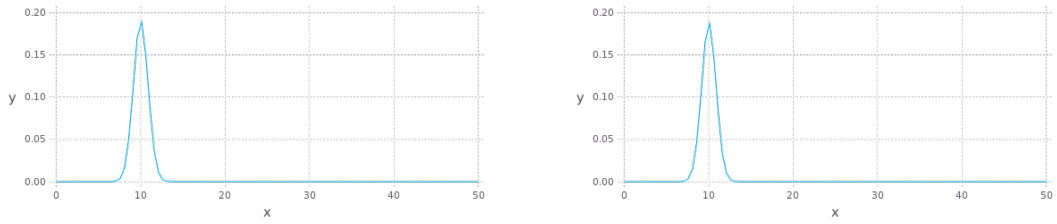


Figure 6-13: Approximated and unnormalized marginal density (using 1000 or 5000 samples, left and right respectively) of robot pose X_3 conditioned on all measurements, $\hat{\mathcal{P}}_{X_3|Z_1,Z_2,Z_3,Z_4}$. This approximation can be compared to the full analytic version shown in Fig. 6-10.

6.4 Multi-modal Posteriors on a Four Door Example

Fig. 6-14 illustrates the localization problem where a robot is driving and sensing familiar landmarks. The robot instantiates poses at points of interest along its trajectory that will form the backbone of the inference task. Please see Fig. 6-17 for the associated factor graph, as well as its re-factorization into a Bayes tree.

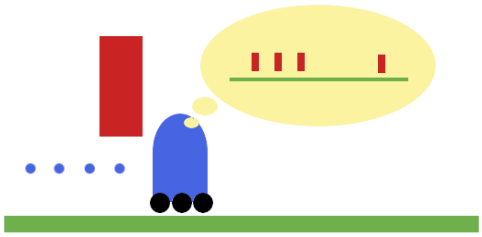


Figure 6-14: Simple pictorial of a robot moving in a one dimensional world, modified from original example in [224].

Given a prior map containing four landmarks at $z_l \in \{-100, 0, 100, 300\}$, the robot believes it must be at one of these four locations when a sighting is made. We can represent these four different landmark positions hypotheses as four modes in the associated measurement factor:

$$\psi_l(x) = \left[\hat{X} \mid z_l\right] = \frac{1}{4}\sum_{k}^{\{a:d\}} \mathcal{N}\left(x;\, \mu_k,\, \sigma^2\right) \tag{6.19}$$

6.4.1 First Three Poses

After the first landmark sighting at pose x_1, the robot moves forward 50 units to new pose x_2. After a further 50 units, a second landmark sighting is made at pose x_3. The odometry measurement is modeled with a Gaussian likelihood function

$$\begin{aligned} z_k \sim [\, Z_{odo} \,|\, X_i, X_{i+1} \,] &= \mathcal{N}\left(\mu = f_{odo}\left(\cdot\right), {\sigma_j}^2\right) \\ f_{odo}\left(x_i, x_{i+1}\right) &= x_{i+1} \ominus x_i \end{aligned} \tag{6.20}$$

where $\ominus$ is used to denote the difference on the manifold, which is important for generalization to higher dimensions. The constraint functions $\varphi_j\left(x_{i+1}, x_i\right)$, indicated by solid filled factor vertexes in Fig. 6-15, are denoted as

$$\varphi_j\left(x_i, x_{i+1}; \sigma_j\right) \propto [\, Z_{odo} \,|\, X_i, X_{i+1} \,] \,.$$

Finally, we construct a proposal sampling function, a likelihood residual, for use with eq. (5.54)

$$0 = \delta_j\left(x_{i+1}^{(k)}, x_i^{(k)};\, z_k\right) = z_k - f_{odo}\left(x_i, x_{i+1}\right). \tag{6.21}$$

The associated factor graph and Bayes tree at this point is shown in Fig. 6-15. Note that multi-modal landmark sightings, $\psi_l\left(x\right)$, are shown as red unary factors.

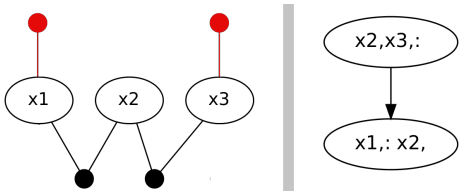


Figure 6-15: A multi-hypothesis factor graph and Bayes tree representing 100 units driving distance and two independent sightings of four indistinguishable but known landmarks.

We know that the first three landmarks are each separated by 100 odometry units. The exact belief over the three poses, x_1, x_2 and x_3, therefore have two major hypotheses. Fig. 6-16 shows the exact and estimated full marginal beliefs over x_1, x_2 and x_3 in red and black, respectively. While the exact solution contains $4^3 = 64$ modes, only the two shown in Fig. 6-16 are significant.

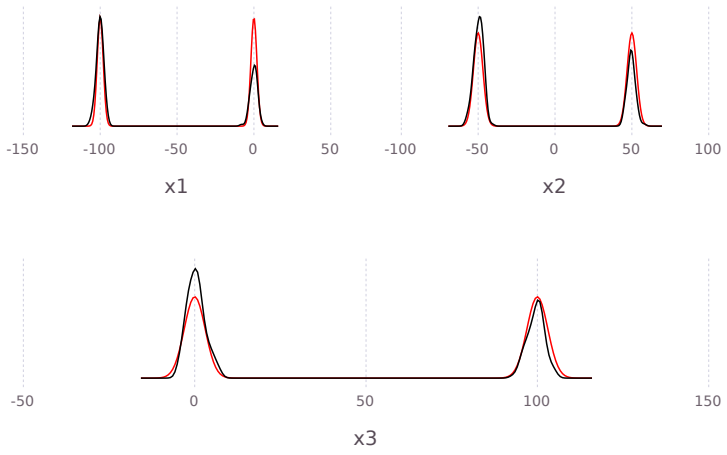


Figure 6-16: Estimated marginal beliefs, in black, of all variables of the factor graph in Fig. 6-15 following one complete up and down belief propagation pass of the Bayes tree. The ground truth belief is shown in red. Notice how the posterior has two significant modes as the final output of the inference process – a multi-modal posterior in SLAM.

We compute the estimated belief using the MCMC chain methodology described in section 5.4.3. For belief propagation, we start with the upward pass from leaves of the Bayes tree, shown in Fig. 6-15. The joint probability of the leaf clique, which contains one odometry and a landmark sighting factor, is computed with Gibbs sampling:

$$\begin{aligned} \{x_{1,1}, \ldots, x_{1,N}\}^{(i+1)} &\sim \left[\hat{X}_1 \,|\, {X_2}^{(i)}, Z_{odo} \right] \left[X_1 \,|\, z_l \right] \\ \{x_{2,1}, \ldots, x_{2,N}\}^{(i+1)} &\sim \left[\hat{X}_2 \,|\, {X_1}^{(i+1)}, Z_{odo} \right]. \end{aligned} \tag{6.22}$$

Transit conditionals, such as $\left[\hat{X}_1 \,|\, {X_2}^{(i)}, Z_{odo} \right]$, are obtainable through:

$$\left[\hat{X}_1 \,|\, {X_2}^{(i)}, Z_{odo} \right] \propto \left[Z_{odo} \,|\, X_1, X_2 \right], \tag{6.23}$$

since we only have a uniform prior on each of the sensor measurement terms. We approximate $\left[\hat{X}_1 \,|\, {X_2}^{(i)}, Z_{odo} \right]$ using eqs. (6.21) and (5.54) to construct the kernel

density estimate with LOOCV bandwidth

$$\left[\hat{X}_1 | x_2^{(1:N)}, z_2\right] \propto \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}\left(X_1;\, x_1^{(i)},\, \Lambda_{loocv}\right). \tag{6.24}$$

The second potential function on X_1 is the multi-modal prior defined by eq. (6.19). The belief product is estimated with the multi-scale product algorithm from section 5.5.1.

Frontal variables of the clique are permanently stored as the MCMC chain progresses, while separator values are only cached. Once the MCMC chain of the leaf clique has converged, we can construct the outgoing belief message from the latest cached belief over the separator variable, $\hat{m}\left(x_2\right) = \left[\, X_2 \,|\, z_{odo}, z_l \,\right]$. This message is sent to the root as per usual belief propagation.

At the root clique the algorithm is repeated to compute the posterior over all variables. Once the algorithm completes at the root, the process is repeated and new belief messages are passed back down the tree to recover all marginal beliefs. All estimated marginal beliefs are shown as black traces in Fig. 6-16.

Note that the correct mode in each case may have a lower probability, and is an artifact of the approximations used. The *sum-product* operations (as we use here) are operating correctly the with lower probability mode and istherefore not a concern as would be in a *max-product*. Max-product style algorithms would have selected the dominant mode at this point and failed.

6.4.2 A Third Landmark Sighting

In Fig. 6-17, we also see the updated Bayes tree for the larger factor graph. The previous leaf clique $x_1 : x_2$ can be directly recycled, as per [115]. Two new leaf cliques and root clique are formed, each with their own MCMC chains.

Now the robot drives 200 more units through four more pose positions, to pose x_7, where a third and final familiar landmark sighting is made. While passing through poses x_3 and x_4, sightings of a new feature of interest, l_1, is made. The new feature l_1 is included in the factor graph. The complete system factor graph is shown in Fig. 6-17.

We will not repeat the same discussion, but note that cliques at the same depth in the Bayes tree can be computed in parallel threads. The estimated and baseline marginals for all variables in the factor graph, following a single upward and downward belief propagation pass of the Bayes tree, are shown in Fig. 6-18. Notice how consensus has been reached, showing a single mode in the posterior belief.

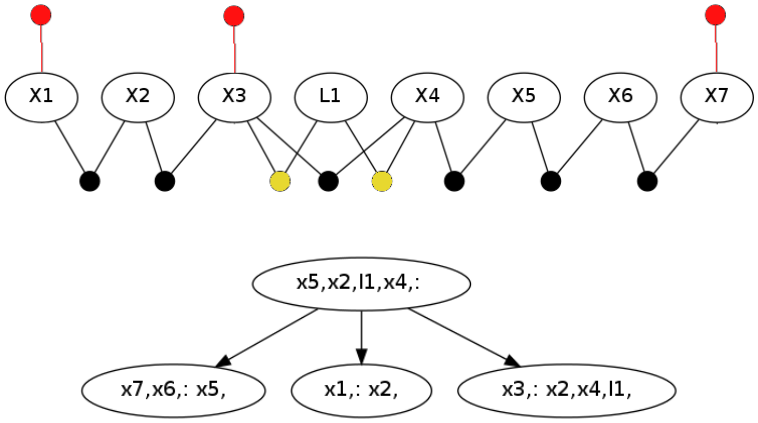


Figure 6-17: *Top*, four door example factor graph with seven poses X_* and one feature of interest L_1. Solid disks represent independent measurements from different sensor types. *Bottom*, the associated Bayes tree for elimination order $x_3, x_4, x_1, x_6, l_1, x_7, x_2, x_5$.

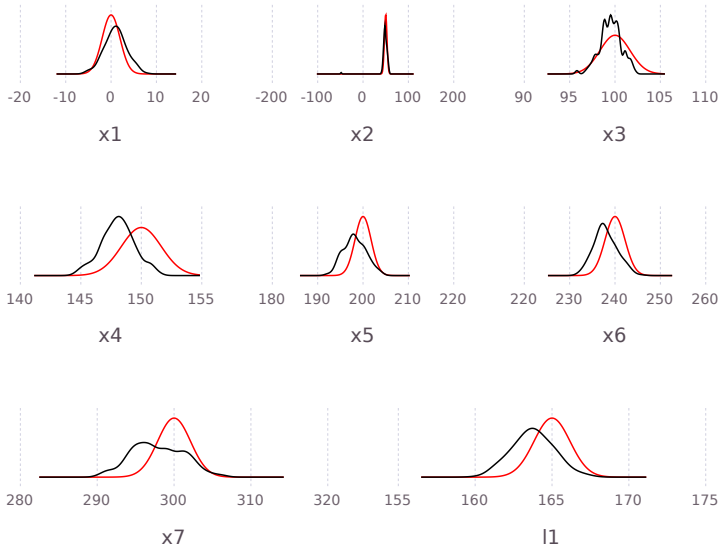


Figure 6-18: Estimated marginal belief of all variables, black traces, following one up and down pass of the Bayes tree in Fig. 6-17. Baseline marginals, shown in red, are computed via standard forward-backward solution over equivalent Hidden Markov Model solution. Note the exact marginal of X_1 can have up to 1024 modes, but only one significant mode at 0 corresponds to ground truth.

6.5 *SLAM-e-donut* example

In this section we discuss a more elaborate two dimensional SLAM example where distance only type measurement likelihoods are available; that is range only odometry, and range only landmark sightings.

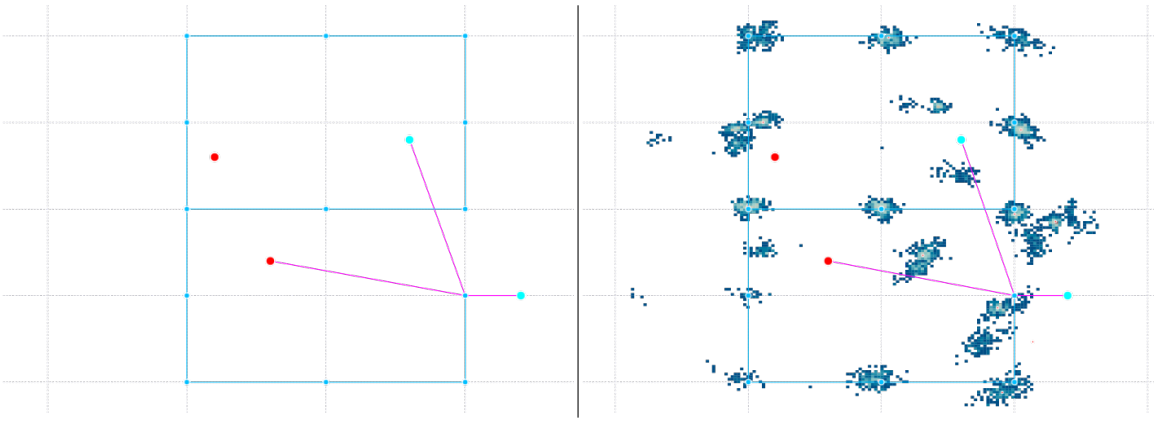


Figure 6-19: *Left*: Simulated distance only odometry SLAM example of a vehicle driving on planar surface in the shape of the letter 'e', with two known (red) and two unknown (cyan) ranging beacons. Magenta lines show a range measurement made to a unique ranging beacon. *Right*: Post inference histogram of all pose marginals, showing concentrated densities at each of the true pose locations as well as a few secondary modes. The origins of the additional modes become apparent with step-by-step analysis.

The left plot in Fig. 6-19 shows a simulated ground truth problem with four unique landmark beacons, two of which have prior known locations (red) and the other two must be mapped by the SLAM process (cyan). We assume that the vehicle is moving in straight lines between poses (cyan trace). The figures show these range measurements with magenta lines between the current vehicle position and the beacon.

Note, the straight line assumption is not needed, but simplifies explanation of this canonical example. We can readily modify this example to use more adept likelihood distributions, such as dispersed odometry likelihood density, allowing travel anywhere from the current point to maximum straight line distance away.

6.5.1 Multi-modal pose estimates

The SLAM nature of the problem is established by introducing prior location information from only two of the ranging beacons, shown as the red dots in Fig. 6-19.

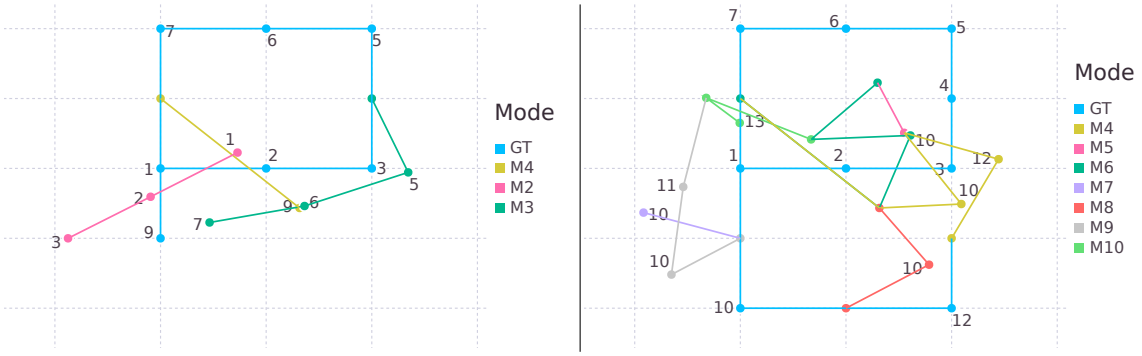


Figure 6-20: Distinct modes in each of the poses, as found by the *Multi-modal iSAM* algorithm for the SLAM-e-donut canonical example problem. The figure is simplified by grouping poses with shared modes, and showing the four groups. Notice how multiple modes collapse as more information is presented via later poses.

Note that the initial position of the vehicle is not known and neither is the position of the two cyan ranging beacons, which must be computed from the collection of distance and range only measurements accumulated during the vehicle trajectory.

To simplify the explanation, we immediately show the final inference result (histogram of all pose marginal samples) obtained for all thirteen poses with multiple beacon measurements on the right plot of Fig. 6-19. We will separately look at the landmark position marginals but note they were simultaneously computed. Note the full factor graph and Bayes tree is shown later in Fig. 6-27.

In Fig. 6-20, we interpret the *Multi-modal iSAM* posterior inference result by hand fitting separate normal densities to each of the multiple modes. The left figure shows possible trajectories for the first nine possible pose locations left, and possible trajectories for the thirteen pose locations on the right. Each of the traces are alternative trajectory modes which fit the data in some way. We study development of each trajectory mode in the paragraphs that follow.

All Pose Marginals $[X_k \,|\, \cdot]$

Top left plot in Fig. 6-21 illustrates the estimate of the first pose location, since no prior position information is available. The two red circles indicate measured ranges from two known beacon locations, which produces two possible intersection locations. The second row, left plot shows the posterior inference result with pose X_1, three measurements to landmarks $L_k,\ k = 1, 2, 3$. The contour map shows the posterior density belief on where pose X_1 is likely to have occurred, and clearly

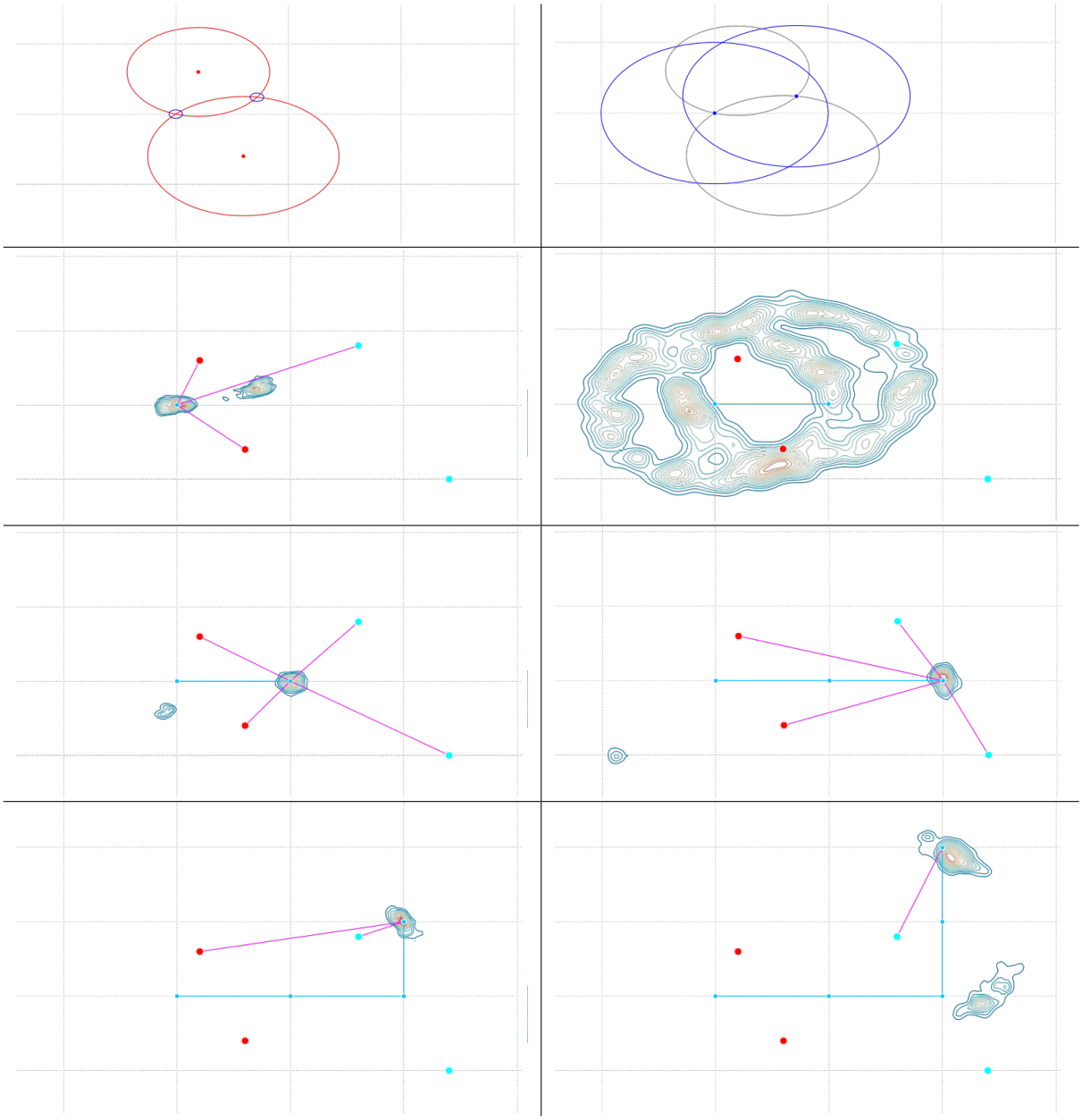


Figure 6-21: Analysis of multi-modal pose marginals $[\, X_k \,|\, \cdot \,]$, $k = 1, 2, ..., 5$ in SLAM-e-donut example (part 1). Illustration of initialization in the top row and evolving latest pose marginal from first to fifth pose position in rows from top left to bottom right. Later poses are shown next in Fig. 6-22.

shows the two possible pose locations similar to the illustration in the plot above.

The platform is moved $50\ units$ in some direction, which is illustrated by the blue circles in the top right plot of Fig. 6-21. The second plot on the right shows the marginal posterior belief for the second pose $[\, X_2 \,|\, \mathbf{Z} \,]$ over all available data $\mathbf{Z}$. The two rings in the contour maps show precisely the likely region where X_2 could have occurred.

Still at position X_2, we now introduce four beacon likelihood measurements, as shown by the magenta lines in Fig. 6-21. Note, this is the first observation of landmark L_4, but the second observation of the first three landmarks. The contour plot on the left, second from bottom, shows the posterior belief $[\, X_2 \,|\, \cdot \,]$ over all information including the new observations, where two distinct probability regions are visible. The two modes represent two possible trajectories: Fig. 6-20 depicts the true trajectory (true) with the cyan trace, and a second possible trajectory is shown in pink (M2).

The plot second to bottom on the right of Fig. 6-21 shows the third pose location and marginal posterior belief over all measurements $[\, X_3 \,|\, \cdot \,]$. Further measurements to each of the four beacons constrains the posterior to favor the correct of two modes, shown by the contour density lines. The second smaller mode is a further extension of mode M2 shown in Fig. 6-20. We note that the two unknown beacons L_3, L_4 are still being surveyed at this point; we will later see how their marginals are simultaneously converging to a stationary distribution.

The lower left plot in Fig. 6-21 show the fourth pose position inference result with posterior marginal $[\, X_4 \,|\, \cdot \,]$ having collapsed to a single mode. Only two beacon measurements are made to prior known L_1 and unknown L_3. At this point we recognize sufficient information has been collected to constrain the latest pose to a single location. We also note when the singular pose consensus marginal information from new measurements is propagated back to previous pose marginals $[\, X_k \,|\, \cdot \,],\ k = 1, 2, 3$ they too are collapsed to the correct single modes. We do not directly show the posterior marginals of older poses, but do trace the evolution in the number of distinct modes for each pose marginal in Fig. 6-23. Trajectory mode M2 is implicitly ended with the consensus found at the fourth pose.

The final bottom right plot of Fig. 6-21 shows the fifth pose. The marginal posterior $[\, X_5 \,|\, \cdot \,]$ again shows two likely regions of the state space. The two modes originate from the single beacon L_3 observation. The vehicle could effectively have continued in the upward direction, or turned sharp to the right and almost back tracked to the second possible location. Recall that we only have odometry distance and not direction of travel available in this example. The new trajectory mode is shown as M3 in Fig. 6-20.

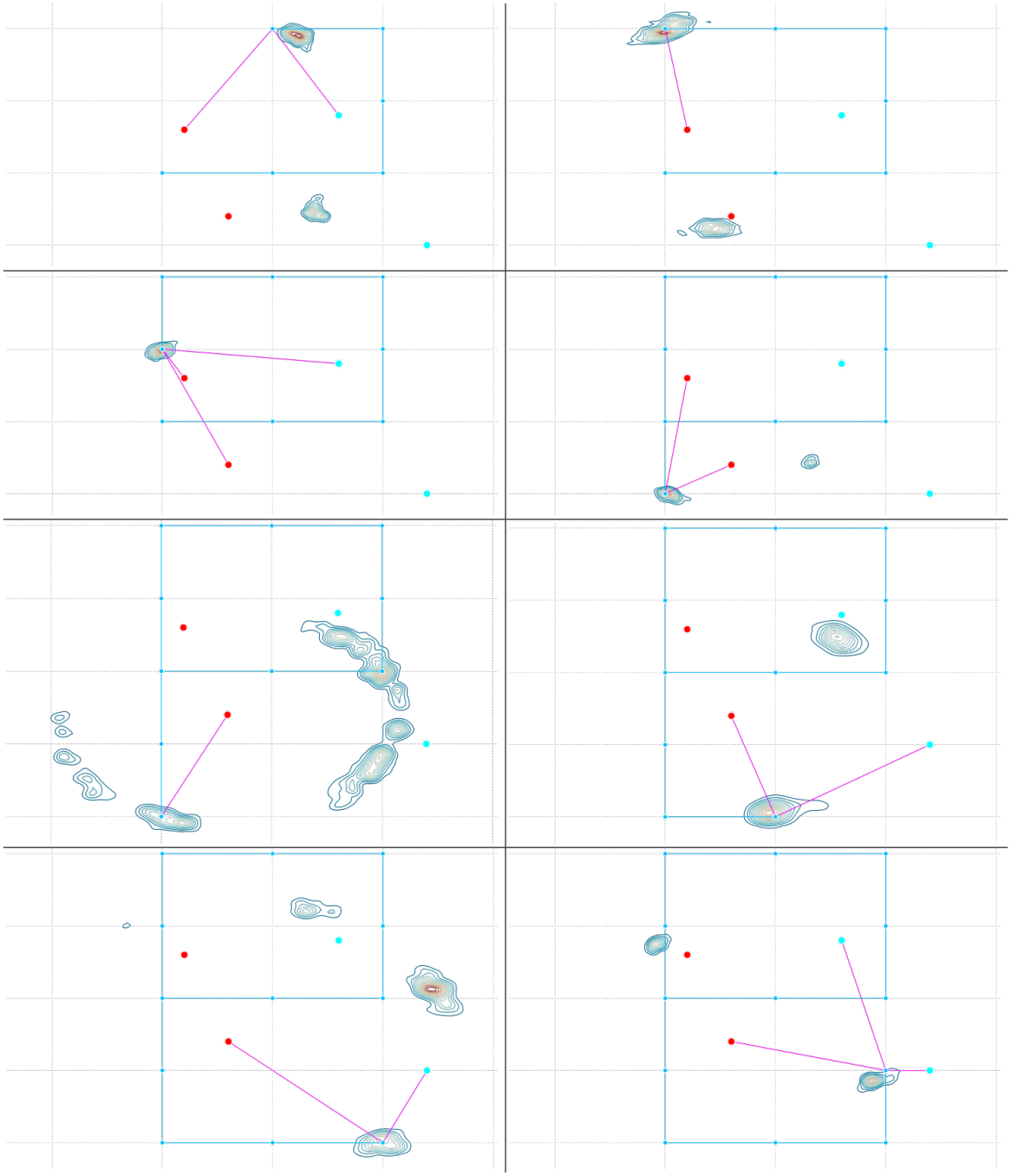


Figure 6-22: Analysis of multi-modal pose marginals $[\, X_k \,|\, \cdot \,]$, $k = 6, 7, .., 13$ for the SLAM-e-donut example (part 2), following from the first five poses that where shown in Fig. 6-21.

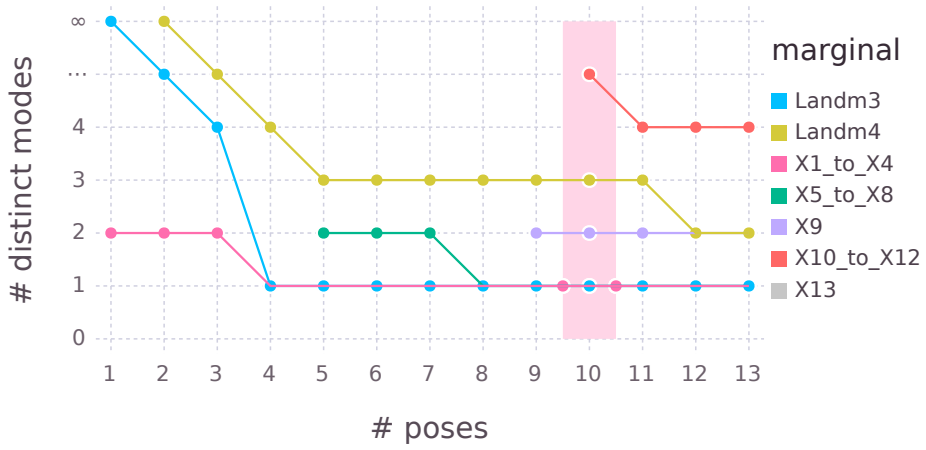


Figure 6-23: Distinct sustained modes in all pose and landmark marginals. Colors group similar variables throughout the entire robot trajectory. For example, landmark L_4 is only observed at the second pose with near infinite possible distinct locations. Notice how the mode counts collapse as more measurement likelihood information is gathered. Due to weaker constraint, L_4 maintains around three distinct modes throughout the latter part of the robot trajectory.

The first two plots, top left and top right, of Fig. 6-22 show the pose posterior marginals over all the measurement data $[X_6 \,|\, \cdot\,], [X_7 \,|\, \cdot\,]$. Two sustained modes are shown in the likelihood density contour, corresponding to trajectory mode M3 (green) in Fig. 6-20. The second left and right plots of Fig. 6-22 show a collapse in belief to a single mode in $[X_8 \,|\, \cdot\,]$ and the creation of a new second mode in $[X_9 \,|\, \cdot\,]$. The new mode corresponds to a new trajectory mode M4, as shown in Fig. 6-20.

The left, second from bottom plot in Fig. 6-22 shows a large increase in the latest pose uncertainty $[X_{10} \,|\, \cdot\,]$, with two broad arcs originating from the two modes previously observed in $[X_9 \,|\, \cdot\,]$. The large increase in uncertainty is due to a single beacon observation and lack of heading odometry information.

The right, second from bottom plot in Fig. 6-22 reduces uncertainty back to two concentrated regions of the state space. However, it is important to note that the two posterior likelihood regions of $[X_{11} \,|\, \cdot\,]$ support multiple trajectory modes, as shown in Fig. 6-20.

Furthermore, the left bottom plot of Fig. 6-22 shows four high likelihood regions, which sustain multiple trajectories. We note the low likelihood mode in the

upper left of $[\, X_{12} \,|\, \cdot \,]$ has little connection to previous modes in $[\, X_{11} \,|\, \cdot \,]$, and shows how later information can bring previously dormant modes back to the fore. The final plot (bottom right) shows two major likelihood regions for the final pose position marginal $[\, X_{13} \,|\, \cdot \,]$.

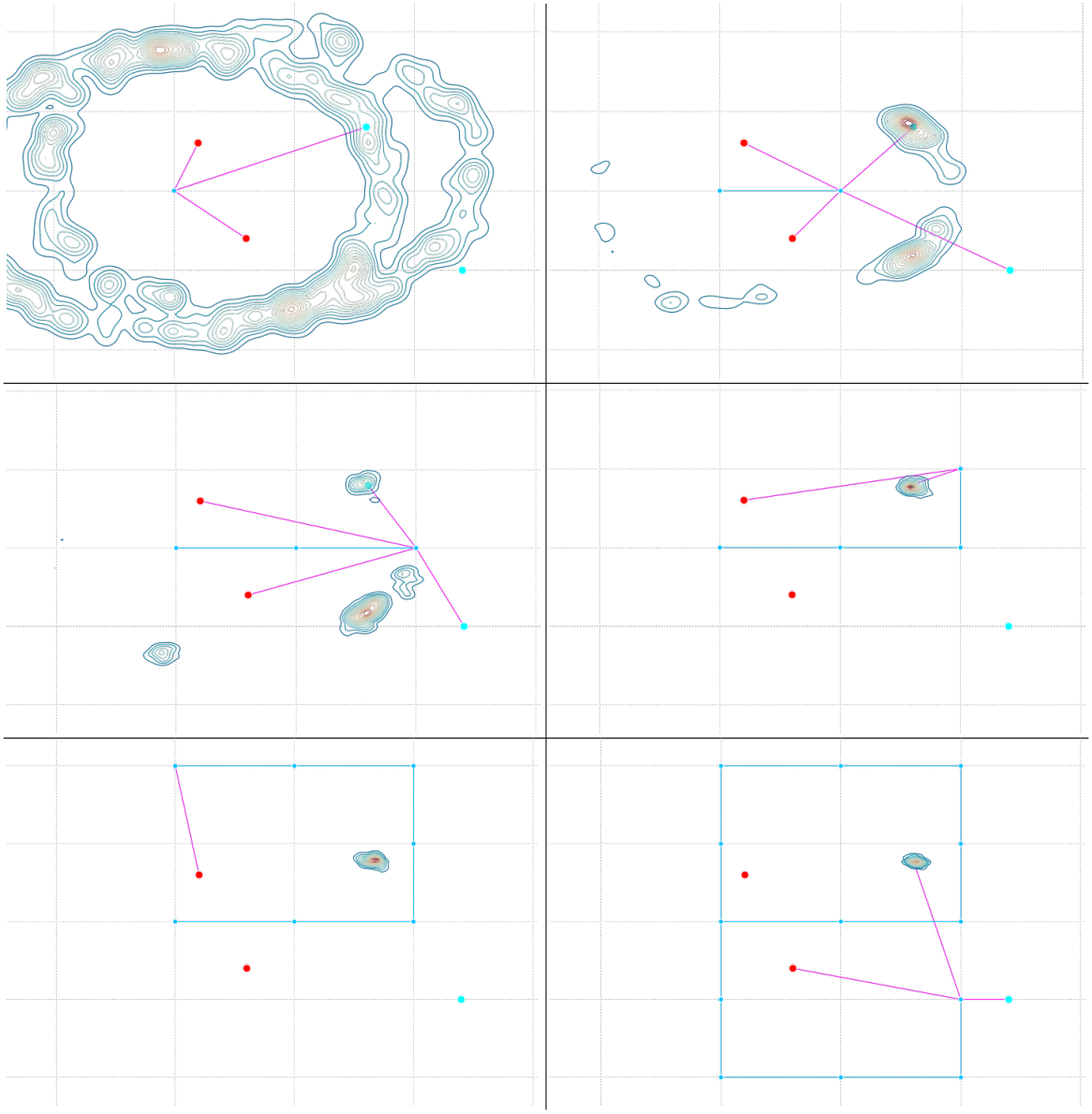


Figure 6-24: Analysis of multi-modal landmark marginals $[\, L_3 \,|\, \cdot \,]$ throughout the robot trajectory. The uncertainty shown as contour lines here are computed simultaneously to individual pose marginals estimated in figs. 6-21, 6-22.

Following the latest pose marginal beliefs in figs 6-21 and 6-22, we see a constant decrease and increase in the number of mode regions. Monitoring the number of likelihood mode regions on each of the posterior modes as the trajectory develops clearly shows consensus of a multi-modal solution, where the number of modes per marginal decreases as more information becomes available. Fig. 6-23 shows the number of distinct poses in each of the pose marginals $[\, X_k \,|\, \cdot \,]$, $k = 1, 2, .., 13$. In the next analysis we will follow the evolution of the landmark position marginals, but note here the number of mode regions in their marginals also drop to between one and three, as shown in Fig. 6-23.

Landmark Marginals $[\, L_k \,|\, \cdot \,]$

This section examines how the marginal beliefs ofthe unknown landmarks, $[\, L_3 \,|\, \cdot \,]$ & $[\, L_4 \,|\, \cdot \,]$, develop as more information is gathered. The landmark and pose marginals are estimated simulatenously. We first look at marginal belief $[\, L_3 \,|\, \cdot \,]$ in Fig. 6-24, followed by $[\, L_4 \,|\, \cdot \,]$ in Fig. 6-25.

The top left plot of Fig. 6-24 shows the marginal posterior belief over landmark L_3 simultaneous to the first pose (top left plot of Fig. 6-21). The two rings in belief (contour lines) show the range measurement from the two possible first pose locations. The second range measurement to the beacon L_3 updates the marginal posterior $[\, L_3 \,|\, \cdot \,]$ to the plot on the top right of Fig. 6-24, where a few major arc likely regions remain.

The second, left plot in Fig. 6-24 shows four major likelihood regions in the marginal posterior $[\, L_3 \,|\, \cdot \,]$, and is obtained with new measurements from the third robot pose position. The second, right plot in Fig. 6-24, corresponding to the fourth pose position, shows the landmark marginal belief collapse to a single mode, as the robot turns left. At this point we can say landmark L_3 has been mapped up to unimodal certainty. Throughout the rest of the robot trajectory, through pose X_13, the marginal belief $[\, L_3 \,|\, \cdot \,]$ remains constrained to a single mode which clearly coincides with the ground truth cyan dot beacon location.

Similarly, Fig. 6-25 shows the marginal posterior $[\, L_4 \,|\, \cdot \,]$ over all measurements as the robot trajectory evolves through all poses. We note the first measurement to beacon L_4 occurs at the second robot pose X_2 as shown in the top left plot. The marginal belief shows two rings of likelihood (contour lines) owing to a single range measurement from two possible locations in pose marginal $[\, X_2 \,|\, \cdot \,]$. The true position of beacon L_4 is the lower right cyan dot.

Top right plot in Fig. 6-25 shows the marginal posterior $[\, L_4 \,|\, \cdot \,]$ for all measurements available when the robot reaches pose X_3. Several modes are still possible

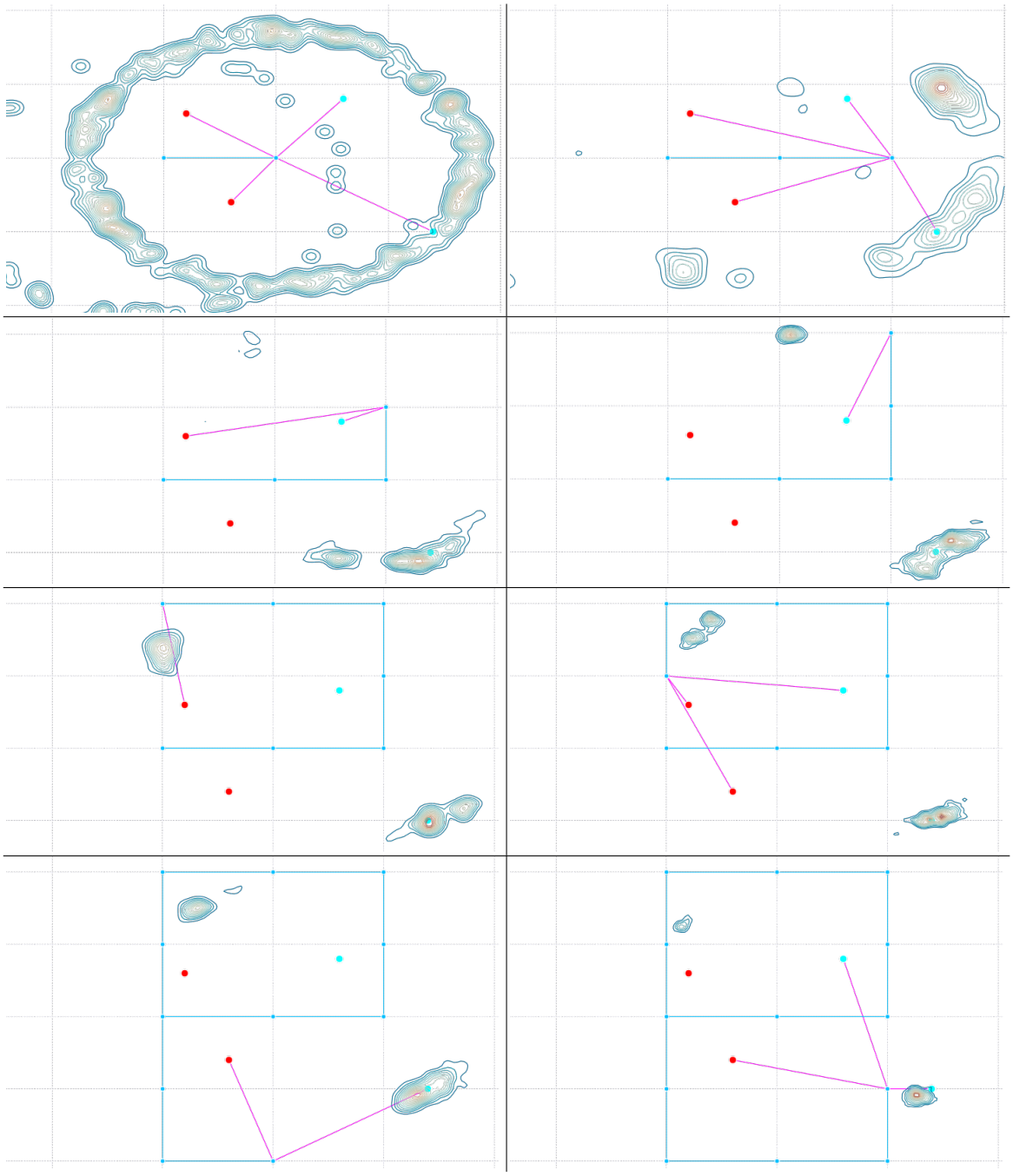


Figure 6-25: Analysis similar to Fig. 6-24 of multi-modal landmark marginals $[\, L_4 \,|\, \cdot \,]$ throughout the robot trajectory in SLAM-e-donut example.

at this point. After the robot has made its first left turn in the second plot on the left, we see there are still three to four likely regions. Only by pose X_5 has the uncertainty in marginal $[L_4|\cdot]$ be reduced to two or three major likely regions. The uncertainty of two or three distinct modes remain throughout the remainder of the robot trajectory to pose X_{13}, as shown in the bottom plots of Fig. 6-25. We summarize the sustained multi-modality as distinct mode counts in Fig. 6-23.

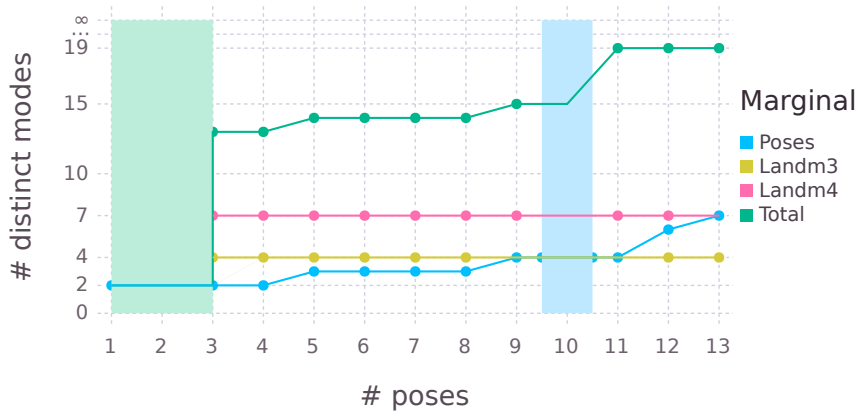


Figure 6-26: Total number of distinct modes considered in the system. See Fig. 6-23 for tracking individual distinct modes per marginal belief in the factor graph.

Fig. 6-23 shows the number of sustained modes in each marginal. By counting all the trajectory modes, we can plot the total number of modes considered by the *Multi-modal iSAM* algorithm for the SLAM-e-donut example, as shown in Fig. 6-26. A total of 19 distinct modes were considered in this problem with 30 dimensions. We count the dimensions as two directions for each of the 13 poses and 4 landmark beacons.

Fig. 6-27 shows the factor graph on the top and Bayes tree at the bottom for all variables and measurement likelihood functions. We emphasize two major points: First, each measurement likelihood in the factor graph contributes a one degree of freedom relative relation (constraint) between two variables, including hidden pose and landmark states. Second, the graph is modeled as nonparametric belief, where inference is conducted across the entire graph using the Bayes tree re-factorization. The *Multi-modal iSAM* algorithm supports incremental updates, which allows previous computations to be recycled while arriving and the full batch inference solution.

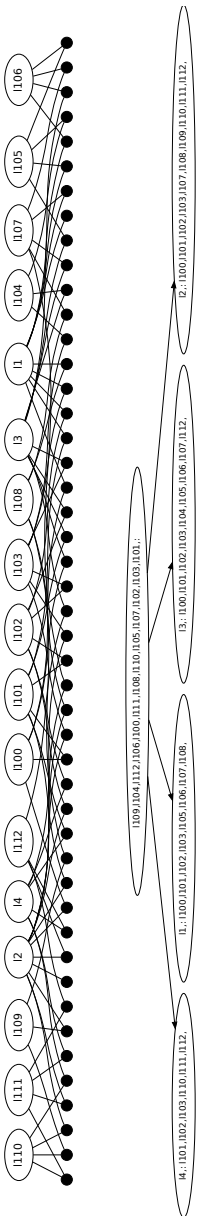


Figure 6-27: *Top*: Complete factor graph of the *SLAM-e-donut* example, where $l1$ to $l4$ are the four landmarks. *Bottom*: The associated Bayes tree, with each of the four landmarks as frontal variables in four leaves and all poses marginalized in the root.

6.5.2 Comparing Robustness and Computational Complexity

The *SLAM-e-donut* example shows how an underconstrained, Gaussian only inference problem ambiguity can be resolved with multi-modality. The correlation between robustness and computational complexity is specific to each problem. Taking the *SLAM-e-donut* example as a benchmark to evaluate the quality of solution based on computational effort spent, and stress that different non-Gaussian beliefs or factors may require a different type of analysis.

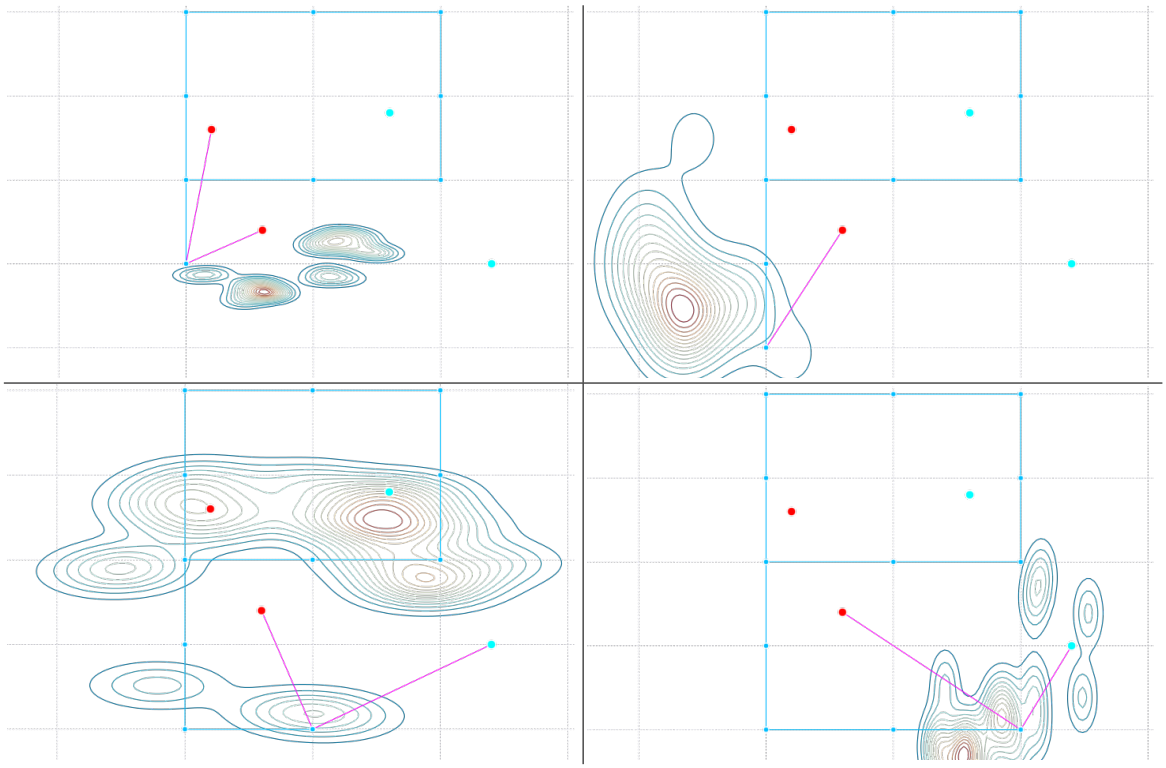


Figure 6-28: Repeat analysis of multi-modal pose marginals $[X_k|\cdot]$, $k = 9, 10, 11, 12$, using 25 particles per marginal. Comparable to Fig. 6-22 using a higher particle count.

Fig. 6-28 shows pose locations X_9, X_{10}, X_{11}, and X_{12} where each marginal posterior is estimated with only 25 particles. These plots can be compared to Fig. 6-22 where 200 particles per marginal were used. The marginal posteriors Fig. 6-28, which are a poorer approximation of the true marginal, do not match the true variable locations well. Evaluating the marginal at the true position produces a low likelihood of occurance, and a maximum likelihood point is poorly matched to the

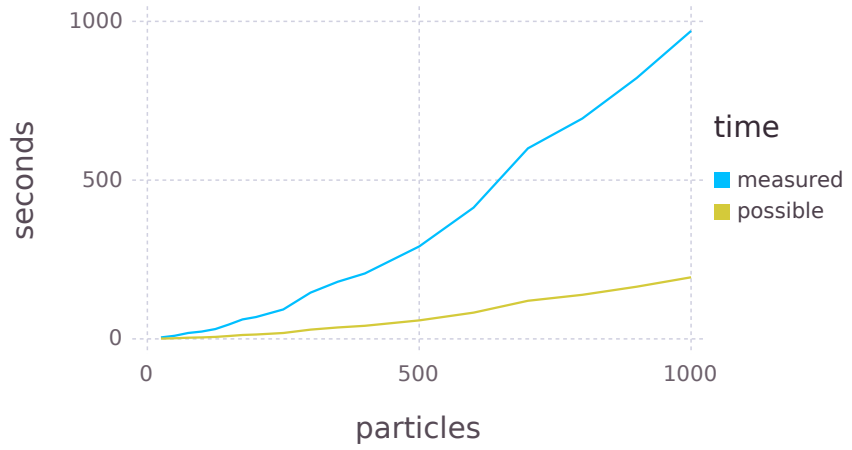


Figure 6-29: Two traces for computation time vs. number of particles per marginal for *SLAM-e-donut* example. The upper blue trace is the actual measured computation time given the current implementation. The lower yellow trace is a five fold improved estimate (made by the author and based on experience of the software implementation) of achievable performance using the current algorithm, but better with a software implementation.

true location. The following discussion quantifies these marginal posteriors using the true variable location θ_i: By evaluating the posterior likelihood $\left[\hat{\Theta}_i = \theta_i \,|\, \mathbf{Z}\right]$ and distance to maximum point on the marginal $\|\theta_i - \text{argmax}\left[\hat{\Theta}_i \,|\, \mathbf{Z}\right]\|$.

Computational complexity is evaluated relative to the number of particles per marginal used for inference and representation, while Monte Carlo chain iteration counts are kept constant throughout. Our analysis will focus on five parameters dependent on changing the number of particles per marginal: (i.) computation time on current implementation version; (ii.) posterior probability likelihoods for each variable at the true location across different particles per marginal; (iii.) L_2 error of each variable from the true location across different particles per marginal; (iv.) posterior probability likelihood of all variable marginals relative to their true location at the end of the trajectory; (v.) L_2 error between the maximum posterior marginal point and true location of all variables at the end of the trajectory.

Fig. 6-29 shows the computation time in seconds for a complete batch solve of the entire *SLAM-e-donut* system (13 poses and 4 landmarks). The actual measured

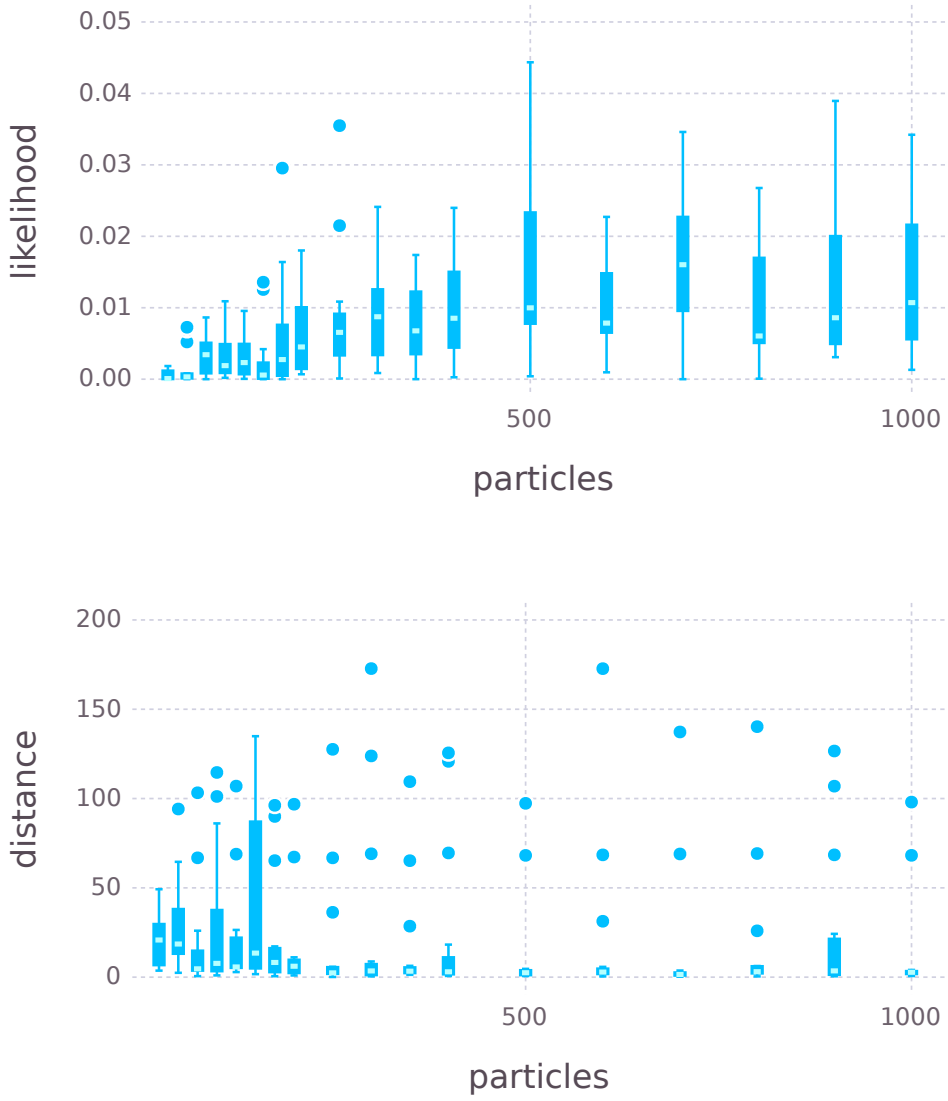


Figure 6-30: *Top*, aggregate statistics of posterior likelihood evaluations for each variable at the true value across various particle per marginal counts. *Bottom*, L_2 error between maximum point on marginal posterior and true value of each variable. Smaller errors are better.

time for a complete batch solution of the system is shown in blue. Computation was performed on an Intel i7-3920XM at $2.9\ GHz$ with $8\ Mb$ cache on a laptop computer with $16\ Gb$ of RAM, and RoME.jl implementation version v0.0.7+, [63]. The yellow line in Fig. 6-29 indicates the authors predicted achievable computation speed with a better engineered implementation of the *Mutli-modal iSAM* algorithm – the current implementation does not yet fully exploit multi-threaded and in-place memory operations. Future versions of the software implementation will include these and other improvements.

The top plot of Fig. 6-30 shows the posterior likelihood evaluation of the marginal density for each variable at the true location of the pose or beacon, $\left[\hat{\Theta}_i = \theta_i \,|\, \mathbf{Z}\right]$. The likelihood evaluation for few particles per marginal are low, given a poorer marginal approximation as shown in Fig. 6-28. An increase in the number of particles increases the fidelity of the marginal approximations and in turn increases the likelihood at the true variable location. Particular to this example, the approximated marginal densities remain similar as the number of particles per marginal exceed 400. This shows as a fairly constant likelihood evaluation in the top plot of Fig. 6-30.

The lower plot in Fig. 6-30 shows the distance between the maximum point on each marginal density from the true location of the pose or beacon, $\|\theta_i - \operatorname{argmax}\left[\hat{\Theta}_i \,|\, \mathbf{Z}\right]\|$. Low particle per marginal counts have poor estimation accuracy, but improves as the particles per marginal increase. Beyond 200 particles per marginal, particular to this example, the maximum point on the marginal beliefs of each variable start to match the true value pretty well.

The dots in Fig. 6-30 relate to the values associated with pose locations X_9, X_{10}, X_{11}, X_{12}, where multiple modes are sustained in the solution. The dots in the bottom plot represent the distance error to a higher secondary mode (the true mode has smaller likelihood). Note the row of dots at around $65\ m$ error regardless of particle count, which indicates a strong secondary mode. However, the likelihood evaluation over the same varialbes show that the true position is still supported by a likely mode in each of the marginal posteriors.

Fig. 6-31 shows a subset of the same information as used in Fig. 6-30, but instead swaps the x-axis for each variable. Different colors represent different a number particles per marginal. The top plot shows posterior likelihood evaluation on the y-axis, and as before higher fidelity marginal density approximations increase the likelihood at the true variable location. In this example, pose variables X_9 and X_{10} remain multi-modal which reduces the likelihood evaluation, as shown in the top plot. The key indicator is that low numbers of particles reduce the marginal

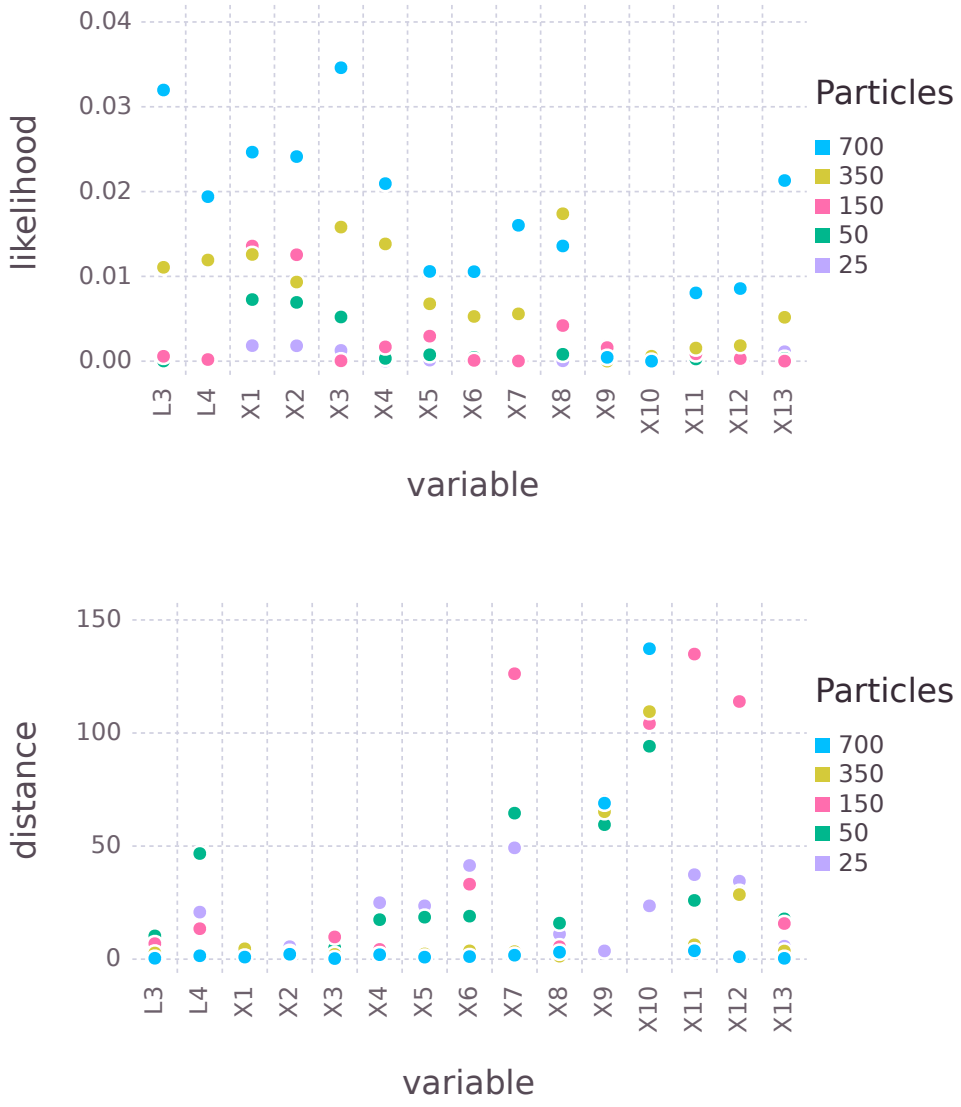


Figure 6-31: Likelihood evaluation and maximum point error of *SLAM-e-donut* example at true variable locations, across a few different particle per marginal counts. Higher values represent better aligned and higher fidelity marginal posterior belief approximation at the true variable location. L_1 and L_2 are ommited due to direct prior factors placing them at known locations.

approximation accuracy such that the true location becomes less likely (dots at the bottom of the top plot).

The bottom plot of Fig. 6-31 again shows the distance between the maximum point on each marginal estimate from the true variable location. When the number of particles per marginal are high, the distance is short – shown by the dots at the bottom. As the particle count is reduces, the marginal posterior approximation fidelity reduces and the maximum point on the density moves further away from the true variable location. As before, pose variables X_9 and X_{10} are multi-modal with the maximum belief point on a higher secondary mode.

It is worth noting a special situation visible in the bottom plot of Fig. 6-31: Pose variables X_9 and X_{10}, where the number of particles per marginal are low (25), actually have small distance errors. At first this incorrect with regard to the general trend, however, the top two plots in Fig. 6-28 show these situations. The marginal posterior fidelity is poor with only 25 particles, resulting in larger blobs of density belief, where the maximum point ends up being near the mean of all the modes. This in turn results in a lower maximum point distance error.

Lastly, we write the true positions of landmarks and poses, and note that range measurements to beacons are only made when within $150\ units$. The known beacons L_1, L_2 are at $(x, y) = (10, 30), (30, -30)$, while the unknown beacons L_3, L_4 are at $(80, 40), (120, -50)$. The 13 true pose locations are, $(0, 0), (50, 0), (100, 0), (100, 50), (100, 100)$, and $(50, 100), (0, 100), (0, 50), (0, -50), (0, -100)$, and $(50, -100), (100, -100), (100, -50)$.

6.6 Acoustic Structure from Motion (3D)

Recent work [99] in imaging sonar demonstrates a measurement ambiguity which can benefit from nonparametric *sum-product* style solutions. The previous examples were one and two dimensional canonical test cases. In this section we look at a three dimensional canonical example that includes Gaussian and uniform error error distribution. We base our example on acoustic structure from motion (ASfM) from see Huang et al. [99] and construct synthetic data for each of the three experiments in this section.

A linear sonar is mounted to the body of an underwater vehicle, and we assume that feature points of interest have been detected in a linear array sonar. Measurements to the features have tight normally distributed variance in bearing and range, however, the sensor has a wide elevation acceptance angle. We model elevation uncertainty with a uniform distribution of $\pm 0.3\ rad$.

For our examples, we are looking to estimate the corner point of a hypothetical box, shown in red in all the related figures that follow. We conduct three experiments where the underwater vehicle is moved between two poses, taking a sonar sensor measurement at each pose. The green ROV model is looking along the $x-axis$, or red axis of the pose triads. We also note that visualizations for this section were done with the Director viewer [146], using various Julia packages, such as DrakeVisualizer.jl and others, to make the visualizations possible.

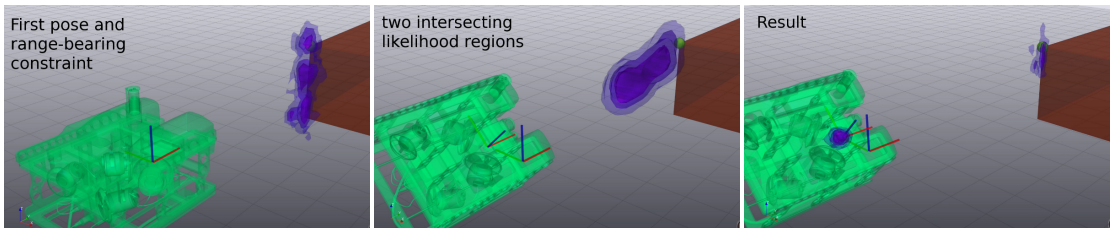


Figure 6-32: Canonical acoustic structure from motion example where a hypothetical ROV translates and rolls 45^o, observing and successfully resolving the posterior belief of where a corner feature is located, using uniformly distributed belief in the elevation measurement from the sonar.

In the first experiment, shown in Fig. 6-32, a hypothetical ROV vehicle starts out looking edge on to the red box object. The corner feature of interest is indicated by a green sphere at the top of the box. In the first pose, shown on the left, one measurement sighting is made to the corner feature. Given the uniform uncertainty in the sonar elevation angle of the sighting, we can see the vertical uncertainty contours as purple iso-contours.

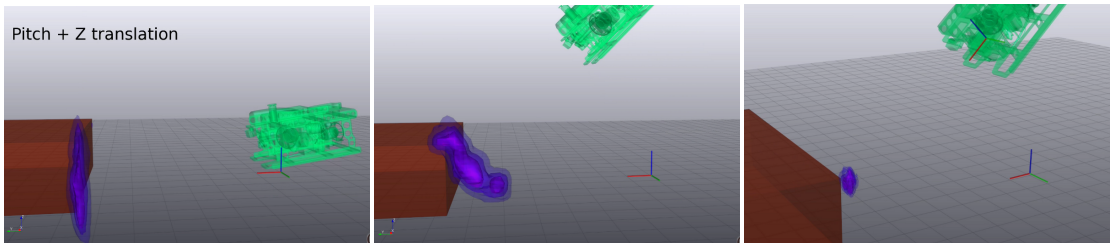


Figure 6-33: Canonical acoustic structure from motion example where a hypothetical ROV translates vertically and pitches 45^o, observing and successfully resolving the posterior belief position of a corner feature on the corner of the red box.

The center image shows the second pose, where the vehicle has moved $0.7\ m$ to the left and rolled over 45^o. We insert a usual three dimensional normally dis-

tributed rigid transform likelihood between the two poses, as an odometry constraint. A second sighting is made to the corner feature, indicated by new iso-contours. A factor graph is assembled with three variable nodes for robot positions and the corner landmark feature. Three conditional likelihoods and a prior uncertainty model included in the factor graph.

The right image of Fig. 6-32 shows the *Multi-modal iSAM* inference result. We can see that the constant probability surfaces (iso-contours) for the posterior belief over the corner feature are accurate. A second set of iso-contours is also shown on the second robot pose, by which we emphasize that both the landmark position and the robot pose position and orientation is being estimated as a true SLAM problem.

We conduct a second canonical experiment, shown in Fig. 6-33, where the ROV is again moved between two sonar measurement poses, but the vehicle is translated vertically and pitched over. We insert a three dimensional rigid transform, which is taken as normally distributed, as the odometry measurement between the poses. The left and center images show the individual proposal beliefs of where the top corner of the red box is. The right image shows the posterior estimate of the landmark position. Again, the posterior belief is the intersection between the

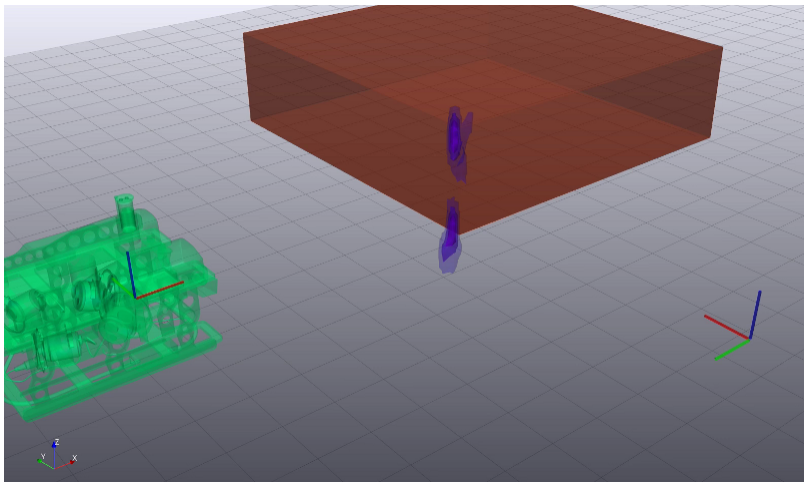


Figure 6-34: Canonical acoustic structure from motion example where a hypothetical ROV translates yaws 90^o, observing twice, and successfully resolving the multi-modal posterior belief position of a corner feature on the corner of the red box, and note in this case the posterior belief is uncertain about top or bottom of the box, is indicated by the multi-modal posterior belief estimate in iso-contours.

two measurement functions.

The last ASfM example shows only the final inference result in Fig. 6-34. The ROV vehicle was yawed 90^o around the edge of the red box, again making two sonar measurements to the corner feature. What is interesting about this example is that with the elevation uncertainty being bigger that the box, we are not sure whether the top or bottom corner was detected. The figure shows the marginal posterior belief iso-contours are multi-modal, with belief mass around the top and bottom corners of the box as we expected.

6.7 Critical Analysis

The first example, finding multi-modal square root solutions through iteration, showed how an initial belief estimate could be migrated to the final and correct posterior marginal beliefs using iterative computations. The robot three door example explored how multi-modality manifests in a sum of Gaussians approach. We were able to find the true posterior distributions analytically. Subsequently, we showed a rudimentary sampling scheme to solve the three doors example specifically, and showed the answers match the analytic ground truth solution. The three door example also clearly illustrates the idea of consensus, where low likelihood modes are practically ignored, favoring computation on the dominant modes.

Next, we upgraded our analysis to a the more general *Multi-modal iSAM* solver on an extended robot four door example. The four door example was designed to produce intermediate multi-modal posterior distributions. We were indeed able to show that the multi-modal beliefs are successfully recovered. By then adding more data with subsequent poses and door sightings, we show that consensus is also found and that all marginal posteriors become unimodal, as we expect. The extended four door example also introduced an element of mapping, by estimating the location of a previously unknown landmark. The results from the four door example exceed previous examples in the literature and contribute to our understanding of of multi-modality at the clique level, rather than multi-hypothesis across the entire trajectory as has been done in the past.

We then examined the two dimensional robot navigation SLAM-e-donut example to illustrate a true SLAM type problem, where the initial condition of the robot was not known ahead of time. Two unknown landmarks were also successfully mapped during the execution. We note that this example produced an inference problem on 30 dimensions, which was easily solved with our algorithm. This example clearly shows high levels of sustained multi-modality across all marginals,

while consensus reduces some of the modes. We believe that the SLAM-e-donut is unique in showing how dominant modes are tracked, and low likelihood modes ignored, without the user having to explicitly consider which decisions or modes should be modeled. This example shows a "black box" approach to nonparametric and efficient multi-modal solutions.

Furthermore, the modeling and construction of the factor graph to describe the problem is easily accomplished. In all the examples, especially the SLAM-e-donut example, we show that multi-modality can in some cases be synonymous with ambiguity from Gaussian-only measurement likelihoods. The SLAM-e-donut example illustrates that by using only Gaussian measurement models we are able to produce highly non-Gaussian posteriors. We do note that the non-Gaussian posteriors occur in weakly constrained situations where a degree of uncertainty remains. Regardless, our algorithm is able to approximate the correct posterior result correctly. Since the algorithm maintains the asymptotic correctness property, we know our posterior approximation can be improved asymptotically to the correct posterior at an increase of computation.

6.8 Conclusion

This chapter complements the development of the *Multi-modal iSAM* algorithm in Chapter 5. Here, we reviewed a few key canonical examples to better explain the origins and computation of multi-modality. We have shown that we are able to construct basic purely parametric or nonparametric factor graphs and solve them with our *Multi-modal iSAM* back end solver, arriving at highly non-Gaussian (nonparametric) posteriors. The examples are simple enough to verify by hand, but show how the complexity in understanding the results quickly grows. We have shown how correct nonparametric posterior approximations are found by carefully interpreting the algorithm output. In the ASfM example, we showed that a multi-modal solution in three dimensions is also possible, and that other parametric models such as uniformly distributed elevation angle can be natively supported by the approximated *sum-product* inference algorithm.

Chapter 7

Centralized Factor Graph

Chapter 3 discusses how to model a joint probability belief in a factor graph using a variety of sensing modality likelihood models. Chapter 5 discusses how to perform robust multi-modal inference over the factor graph model. This chapter argues that navigation processes should be centralized around the factor graph in a server-client model, rather than the front-end process, to achieve separation-of-concerns between all stakeholders. We also promote the use of starved graph database technology for reasons including concurrency, large data throughput, and multi-language support. In Chapter 8, we present and discuss an example use case with a Turtlebot wheeled robot. Work in this chapter was published in collaboration with S. Claassens, S. Pillai, R. Mata and J. Leonard [64].

7.1 Introduction

Realizing tangible value from robotic data requires a versatile and highly-accessible data representation. Novel database technologies provide advantages in representation, manipulation, and extraction. We argue the benefits of graph databases in robotics by exploring spatio-temporal representation and centralized persistence. This enables situationally-aware querying and inference of the robot's task-specific state at any point in its history. We view this as a critical component in the robot's ability to learn newer representations from previous experiences.

We propose a two-tier persistence architecture as depicted in Fig. 7-1 that maintains independent databases for: (i) A graph that provides both the master data index as well a store for relational, queryable data (e.g. the robot's Maximum-A-Posteriori state estimate); (ii) A key-value store retaining arbitrarily large sensor data (e.g. RGB and Depth (RGB-D) imagery and laser scans).

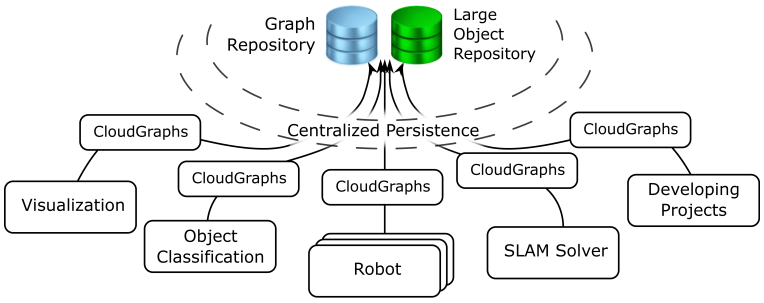


Figure 7-1: Conceptual description of a navigation graph centralized robot data persistence and recall system. The graph database provides an efficient spatio-temporal search index while the larger sensor data is retained in the key store.

Through potential use-cases and experiments, we emphasize the flexibility of graph databases for a multitude of concurrent tasks: (i) Post-hoc loop closure detection and incorporation; (ii) Continuous factor-graph solving for simultaneous localization and mapping; (iii) Incorporation of post-processed information such as per-pixel semantic segmentation of individual keyframes; (iv) Querying relevant scene entities as shown in Fig. 8-22.

Next we present three major themes relating to a central navigation and data store system. We showcase three data queries which illustrate the benefits of a central database system in robotics, followed by related work and limitations. We present an implementation discussion and experimental results using a Turtlebot robot [182], discussed with other results in Chapter 8.

7.2 Principal Themes

Our approach emphasizes three core requirements in a robotics system: *Random access via query* relating to interaction with collected data in unpredictable ways; A *Centralized architecture* relating to in-situ interaction with the data from multiple, weakly independent processes; and *Horizontal scalability* relating to offloading large computation in real-time.

7.2.1 Random Access Queries

Insertion, modification, and extraction from any persistence framework should be as efficient as possible. The underlying complexity of this requirement is that an

architect has to address the variety of different roles. We are proposing that a pose-keyed graph fulfills a good balance of ease-of-access with insertion speed. The natural structure of the graph also allows custom keying to both index and retrieve specialized datasets from the collective data (such as keying data by tracked features).

Three example queries are provided in Section 7.4 to demonstrate how the graph database enables random-access queries.

7.2.2 Centralized Architecture

Local processing of sensory data reduces the need for large-bandwidth communication systems, however it also isolates the data and places large demands on the processing power encapsulated in the robot. This is relevant in exploratory robotics, where data volumes, bandwidth and latency are fundamental concerns during design. Domestic and urban environments, however, offer greater communications bandwidth but large scale robotic processing power remains expensive.

The question becomes one of balancing local robot processing with large scale centralized processing. In these settings, centralizing the data would allow: (i) long-term data accessibility by any number of agents; (ii) minimization of local, client-side processing; (iii) sharing of data and analysis between different sessions and robots; (iv) aggregation and refinement of the collective data (e.g. summary maps).

7.2.3 Horizontal Scalability and Concurrency

Requiring horizontal scalability enforces that the underlying data structure may be processed by multiple independent applications. We argue that a horizontally scalable system is advantageous as we are not constraining the system to a single application. Rather, new agents may be introduced at will, and the underlying structure should support operation by any number of concurrent applications.

7.3 Graph Databases for Robotics

We argue that relational databases are a critical step forward as they provide: (i) a rich scalable data structure where robots and task-specific processing are not required to store the complete graph in-memory; (ii) centralization of the data persistence layer so interacting-processors may operate concurrently on the shared

data; (iii) a relational query language for powerful data extraction where only the relevant data is returned to the client.

However, using a relational database for data storage has drawbacks, as: (i) it requires the graph be represented as tables and tabular relationships, which causes unnecessary, constant translation from graph to tabular structure; (ii) it limits flexibility as the relational structure must be defined beforehand and is difficult to modify in situ; (iii) large sensory data will bloat the database with binary blob elements, negatively affecting the overall database performance.

Classic relational databases represent data and relationships using a tabular structure. Similarly, graph databases represent data and relationships using a native graph datastructure. Unlike statically typed tables in relational databases, graph databases allow for richer data types and complex relationships [1]. Accompanying the richer elements is a query language that can interrogate and traverse the data structure.

We propose a native graph database and two-tier storage structure to offload large sensory data to a NoSQL key-value store. NoSQL key-value stores are designed for efficient insertion and extraction, which is highly applicable for persisting the larger sensory data. This is done to realize a 'starved graph', i.e. a graph that can be efficiently leveraged for querying often-used data and relationships, with the ability to extract the larger binary on demand (image and sensor data). The keys are stored in the graph, providing a link between the two systems. Users of the persistence system choose which data should be included in the graph and which should be offloaded, and this can change dynamically. This architecture allows additional systems and data to be bound to the graph without bloating the structure.

Prior to discussing implementation or experimental results, we wish to highlight the advantages of the centralized robot graph database with a few working examples. Consider that in all cases the query execution is performed on the server-side, resulting in a small but relevant fraction of the complete dataset being transferred to the client.

7.4 Working Examples

Three example queries are chosen to illustrate how the structure can simplify otherwise complicated random access queries. We use Cypher syntax, a declarative graph database language that is native to Neo4j [4].

```
MATCH n:POSE
WHERE n.timestamp = max(n.timestamp)
AND exists(n.SLAM_Estimate)
RETURN n.label, n.SLAM_Estimate
```

Figure 7-2: Retrieving the latest refined pose estimate.

```
MATCH n:POSE
WHERE n.timestamp = min(n.timestamp)
AND exists(n.SLAM_Estimate)
RETURN n.label, n.SLAM_Estimate
```

Figure 7-3: Retrieving the initial refined pose estimate.

7.4.1 Temporal Queries

Computing the start and end positions from a ROS bag [192] or datafile involves scanning the complete dataset. The *SLAMinDB* graph representation allows a succinct query to perform server-side searches from indexed properties (such as timestamp). Given that `n.timestamp` is the timestamp parameter and `n.SLAM_Estimate` we wish to extract:

The solver is computing the SLAM solution concurrently—captured by the `exists(n.SLAM_Estimate)`—if this exists then we have a valid SLAM estimate. Omitting it would retrieve the latest raw pose in the SLAMinDB implementation. As the solution is calculated in-situ, the simple conditional create an index-searched operation that is arguably difficult with flat file datasets.

Similarly we can retrieve the start location of the refined graph by simply changing our search criterion:

In multi-agent scenarios, the graph retains the complete history of all agents. Queries can be constructed that relate information of interest (such as identified objects, proximity, or time) to the cumulative history of the robots. Additional indexing allows the results to be efficiently extracted. An example of such an extraction would be retrieval of the latest pose (`n.SLAM_Estimate`) and sensor data (`n.bigData`) for `ROBOT1` during its fifth session, `SESSION5`:

```
MATCH (n:POSE:ROBOT1:SESSION5)
AND n.timestamp = max(n.timestamp)
AND exists(n.SLAM_Estimate)
RETURN n.label, n.SLAM_Estimate, n.bigData
```

Figure 7-4: Retrieving the latest refined pose estimate for ROBOT1 during SESSION5 run.

7.4.2 "Foveation" and Spatial Queries

In addition to temporal queries, the graph can leverage position as a filter. A useful example of such a query would be to extract all nodes within a vicinity and within view. This is referred to as a foveate query and can be implemented by modifying the WHERE clause. Additionally we are making use of two server-side user-defined functions which augment the query language with SLAMinDB-specific functionality. The following query will return all factor graph nodes with a [2,5] meter range and within a 45^o field-of-view:

```
WITH [0, 0] as position, pi/4 as fov
MATCH (n)
WHERE
// Region filtering
        cg.withinDist2D(n, position, 2, 5)
AND
// Frustum cutoff filtering
        cg.withinFOV2D(n, position, fov)
RETURN n
```

Figure 7-5: Foveation calculation as Cypher query.

User-defined functions and procedures can address scenarios where often-used queries should be encapsulated, or in cases where procedural steps are required. For reference, we have included a simplified form of the `cg.withinFOV2D` function:

User-defined procedures allow more comprehensive code to be embedded server-side and if (as in the case above) we can optimize the search on the server, the foveate query can be succinctly expressed as:

```
@UserFunction
public boolean withinFOV2D(
  @Name("node") Node node,
  @Name("position") List<Double> pos,
  @Name("fov") double fovRad
) {
  double[] pose = (double[])node.getProperty("MAP\_est");
  double poseAng = pose [2];
  //Calc pose-forward and pose-to-position vectors
  SimpleMatrix
    pose2POI = toVec(pos.get(0)-pose[0],pos.get(1)-pose[1]),
    poseFor = toVec(Math.cos(poseAng),Math.sin(poseAng));
  pose2POI = pose2POI.divide(pose2POI.normF());
  //Use dot product to determine if within FOV
  return Math.acos(pose2POI.dot(poseFor)) <= fovRad;
}
```

Figure 7-6: Simplified form of the cg.withinFOV2D user-defined function in Java.

```
WITH [0, 0] as position, pi/4 as fov
CALL cg.foveate(position, 2, 5, fov)
RETURN n
```

Figure 7-7: Foveation calculation as user-defined procedure.

7.4.3 Interactive SLAM

Graph relationships provide rich functionality for generating and traversing elements. Consider two concurrent processes in SLAMinDB: one processing the sensor data and suggesting potential loop closures, and the second processing a parallel solver consuming the changes when suitable. The two agents can be on different systems, operating with minimal interaction as fully decoupled applications.

The production of the loop closures could, for example, be a supervised application. In the event that a loop closure is confirmed, an edge can be introduced to indicate to the solver that it has new relationships to process. This can be done in the graph with the following update query:

When the SLAM solver is in a suitable state to integrate the new relationships,

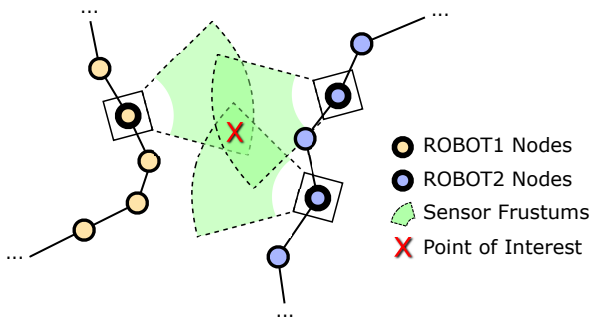


Figure 7-8: Illustration of foveation query and node selection for multiple robots in the centralized SLAM-aware database.

```
MATCH (n:LANDMARK), (m:LANDMARK)
WHERE n.label="110" and m.label="184"
SET (n)-[r:SAMELANDMARK]-(m)
```

Figure 7-9: Introducing edges in the case of a loop closure.

it can consume the unique edges and produce the correct function nodes:

```
MATCH (n)-[r:SAMELANDMARK]-(m)
RETURN n.label, m.label, r
```

Figure 7-10: Solver retrieval of edges during client-side graph update.

The refined graph is updated in place, resulting in incremental improvement of the dataset without processing off-line or requiring complex handshaking.

7.5 Implementation

The experimental implementation makes use of a Neo4j graph database [4] for the graph persistence and a MongoDB NoSQL database [3] for the key-value store[1]. The codebase was developed principally in Julia [18] with the robot front-end

[1]The code for the experiment is available at `https://github.com/dehann/Caesar.jl` and `https://github.com/GearsAD/CloudGraphs.jl`

and image recognition in Python[2]. The integration of both the Multi-Modal iSAM solver (Caesar.jl) [62] and the underlying data persistence layer (CloudGraphs.jl) is referred to as SLAMinDB.

On insertion, data partitioned into four categories: (i) Labels for indexed searching; (ii) Properties for searching and filtering; (iii) Packed data for local vertex storage; and (iv) Large object storage for MongoDB storage.

Labels and properties can be used to build relational queries and traverse the graph in task-specific ways. Vertices, for example, can be labelled when AprilTags [179] are detected, and the properties can contain more detail about the specific tag. The packed data permits critical binary data to be stored in the vertices - a compromise between the large data store and simple properties. Custom labelled edges can also be inserted by developers to insert task-specific traversals which augment the existing factor graph.

Large sensor data is trimmed from the vertices and persisted in MongoDB. The vertices are appended with the MongoDB keys to maintain the relationship between the two stores. The separation is opaque to graph consumers, which operate via the CloudGraphs API, splitting and re-splicing as the data as required.

7.5.1 Illustrating Concurrency and Random Access

We are also particularly interested in the ability to offload computation from the robot, whereby more post-hoc / in-situ agents, or human operators, can interact with the data in a rich manner.

In our experiments, we used a Turtlebot outfitted with a RGB-D structured light camera and demonstrate tangible experimental results relating to the three main themes discussed in Section 7.2: Random query access, centralized and atomic transactional structure, and horizontal scalability. The robot is tele-operated in an AprilTag-laden office environment for benchmarking purposes. Through an interactive procedure, we incorporate cross-session navigational loop closures—via AprilTags—as constraints to the SLAM factor-graphs from each session.

Figure 8-23 shows a visualization of the merged map reconstructed from the Turtlebot data. The local processes, designed to be light-weight, communicates relevant measurements back to the central database for SLAM solver consumption. Typically, the central database is hosted on a more powerful server computer, while the the robot connects to it through a low-bandwidth network. The local robot processes incrementally push new measurement factors and variables to the centralized database, all while the database consumes and solves with these added

[2] Pybot is available at `https://github.com/spillai/pybot`

constraints independent of each other. Larger key-value entries such as raw color and depth imagery from the RGB-D sensor are stored in a local MongoDB instance. These entries are synchronized with the server MongoDB data store at lower priority as permitted by network availability. Additionally, we generate globally unique identifiers at the initial commit to the local store, which will become globally available as the network availability allows.

At any point in time, the robot may choose to run its own queries against the central database. For example, the latest available Maximum-A-Posteriori (MAP) location estimate (independently computed by the SLAM agent) can be required using the query presented in Section 7.4.1. Depending on network traffic and length of network between the robot and server, this query nominally takes on the order of tens to hundreds of milliseconds to run. We'd like to emphasize that the query sent and returning result are only small single line text strings, with most computation happening on the central server.

The server computes and then returns the latest available SLAM pose estimate. The ability to make such queries has made an otherwise complicated processes remarkably simple. The robot may now incorporate these return results, registered against a previously known pose ID to improve its own location estimate. A process not entirely dissimilar from how the Apollo spacecraft navigated to the moon and back, resetting the spacecraft's local estimates using Earth-based radio navigation and computation.

7.6 Conclusion

This work demonstrates the applicability of a centralized factor-graph for data persistence. By leveraging the natural structure of the factor graph as the principal data representation in a robotic mobile system we realize a model that supports: (i) Random access for data; (ii) Centralization of data; (iii) Horizontal scalability. There are several advantages of relating most robotic data to a single versatile index, which are demonstrated in the examples. While centralizing the factor graph in such a manner may at first not seem an optimal choice, we show that the advantages of using dedicated data layer technologies far outweigh the marginal increase in complexity to existing robotic navigation and recall systems, and that this is superior to local, independently operating solutions, especially with regard to cooperative agents.

Chapter 8

Experimental Results

In Chapters 3 and 4, we developed a method to combine parametric and nonparametric measurement likelihoods into a joint probability factor graph description, and we described four factor groups. In Chapter 5, we developed the *Multi-modal iSAM* inference algorithm to resolve the marginals of the joint posterior belief, and in Chapter 6 looked at a few canonical examples. This chapter reports the results from practical examples, individually showing the three of the four factor groups. Chapter 9 will interpret and conclude the discussion of this thesis.

8.1 Introduction

In this chapter we present findings from five different experiments. Each experiment illustrates different combinations of the factor groups II, II, and III – see Section 3.4.1 for details. The first two of five experiments use the new inertial odometry factor with a parametric optimization, while the last three of five experiments use the *Multi-modal iSAM* solver.

Each of the five experiments require a separate front-end pipeline to construct the associated joint probability factor graph. To illustrate the commonality between the five experiments, we refer back to a general joint probability factor graph illustration from Fig. 3-1 in Chapter 3. In each case, we highlight the variables and factors of interest, and note that slight variations are required to the common joint probability figure, but point them out as we go along.

The five experiments include (i.) hand-held monocular camera and MEMS inertial in a VICON motion capture room; (ii.) Data from the BDi Atlas humanoid robot, including tactical grade inertial measurements, leg kinematics, and monocular feature tracking from a head mounted camera; (iii.) the Victoria Park car driv-

ing data set [168], including wheel and steering angle encoders along with range and bearing object measurements; (iv.) various data from a wheeled Turtlebot robot autonomously exploring an indoor setting; and (v.) beam formed acoustic data from an autonomous underwater vehicle.

8.2 A Free-Motion Device (Group I & II)

We conduct two experiments to showcase how bearing only camera features can be used alongside inertial odometry. A follow-on experiment then includes a few sparse AprilTag sightings as loop closures, which are introduced through three dimensional range and bearing measurements. All visual sightings are made with a monochrome monocular camera from manufacturer Point Grey.

The monocular experiment presented here is divided into two cases: First, a sparse corner feature tracking only case. Second, we reuse the same trajectory but then include a few AprilTag detections to act as visual loop closures constraints in the factor graph. Before discussing the 'with' and 'without' loop closure cases, we briefly look at the visual feature tracking scheme used.

8.2.1 Brief on Monocular Feature Tracking

The left side of Fig. 8-1 shows a single frame with some feature points being tracked from one of the test trajectories of a factor graph based visual-inertial system. Longer feature tracks of rigid corner points through the image sequence improves the solution quality. Longer feature tracks also minimize duplicate features, which are lost and regained, as the trajectory progresses.

Maintaining tracking lock of a visual feature under significant scale, rotation and background variation is a major challenge. We had initially tried feature detection and descriptor matching BRISK [136], ORB [200], SURF [15] and SIFT [139], but found feature tracks to be relatively short at best.

We also tried the Kanade-Lucas-Tomasi feature tracker [21, 142, 207] – using Birchfield's implementation [22] – but again found the feature tracks to be relatively short and with multiple outliers, even with the affine consistency improvement enabled [22, 230]. Dynamic objects in the scene are a further concern with both the aforementioned tracking schemes.

In an attempt to increase feature tracking length and robustness, we developed our own predictive template matching style feature tracker similar to [157]. Tracking is achieved by selecting Harris corners [87] as good features to track around which a template region is extracted.

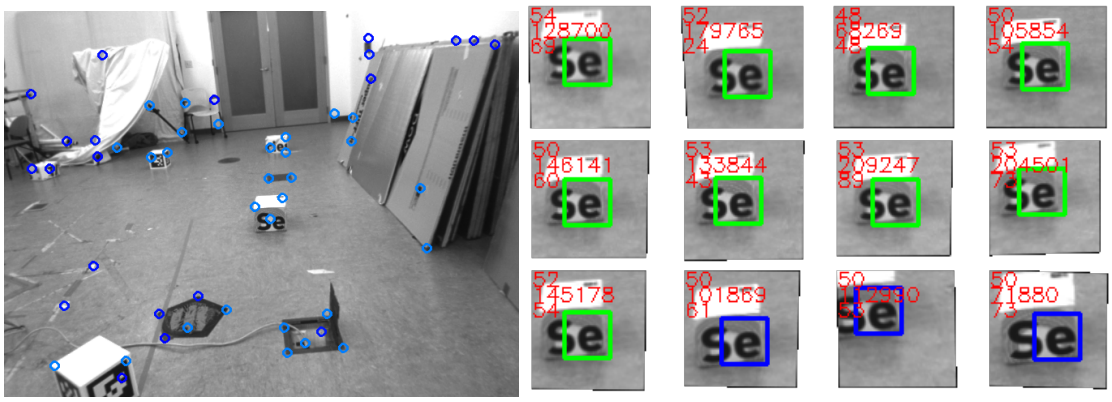


Figure 8-1: Left: Example image from monocular camera, tracking Harris corner features in the scene. An AprilTag is also visible, and is used for visual loop closure constraints – five tag markers were randomly placed in the environment, with no prior knowledge as to their location. Right: Example segment of feature point tracking across several key-frames. The square indicates chosen match location, while the numbers top-left indicate various parameters of the matching process.

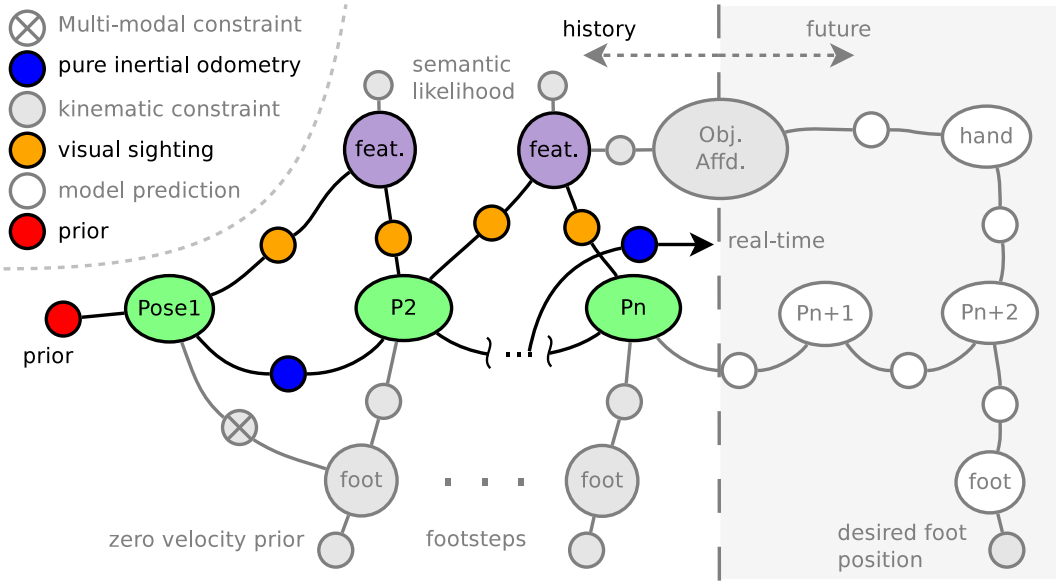


Figure 8-2: Joint probability factor graph, for a visual-inertial experiment with factors from Group I and II, where blue circles are pure inertial odometry factors for IMU measurements and yellow factors represent feature tracking measurements from a monocular camera rigidly attached to the IMU.

Using the real-time inertial odometry solution we predict how feature patches will move in sequential images and extract regions of interest for each image. Still using the inertial prediction, an approximate inverse homography transform is then applied to the current predicted region of the interest patch, to transform back to the reference pose from which the template patch was extracted.

A new feature point is selected or discarded based on a testthat includes a high convolutional match score, the clear existence of a Harris corner feature, and low ambiguity with other Harris corner features in the region of interest. A short excerpt of feature tracking is shown in Fig. 8-1, where each of the steps above have already been performed. Each of the images is down sampled at each keyframe (pose location) in the trajectory, and represents the bearing only measurement made to that feature.

8.2.2 Retroactive Visual-Inertial Trajectory

Using a Microstrain GX3-25 IMU and monocular camera feature tracks as discussed above, we can assemble a joint probability factor graph as illustrated in Fig. 8-2. Inertial odometry measurements are taken between discrete pose locations. Each pose has a corresponding keyframe image, as illustrated with Fig. 8-1 and discussed above.

In this case, most of the system was implemented in C++, such as visual feature tracking. The inertial odometry factor implementation was done with a C++ modification to the iSAM1.7 library, whilecthe actual preintegration process was performed with a Julia implementation. The inertial odometry factors and preintegration of measurements, compensation gradients and predicted error covariance is as described in Chapter 4.

Two datasets were captured with camera and IMU rigidly attached and walked by hand inside a VICON motion capture room. The MIT LCM data logging and playbag toolchain [95] was used to store and process all data. We use the VICON motion capture system to extract the initial condition of the trajectory, which would then allow us to evaluate the trajectory estimate accuracy. The VICON room dimensions are approximately $12 \times 7\ m$.

We stress that our trajectory evaluations are not a post inference fit to the VICON data, but rather a free trajectory with only the first pose in common with the VICON setup. The errors we report include all drift factors accumulated starting from the first pose. In particular, we also note that scale in the solution is obtained through double integration of accelerometer measurements.

Each of the two datasets were reprocessed twice, one using only visual feature

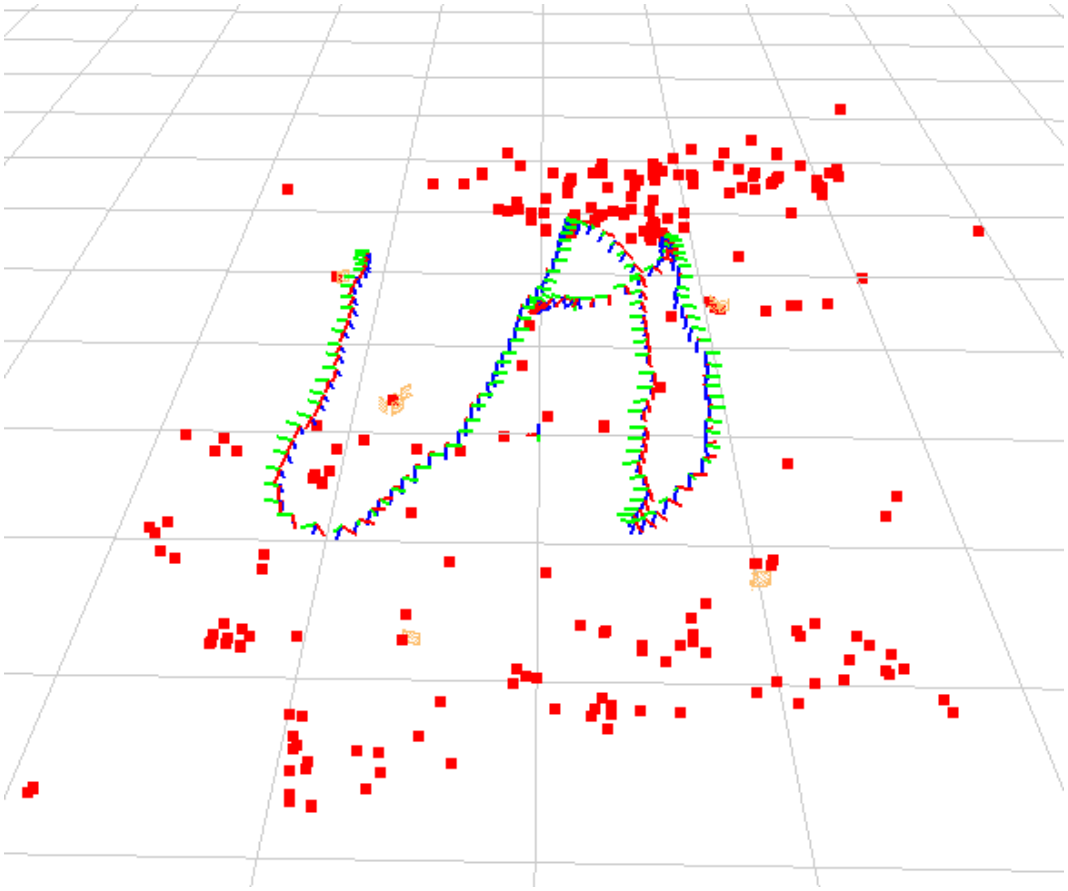


Figure 8-3: Three dimensional visualization of the second hand-held inertial and monocular trajectory segment from a SLAM solution, starting top left and walking three anticlockwise loops. Red-green-blue triads represent IMU orientation (camera is different). Red dots are opportunistic corner features in the room and orange squares are opportunistic AprilTags (used for loop closures). The visualization was generated using the MIT Collections Viewer package.

tracks, and once using feature tracks and the few AprilTags as common loop closure measurements. Each of the figures that follow may show 'Monocular' for features only, and 'Monocular + loops' for features and AprilTag measurements. We discuss the monocular and monocular with loop closure cases in the two separate sub-sections below.

Fig. 8-3 and Fig. 8-4 show three dimensional visualizations of the monocular with loop closure SLAM solution. The trajectory shown in Fig. 8-3 is shorter and

snakes back and forth, while the second in Fig. 8-4 walks around in multiple loops through the room. The red-green-blue triads show IMU poses position and orientation in the VICON room reference frame, and red points are opportunistically monocular corner features positions as estimated by the SLAM process.

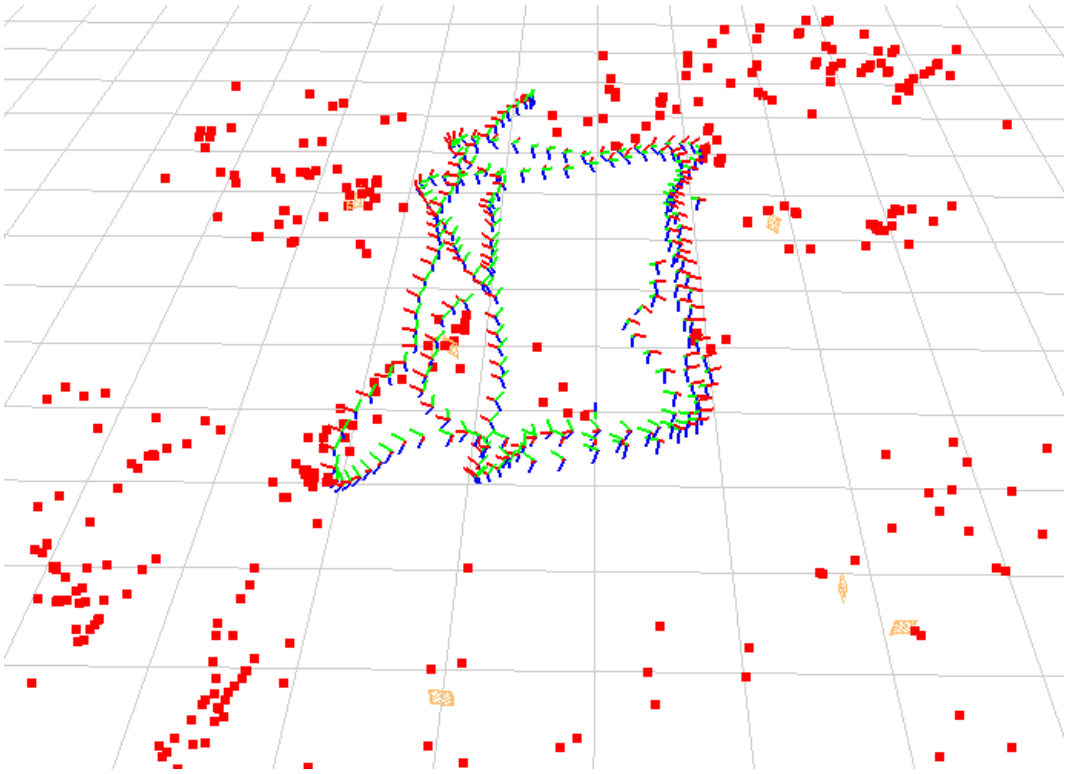


Figure 8-4: Three dimensional visualization of the first hand-held inertial and monocular trajectory segment from a SLAM solution, starting top left and walking three anticlockwise loops. The elements descriptions are the same as 8-3.

Sparse corner features only

In this experiment sparse corner feature sightings – from a monocular camera – are used to aid the inertial odometry bias estimates. Fig. 8-2 shows the factor graph as the combination of a Structure from Motion problem (monocular feature sightings) and inertial dead reckoning. The least squares solution to this combined system of equations is computed incrementally with iSAM [116] and trust region extension [197].

Accuracy of orientation estimates is a key indicator to inertial navigation performance. Errors in pitch and roll generally produce the dominant inertially driven velocity and position errors, due to gravity cross coupling. In this test, monocular feature sightings constrain the change in position between consecutive image frames, making pitch and roll of the platform observable through gravity, as measured by accelerometers in the inertial odometry factors.

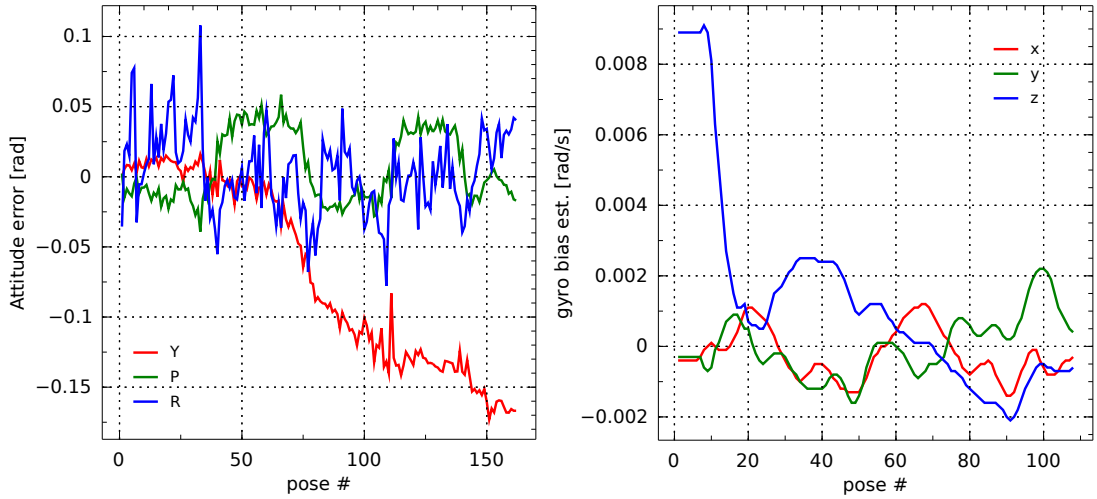


Figure 8-5: *Left*, residual yaw, pitch and roll orientation errors from VICON estimated orientation reference, for SLAM trajectory using inertial odometry and sparse monocular tracked corner features. Loop closures are not used for results in this figure. *Right*, gyroscope sensor triad bias estimates across entire trajectory, showing estimate at each pose.

Pitch and roll variations between VICON and SLAM solution are around $3\ deg$, which is pretty poor – although, evidence suggests this variation originates in the VICON orientation estimate. Empirically we observed fluctuations in the VICON orientation estimate that may be due to poor optical marker placement. The position error resulting from gravity cross coupling with $3\ deg$ orientation error over a 10 second interpose constraint would be around $9.81 \times \int\int_0^{10} (\sin 3^o)\, d\tau^2 \approx 25\ m$. Our trajectory runs longer and has far less positional error. We expect our pitch and roll estimate errors are well below $3\ deg$.

The red trace in Fig. 8-6 shows the absolute heading error of the inertial with monocular test case. We see the heading error continually drifts without bound. This heading drift is expected, since we have no measurements other than the initialization prior to align the poses to the global orientation. We also note that when

SLAM solutions are aligned to floor plans by hand, the absolute heading error is hidden; Fig. 8-6 shows the true heading error.

Sparse corner features and loop closures

We reuse the data from the previous experiment, but now process camera images to include five distinct AprilTag [179] marker cubes in the room. These markers act as unique visual landmarks to emulate loop closure constraints. AprilTags avoid difficulties in data association commonly encountered with opportunistic loop closures [133].

Fig. 8-5 shows that the orientation errors for the monocular sparse feature tracking and inertial odometry test case are similar to those presented in Fig. 8-6 for pitch and roll, but yaw error remains unconstrained without loop closures.

Loop closures constrain relative positions and orientations between two separated poses in the graph. Fig. 8-6 shows how global heading error angle is constrained and not left to grow unbounded. We do note that absolute heading is still not observable without incorporating external heading aiding information. In other words, the average yaw offset from an absolute reference is still only weakly constrained through the initial condition [68]. Furthermore, we note that the heading error has a significant impact on global position accuracy of poses. Fig. 8-7 shows the error in absolute position for the monocular only, and monocular with loop closure test cases.

Fig. 8-8 shows inertial sensor bias estimation consistency when aided by monocular camera. Although we do not know what the sensor biases were during the test, we can use strong VICON constraints to produce bias estimates to compare to the camera only aiding case. We note further that while it is difficult to explicitly determine errors in these plots, we can evaluate the integrated sensor measurements after compensation with Figs. 8-6 and 8-7. We can see from Fig. 8-8 that camera only aiding produces bias estimates that are comparable to a well constrained reference trajectory.

8.2.3 Computational load

Inertial odometry increases state dimension at each pose and constraint, but can reduce the number of poses required if good odometry constraint can be achieved – understanding the computational penalties relating to this increase in per pose state dimension is important.

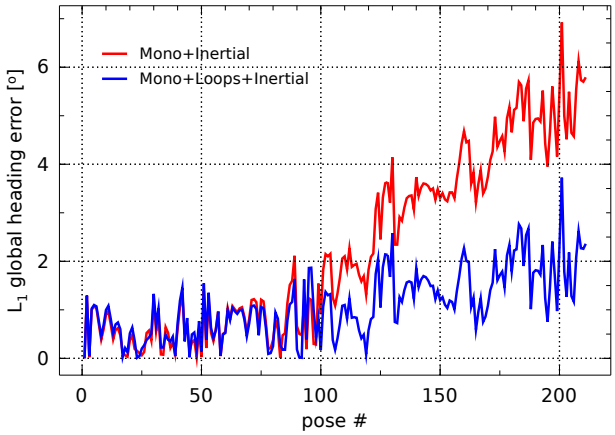


Figure 8-6: Absolute global heading errors for SLAM trajectory referenced against VICON (reference); respectively using inertial odometry with sparse monocular tracked corner features, and camera sighted *AprilTag* features as loop closures also.

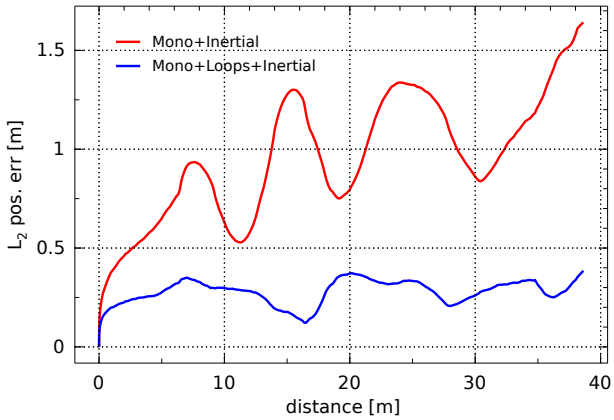


Figure 8-7: Absolute position error, $\|\mathbf{p}_i - \hat{\mathbf{p}}_i\|_2$, for each of the poses at the end of the trajectory. Red trace shows monocular and inertial odometry SLAM solution errors, while the blue trace includes AprilTag sightings as loop closures also. Global heading errors are inclusive to these errors.

Fig. 8-9 shows the per iteration computational time of inertial odometry constraints. Here we used the iSAM1.7 implementation to perform incremental smoothing inference.

This concludes our investigation of the hand-held visual-inertial experiments on a factor graph, and we progress to include kinematic constraints using the BDi

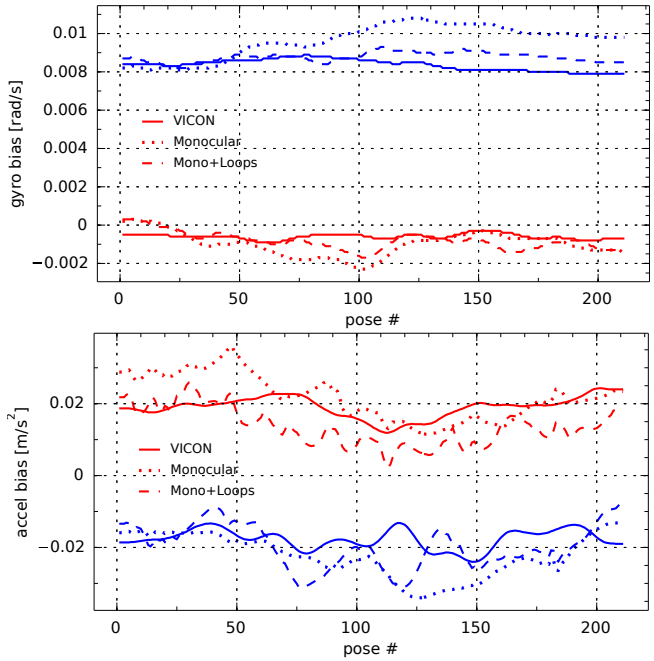


Figure 8-8: Gyro and accelerometer retroactive bias estimates during three repeat runs of hand-held monocular camera test. VICON traces represent a solution highly constrained to ground truth position measurements (3DOF), and is taken as a reference for sensor biases. Red traces are $x-axis$ and blue for $z-axis$, $y-axis$ is omitted for clarity.

Atlas humanoid robot.

8.3 BDi Atlas Humanoid Robot (Group I & II)

Work on the BDi Atlas humanoid robot was done in collaboration with the MIT DARPA Robotics Challenge team, using data and tools available from the competition. Our test here only uses inertial odometry, joint kinematics, and head mounted monocular camera feature sensing modalities. To the best of our knowledge, is the first factor graph based state estimation and mapping framework that includes kinematics, for a humanoid robot.

Humanoids require both a high bandwidth vehicle state estimate, as well as a long term stable and accurate solution [54, 223]. Existing state estimate solutions, including [5,24,55,199], use Kalman filtering to infer the state as a single snapshot

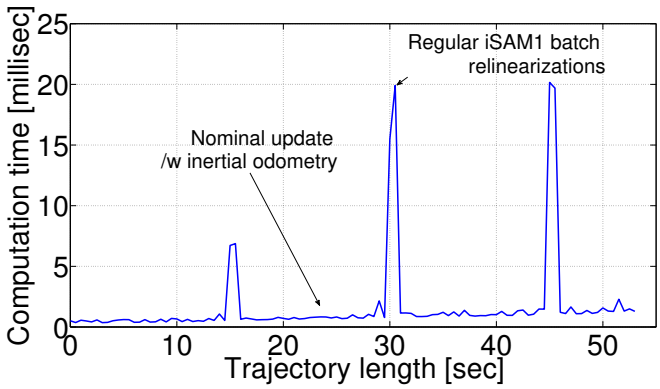


Figure 8-9: Incremental smoothing (per iteration) computation time with inertial odometry factors is on the order of milliseconds. Results are based on an iSAM1.7 implementation [116], which requires regular batch relinearization steps.

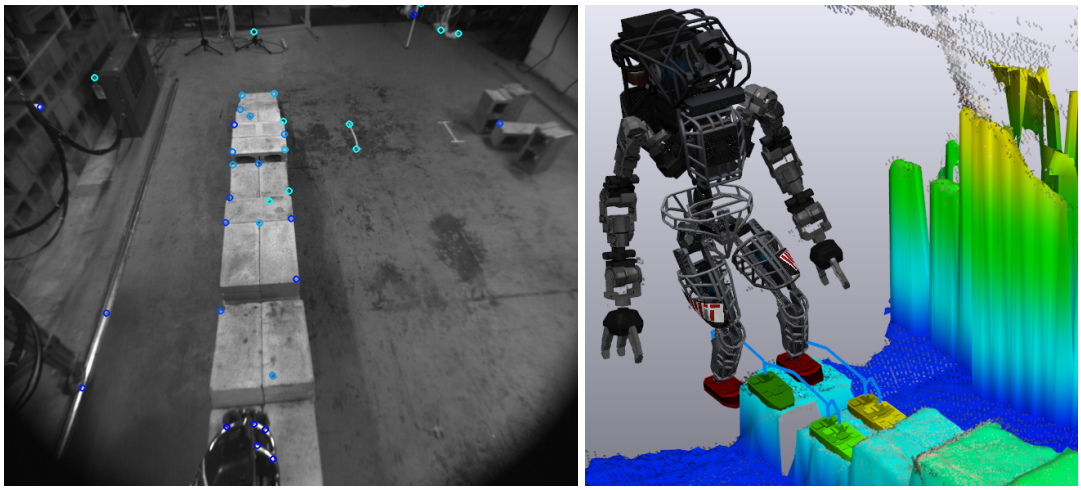


Figure 8-10: BDi Atlas stepping onto cinder blocks. Views are from real robot test data. *Left*: Monocular head camera image example where features are being tracked in the image sequence and and integrated into the factor graph as visual features. Feature sightings factors, alongside inertial odometry factors, and leg kinematics are integrated into the joint joit probability factor graph shown in Fig. 8-11. *Right*: Third person view of a synthetic environment from Team MIT's DARPA Robotics Challenge toolchain [54].

estimate.

We propose an approach whereby inertial, leg kinematics and camera data are

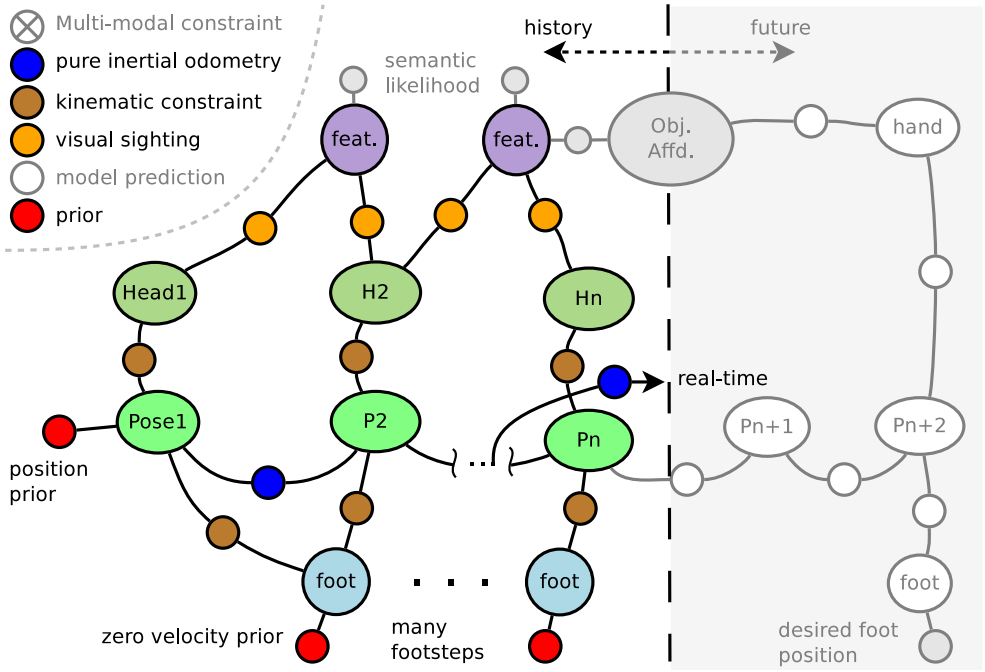


Figure 8-11: Joint probability factor graph representation for BDi Atlas humanoid kinematic robot walking over cinder blocks while observing visual features in the world, using factors from Group I and II.

fused with a factor graph approach. We exclude lidar, and rather construct a passive sensing localization and mapping solution which does not make any prior map assumptions. Further, this formulation simultaneously estimates the map and location of the robot and does not use a known map assumption. The purpose of our experiment and processing is to show that we are able to map and localize simultaneously, rather than in separate or pseudo-parallel steps.

A specially developed front-end process interprets IMU, joint angle, foot pressure, and head mounted monocular camera data, which is used to construct a joint probability factor graph, as shown in Fig. 8-11. The main advantage with a factor graph approach, rather than filtering, is that we are able to spread footstep localization in the world frame across an entire foot contact period, rather than a one-shot footstep placement as generally done in a filtering framework.

Fig. 8-12 shows the pelvis IMU trajectory solution from SLAM using inertial odometry for interpose relations every $0.5\ Hz$ with data originating from the IMU at $1\ kHz$. Joint encoder measurements, through leg kinematics, are attached to each pose and the associated footstep pose. The robot is walking over stacked

cinder-blocks, as shown in Fig. 8-10. Around 250 monocular features were included in the SLAM solution spanning around 20 robot footsteps.

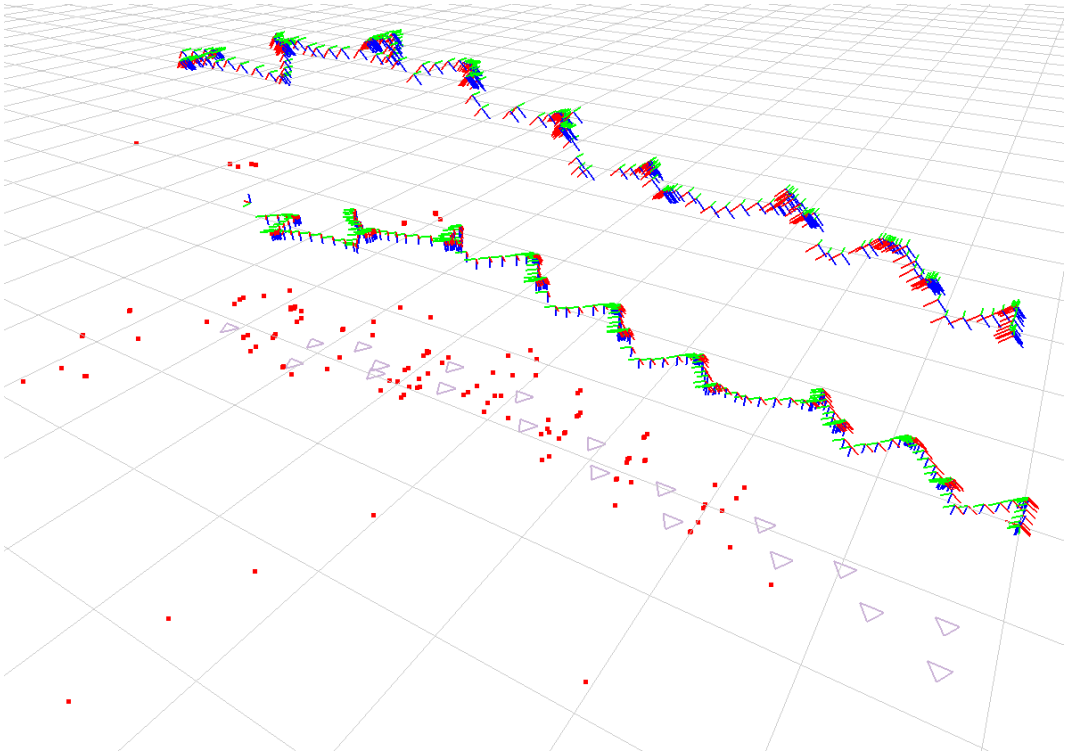


Figure 8-12: Visualization of pelvis IMU pose and footstep position/orientation (triangles) estimates of the BDi Atlas robot traversing cinder-blocks from SLAM solution. The top row of triads depict the head mounted camera pose used for monocular opportunistic corner feature sightings (as shown in Fig. 8-10). The factor graph for this solution is illustrated in Fig. 4-1. The incremental solution was computed with iSAM1.7 using the trust region method.

We compare the SLAM solution accuracy to the native *BDi* localization solution supplied with the robot in Fig. 8-13. We note that the SLAM solution here is using 500 times less kinematic measurements than is available, but inertial information is summarized without loss of information. We also note that the solution produced by [55] was used as the position reference solution for our experiments below, since we know its has good long term position accuracy (prior lidar map).

Fig. 8-13 shows the SLAM solution is capable of out performing the native IMU and kinematics state estimation solution supplied with the robot; we do note, how-

ever, that there are specific difficulties relating to the factor graph style SLAM solution. During testing we found the solution to be sensitive to local minima in the resulting objective cost function. We spent a fair amount of time in our implementation to improve the initialization of the trajectory variables before optimization was performed.

Poor initialization would result in trajectories near the BDi solution, sometimes a little worse, sometimes a little better. Initialization of opportunistic monocular features are also important, since poor initialization would tend to reduce the final SLAM solution accuracy. Even with the initialization peculiarities, we can show the SLAM style solution is readily able to produce accuracy results comparable to the industrialized BDi solution.

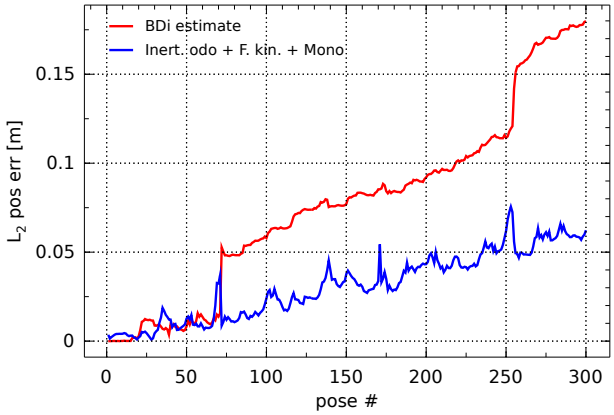


Figure 8-13: L_2 translation accuracy of trajectory poses estimated by inertial odometry, kinematics and monocular features in a SLAM solution, and comparative BDi state estimate solution.

Regarding the computational load of our humanoid factor graph based state estimate solution – the inertial odometry computations, along with incremental smoothing from iSAM are able to compute the SLAM solution in real-time on an Intel i7, 16Gb computer. Twenty robot steps with an interpose rate of $0.5\ Hz$ results in around 4000 new variables in the SLAM solution.

We now transition away from purely parametric solutions, having shown that inertial odometry factors can successfully be used in a factor graph context. The sections above noted the difficulties in getting the front-end process to correctly extract measurements with truly Gaussian error distribution, which could be added to a optimization based inference framework.

8.4 Wheeled Robots (Group I, II, & III)

In this section we present two different wheeled robot datasets. First is the well known Victoria Park outdoor wheel odometry and laser scanner, where tree like objects are detected and used as bearing range type features. The second dataset is from a Turtlebot [182] autonomously exploring the second floor of the Stata Center building, over three separate sessions, using wheel odometry and monocular sightings of several AprilTags [179] sparsely placed in the environment.

We indicate measurement likelihood factor groups I, II, & III are used, since we add familiar parametric and new nonparametric factors into our joint probability. Factor group II is also included due to wheel odometry, which we also use for real-time prediction of our Turtlebot robot in the latter example. We do not include inertial odometry factors in this example, but our approach does not exclude or prevent use of inertial factors.

All inference examples in the remainder of this chapter were performed by the *Multi-modal iSAM* algorithm, using the *Caesar.jl* implementation [65]. Both experiments are conducted in a two dimensional mechanization, and note the *Caesar.jl* implementation also offers three dimensional mechanizations. We note, that our solution allows nonparametric factors which the previous versions of iSAM are cannot accommodate. The experiments selected in this section are chosen to expand the discussion from the canonical multi-modal examples in Chapter 6.

8.4.1 Multi-modal Victoria Park

The Victoria Park dataset [168] consists of a car with a planar laser scanner and an automated tree trunk detection algorithm. The data association amongst tree-like object sightings in the laser scans are not known; and this makes the Victoria Park dataset a good ambiguous data association example to explore how multi-modal solutions can be used. Fig. 8-14 shows a single snapshot of the front-end navigation process tracking tree-like objects with a Bayes filter, using inverse odometry and laser scanner detections.

Raw data is received as wheel and steering angle encoder measurements, which we convert into odometry using a basic car model and integration process. The odometry is used as distance, turn angle, or time trigger for generating interpose rigid transform relations – conditional likelihoods such as $[\,\mathbf{Z}\,|\,\mathbf{X}_i, \mathbf{X}_{i+1}\,]$. We had to recalibrate the raw data with offset, scale, and extrinsic parameters, and used a Gibbs like stochastic decent method: Our method minimized the enclosed area of narrow portions in the top down view that would result from an odometry only

trajectory.

Our objective is to recover the vehicle trajectory and local map of the detected tree-like objects, while demonstrating how multi-modal factors can be used to overcome data association difficulties. Our approach is to develop a front-end process to interpret the sensor data shown in Fig. 8-14 and assemble a joint probability factor graph, as illustrated in Fig. 8-15. To do this, we must first illustrate how the multi-modal factors are extracted from sensor measurements. The process is illustrated in Fig. 8-16.

After studying the raw data, we also realize that the environment is dynamic, with several objects moving around in the scene. As the car returns to a previously visited area, we get repeated measurements from the same objects in the scene, but also notice significant variations between existing landmarks and new feature measurements.

Figure 8-14: Single snapshot view of front-end processing for the Victoria Park outdoor dataset, showing laser scanner detections and tracked features. Tracking of objects is performed with inverse odometry and a Bayesian filter — using Sudderth & Ihler et al.'s multi-scale Gibbs sampling strategy on kernel density estimates [218] — to assign non-overlapping feature IDs while detections are in the field of view. The vehicle is traveling from left to right.

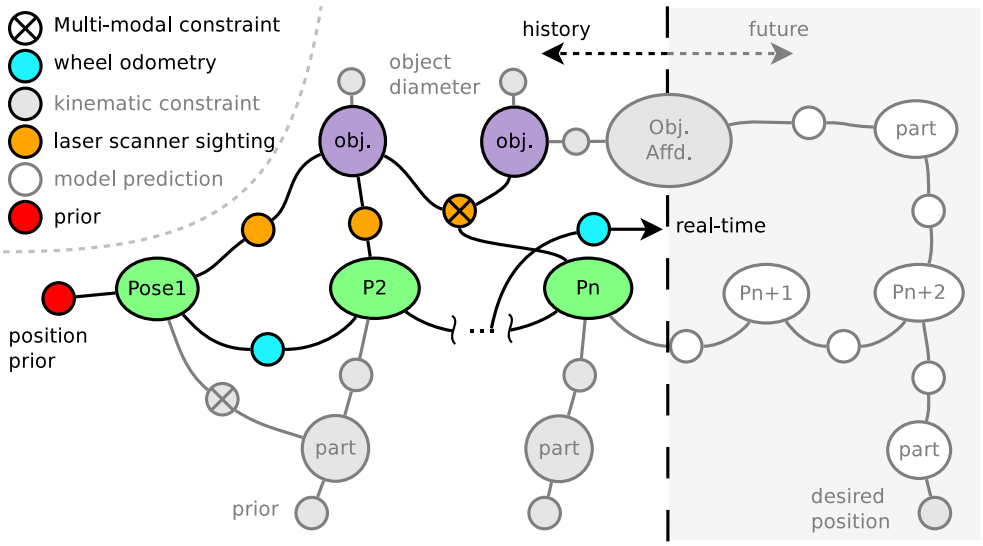


Figure 8-15: Joint probability factor graph representation for Victoria Park car driving dataset, with unceration loop closures and using factors from Group I, II, and III.

An earlier approach used by Kaess [111], used the Joint Compatibility, Branch and Bound (JCBB), Neira et al. [169], technique to find a reasonable data association solution. Results reported for the iSAM1 algorithm in [111] uses pre-process JCBB data association while the SLAM solution is under human supervision, and the reported timing results excludes this data association pre-processing step.

Repeated visits to an area usually observe various subsets of the tree-like objects in that area: sometimes new features are seen and other times less features are seen. These factors imply that direct pattern matching between the estimated map and a new snapshot measurement is not always obvious (even to a human observer). Automated matching methods such as JCBB sometimes incorrectly associates nearby objects to just one previous object or drops the correct measurement association in favor of a new object in the scene. These characteristics require parameter tuning for the JCBB algorithm as part of the front-end, back-end interactions.

We purposefully pursue the opposite, a ”thin” front-end type process which inserts a fair number of multi-modal constraints — each time a measurement associations is ambiguous — to illustrate the effectiveness and computational feasibility of our method. In the latter full dataset test, we also randomly corrupt varying fractions of the proposed loop closures, inserting randomized associations into the

graph to show the innate robustness of a *sum-product* style solution.

We stress that the *Multi-modal iSAM* approach does not preclude good front-end design – our intention here is to show the usefulness of relaxing the classic Gaussian-only loop closures as limited by current parametric methods. Our approach allows us to deal with data association uncertainty (such as loop closure proposals) in a novel way. To the best of our knowledge, the approach illustrated in Fig. 8-15 differs significantly from previous robust SLAM approaches.

For example, recent work in nonparametric SLAM by Mu et al. [160] employs an iterative clustering and solve technique. A Dirichlet based clustering technique searches for measurements that are common to the same object in the map space. Each new factor is solved after which the process returns to the clustering step. In our approach, we only build the factor graph once, implicitly defining the non-parametric variables within the likelihood beliefs.

We are instead interested in an approach where the user may associate measurements with multiple feature objects, as discussed with multi-modal measurement likelihood factors in Section 3.4.3. Feature matching strategies, such as JCBB, are generally *max-product* style techniques – taking only the best assumed match associations. An alternative *sum-product* approach would rather pass the ambiguity in the selection along as a likelihood distribution of possible associations. The *sum-product* approach then allows each measurement to be associated with more than one object, and specifically point to eq. (3.23) for the bi-modal association model we used for the Victoria Park dataset.

Fig. 8-16 illustrates uncertain data association for loop closures in two conceptual situations, where the robot pose belief (shown in blue) produces likelihood regions for landmarks via a single bearing range measurement (pink regions). The predicted landmark locations may or may not overlap with existing landmark position beliefs, as shown. These two situations both produce ambiguous associations and are described by eq. (3.23). The hyper parameter ρ can be selected uniquely for each situation. The multi-modal approach does not discard any measurement information as null-hypothesis.

A bespoke front-end converts the odometry and laser scanned detected objects into a factors in the factor graph. The front-end produces the dead-reckoning predicted trajectory and map initialization as shown in Fig. 8-17. Note this figure represents an intermediate point in the robot trajectory. The front-end will associate uncertain measurements (loop closure proposals) to objects with one of two most likely objects in the scene, thereby deferring loop closure data association to consensus in the *sum-product* inference solution. More than two options could be included.

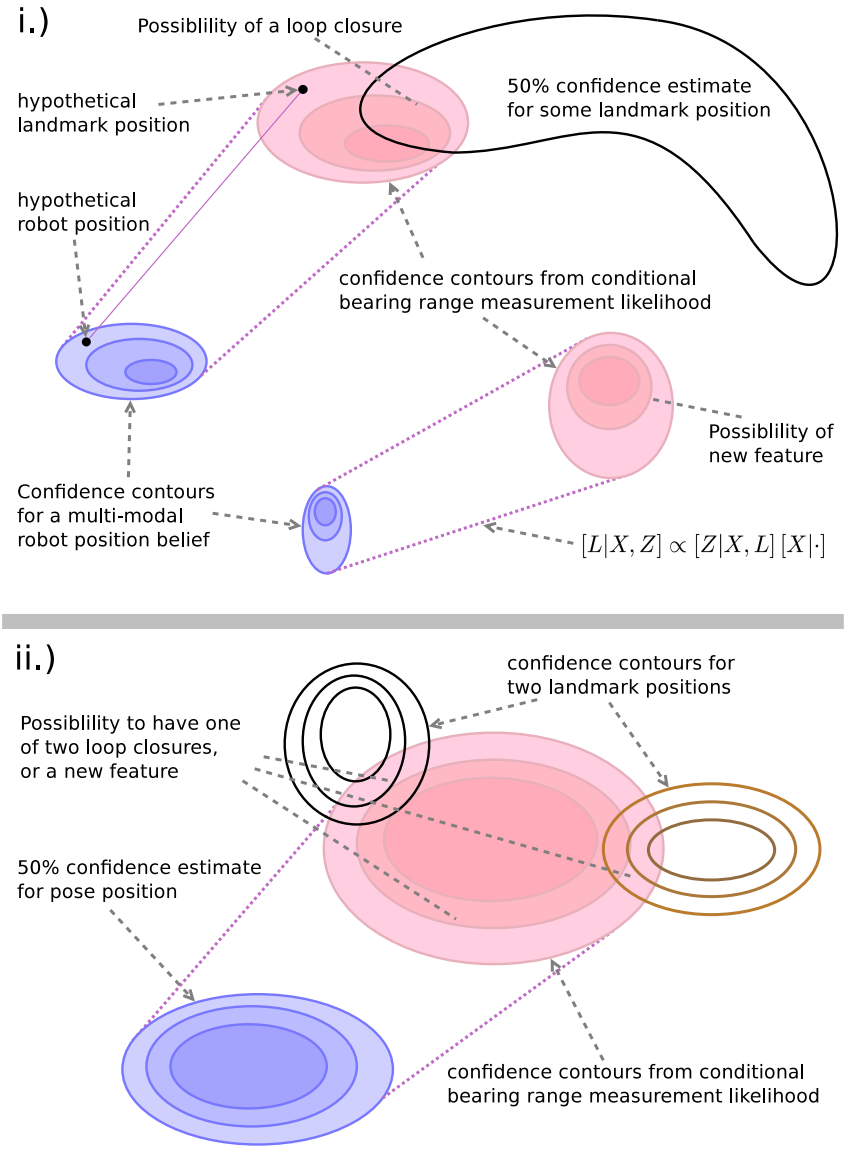


Figure 8-16: Two illustrations of how loop closure ambiguity can result from a single bearing range measurement to some feature in the world. THe belief over the pose position $[\,\mathbf{X}\,|\,\cdot\,]$ gets convolved with the bearing-range measurement (conditional) likelihood to an actual feature observation $[\,\mathbf{Z}\,|\,\mathbf{X},\mathbf{L}\,]$ (several shown in Fig. 8-14), resulting in the red likelihood region $[\,\mathbf{L}\,|\,\mathbf{Z},\mathbf{X}\,]$. Top illustration shows multi-modal pose belief $[\,\mathbf{X}\,|\,\cdot\,]$, bottom shows likelihood intersection with two existing landmarks $[\,\mathbf{L}_i\,|\,\cdot\,]$.

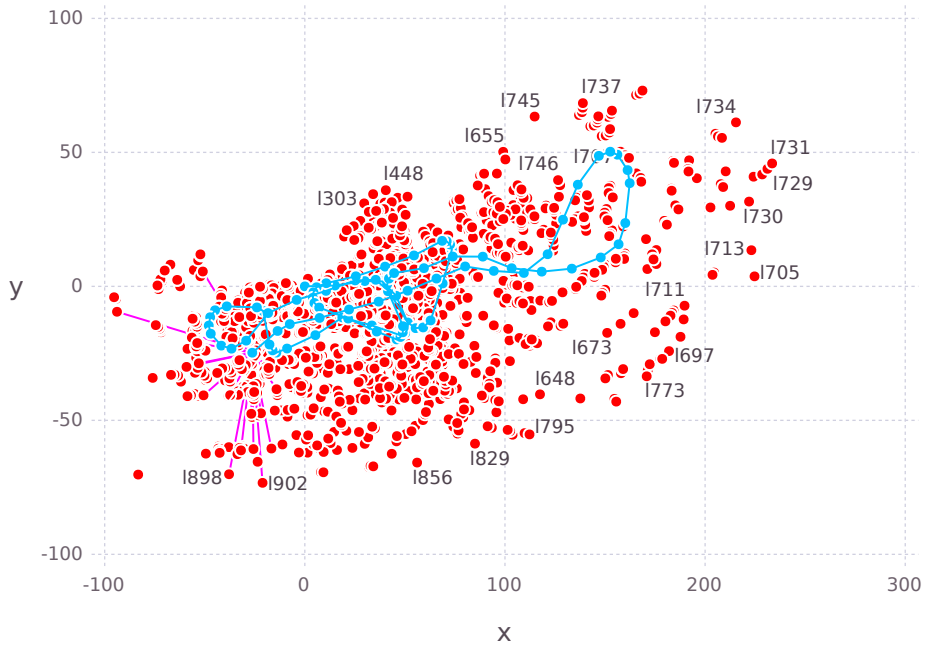


Figure 8-17: Intermediate factor graph pre-inference setup of Victoria Park outdoor car based factor graph model, where cyan lines represent the estimated trajectory and red dots represent laser scanner detections of tree-like objects. Magenta lines represent laser scanner measurements from the latest pose.

In order to get ground truth data association, we work through all the laser sighting data by hand — using the front-end process to speed the work flow — to obtain the most likely associations. During this process we could see the subtle variations and errors made by the JCBB algorithm. After carefully working through all the data and using the multi-modal inference, we empirically believe our hand picked ground truth data association to be about 99% correct. Again, there are dynamic objects in the scene and feature detection is not perfectly consistent.

When consuming the Victoria Park data with our front-end, we preload the ground truth associations and ensure that uncertain loop closures have a $50\%/50\%$ bi-modal factor where one of the landmarks is in fact the correct association. The other feature is a likely candidate or new feature. We also note that new features are only created for the duration of a feature track. Furthermore, we avoid the situation where each measurement is allowed a new feature, which would have resulted in too much freedom – measurement information could be pushed into

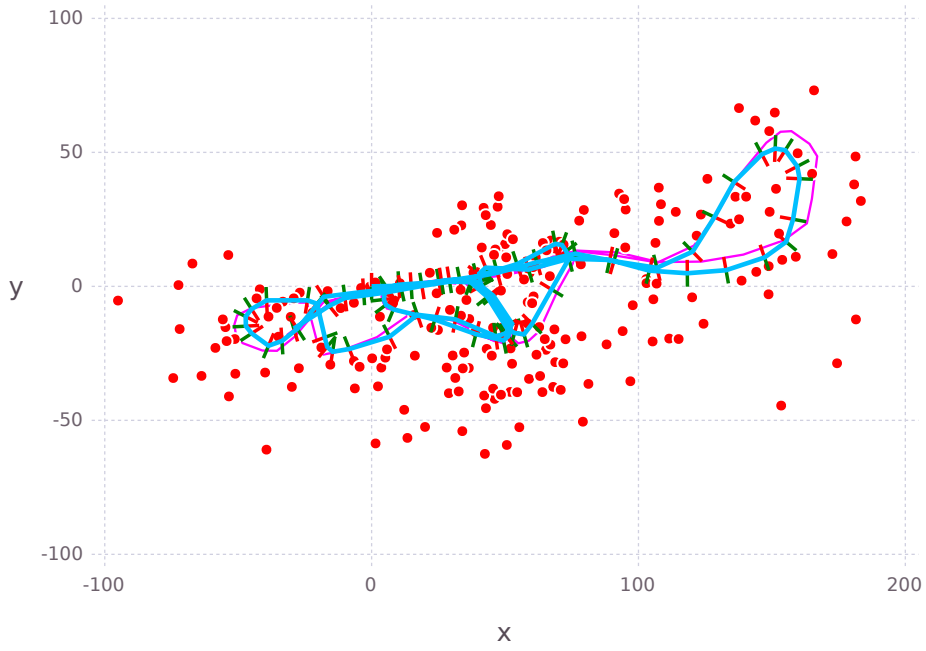


Figure 8-18: Intermediate result of Victoria Park outdoor car dataset, where red and green lines indicate pose orientation as left and right sides of the vehicle, respectively. The magenta line shows the equivalent parametric, *max-product* result using our 99% confidence ground truth loop closure data associations.

each of the free variables.

After constructing the factor graph, with information as illustrated in Fig. 8-17, we let the *Multi-modal iSAM* solution consume and infer all the marginal posterior beliefs of the joint distribution. Fig. 8-18 shows inference result by choosing the maximum point from the marginal posterior belief of each variable (primary mode) – secondary modes are not visualized. Fig. 8-18 clearly shows that the bi-modal associations are considered and loop closures are being included, given the improved state estimate accuracy, and a reduction in scattering of tree-like objects in the map, in comparison to Fig. 8-17.

We proceed to process all 23 $mins$ of driving data, resulting in a factor graph with approximately 3800 variable dimensions and more than 8500 constraint dimensions. There are approximately 1700 are bi-modal likelihood factors (uncertain loop closures), and contains 482 additional landmarks variables. The theoretic number of modes in the system is very large (more than 2^{400}). Our focus is to estimate the much smaller number of prominent modes. In fact, there are theoret-

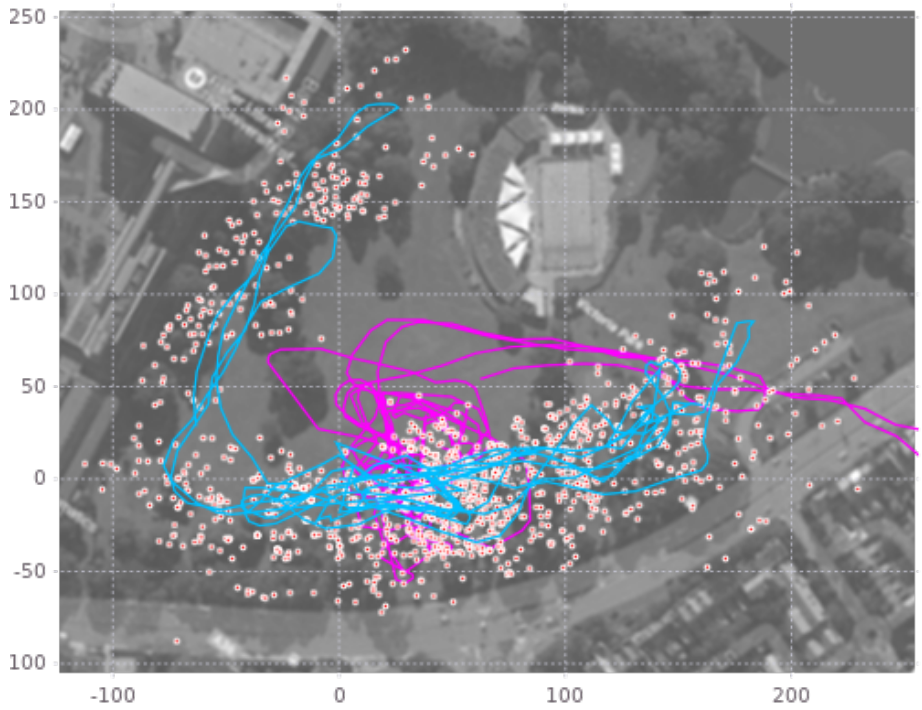


Figure 8-19: The blue trace shows Victoria park $argmax\ p(\Theta|\mathbf{Z})$ point estimate, with 10% erroneous loop closure proposals. The magenta trace shows the same result for naive maximum likelihood estimate. Alternative modes, which were calculated, are difficult to visualize and not shown here. This figure was used in [62].

ically near 2^{1700} different permutations (modes) which are modeled in this Victoria park solution.

While on the subject of problem size: recovering all marginal beliefs for this size problem takes around 3.5 hours on a dual Xeon, $64\ Gb$ RAM computer, utilizing around $5-6$ processing cores on average. Our implementation has room for several significant implementation speed-ups. We expect a factor of up to ten speedup is possible. This would involve better multi-threaded exploitation of convolution operations, and better in-place memory operations.

The blue trace in Fig. 8-20 represents the maximum a posteriori point estimate extracted from variable beliefs after multi-modal posterior estimation has completed (MM-MAP). The magenta trace represents an equivalent maximum likelihood estimate (MLE, iSAM1) with all loop closures as classical unimodal con-

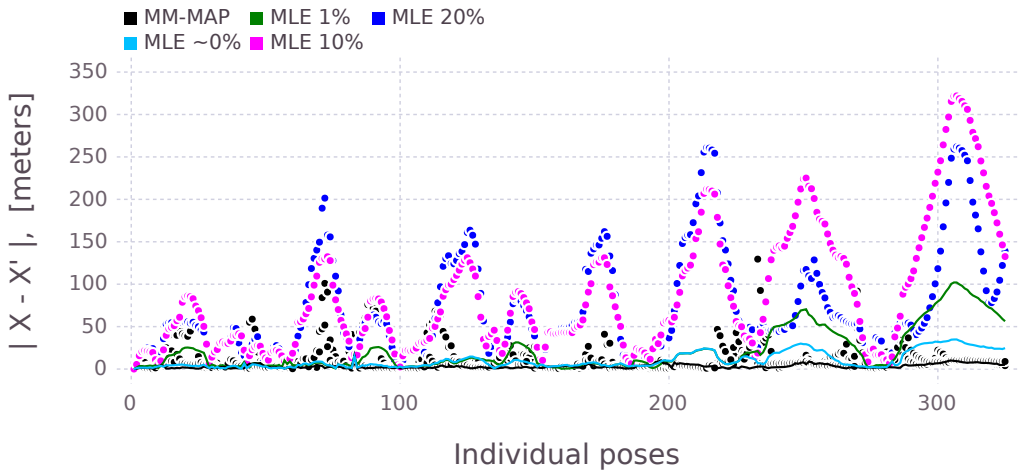


Figure 8-20: Distance between similar poses, using MM-MAP 0% estimate as the baseline, with varying levels of loop closure proposal corruption: 1%, 10% and 20% corruption. The large values correspond to an erroneous unimodal MLE equivalent. Notice the black trace and dots representing equivalent MM-MAP estimates (for 1%, 10% and 20% corruption cases) which have much smaller errors. In this example, corrupted data implies less loop closures are in effect, reducing overall accuracy.

straints. Fig. 8-20 shows how MAP estimates remain consistent in the presence of significant false proposals. A few poses have an alternate mode that is more prominent. Further inference passes over the tree, or addition of new data will most likely emphasize the correct mode as most likely.

8.4.2 Multi-session Turtlebot

In Chapter 1, Section 1.3, we discussed how a navigation solution should be able to span various timescales. In this section we wish to illustrate the power of combining technology developments, such as graph databases, with our centralized factor graph approach and *Multi-modal iSAM* inference solution. The work and results in this section were conducted in collaboration with S. Claassens, S. Pillai, and R. Mata; and in part previously published in [64].

In the previous sections, we showed how to develop a high bandwidth navigation system using inertial odometry factors for an IMU and visual feature sightings from a Camera, or kinematic relations of a robot through ground contact footsteps.

The Victoria Park experiment showed how wheel odometry could be combined with many bi-modal factors, while deferring data association ambiguities to the multi-modal inference process.

In this experiment, we will be using two types of nonparametric measurement factors (Group III, see Section 3.4.3): The first is multi-modal association factors, as used in the Victoria Park experiment above. These factors are used when monocular camera sightings of AprilTags [179] are ambiguous or seemed to have moved. This allows us to deal with dynamics in the scene. The second nonparametric factor is for multi-session priors, where we combine the location estimates of landmarks common to other sessions into a multi-modal belief for each local session. Taking care not including your belief of the same session into the prior.

The experiment is intended to show that any of these navigation systems can be centralized around a server system, using a transactional navigation system, to enable in-situ interaction between the human operator and many different autonomous software elements. Our contention is that only by centralizing around the factor graph with a SLAM-aware client-server architecture will be able to build system that achieve the required throughput and systems integration required for building a artificial *location and dynamic awareness*.

As we consider longer timescales of in-situ and real-time operations, we are faced with the problem of combining high bandwidth odometry type measurements with large volumes of existing data. We also show that an autonomous navigation capability needs to be repeatable across robot operating sessions, while aggregating data collected and allowing third party interactions to leverage the quantity of data into a value-adding product.

In the next wheeled robot example, we look at the familiar Turtlebot robot [182], using wheel odometry and monocular AprilTag sightings to resolve the robot position in an indoor environment over various timescales. Three robot sessions were conducted over two days, while some of the AprilTags were moved during and between sessions. We will use the robustness of *Multi-modal iSAM* to overcome the issue of moving tags.

The three robot datasets were combined (in-situ) using the centralized framework discussed in Chapter 7. We use the centralized system to resolve a robot state estimate and map solution, and aggregate the relevant information from three sessions using nonparametric multi-session alignment, using the framework illustrated and discussion at Fig. 7-1. Lastly, we extract value from having larger volumes of data available in a common reference frame, showing how the centralized SLAM-aware system can benefit robot systems even while the robot is not in use.

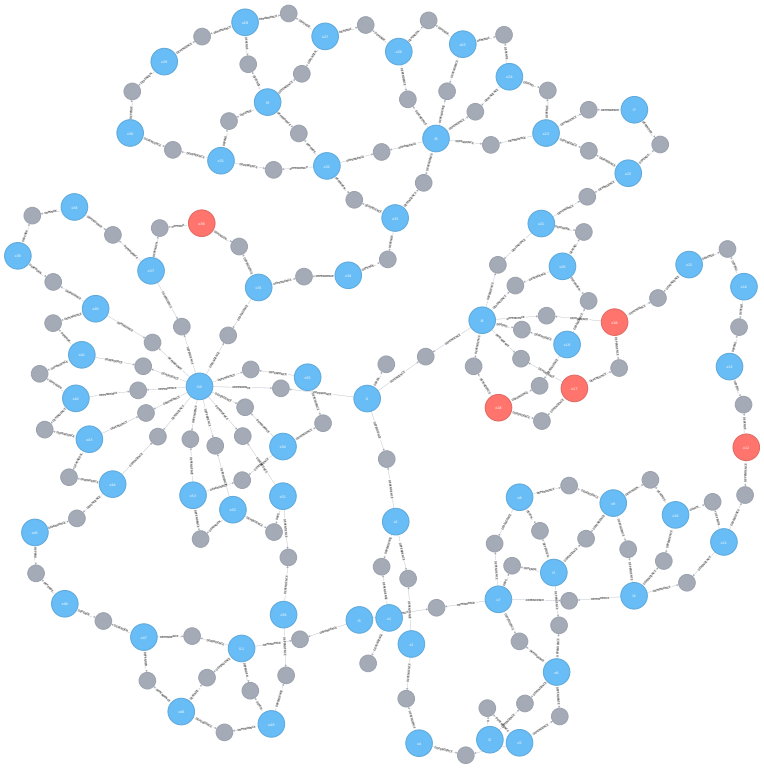


Figure 8-21: Snippet illustration of a centralized factor graph data structure (using Neo4j [4]), where red nodes indicate poses selected from a foveation query in short trajectory segment. Landmarks and other poses are indicated in blue, and connected by constraint gray factors. Notice how poses with common camera views, in red, are not necessarily direct neighbors in the graph.

In the previous Victoria Park example, only wheel encoder measurements and laser scanner detections of tree-like objects were available – resulting in fairly small data volumes. In this experiment we use a Turtlebot robot with integrated wheel odometry and a RGB-D camera, and AprilTags which are sparsely placed in the environment. the AprilTags are purposefully moved around between and during robot sessions.

In this experiment we also show how common landmarks from different sessions are combined together in the common centralized framework. During these experiments, we used a separate server computer to which the Turtlebot robot would connect during each of the operating sessions. The significance of our approach is that the robot itself never computes a SLAM solution, but is instead in-

terfaced with the more powerful server computer.

Furthermore, our approach allows us to share information between three inference solutions on the server, rather than construct just one large optimization, greatly improving the optimization time. We will show how a near limitless number of smaller processes can access and operate on data, dramatically simplifying the development and prototyping process. This approach allows the user to develop individual purpose applications rather than one or two large processes, as is classically the case in most SLAM systems.

Our approach uses wheel odometry for state prediction and dead-reckoning to a real-time estimate, using the AprilTags for loop closures. Throughout this experiment, the *Multi-modal iSAM* solution is able to resolve a new complete batch solution in the range of tens to hundreds of seconds, well within the operational period of the robot. The inference procedure could be used on all three sessions in parallel, since our centralized approach is not limited to the operation of just one robot at a time.

Assembling the Factor Graph (Robot Front-end)

Instead of trying to process all the information on the robot computer, we use the robot as a "thin" client to perform a few basic periphery tasks, including odometry tracking & pose triggering, monocular AprilTag detection, RGB-D image blob store, and factor graph modification operations. Turtlebot sensor measurements are converted into measurement likelihood factors which are added to the joint probability factor graph as described in Fig. 3-6 while the robot is traversing the local environment. Modifications to the factor graph are on the order or three or four factors every few seconds.

We extract a real-time solution by periodically recovering the inference result of the most recent pose state — as discussed with Fig. 7-2 — and projecting the remaining time difference from fast wheel odometry up to a real-time state estimate. We describe the combination of Group II type factors where slower multi-sensor inference meets real-time state estimation in Section 3.4.2, but note in this particular experiment we use wheel odometry, as an equivalent replacement for inertial odometry used in the earlier experiments.

Low bandwidth factor graph updates are uploaded through a wireless network to the server computer, using the starved graph database framework outlined in Chapter 7. The server is then able to incorporate the new information into the current SLAM solution, while new data is continuously being uploaded from the operational robot.

Fig. 8-21 shows a short snippet of the factor graph maintained and visualized by the graph database system (Neo4j [4]), where variables in blue are connected by various factors in gray and; while red nodes are the result of a foveation query which we shall discuss below. We also discuss the SLAM solution as part of the third party interactions section below.

Replicas of larger data blobs are then uploaded from the robot to the server at lower priority, and are indexed according the unique identifier (UUID) which was created at the instant the data blob was first uploaded into the system. This approach allows a starving strategy where the graph database only stores the UUID for later retrieval from a separate blob store. Large data blobs on the robot which have already been uploaded to the server and are no longer required for immediate operations, can be deleted from the robot blob store to preserve hard-drive space. In this case, we upload RGB-D keyframe images and depth clouds as independent data chunks, which are identified by their UUID. For completeness, we also store data from an onboard laser scanner, but is not used in this experiment.

Illustrating Third Party Interactions (Everything else)

The database acts as a central focus point for all the data, simplifying third party interactions. By attaching a separate process to the server, such as image segmentation, user visualization, or SLAM inference, we can process, add, or modify values independently. We can also add more large data blobs to any node in the starved graph database system. We emphasize that one or multiple robots can constantly be streaming data back and forth from the server. The database also decouples the concurrent processes, by ensuring all transactions are atomic.

Dedicated, larger computational tasks may very well be placed near or on the same server computers, and the SLAM solution is one key example. We use multimodal iSAM to query, infer, and insert current best Maximum-a-Posteriori estimates back into respective vertices of the database. The estimates provided by the SLAM service therefore become available for global consumption via the database. Next to the SLAM inference process working away at the database system, we can add another process to do image segmentation.

A centralized graph database system can help us develop *dreaming* capabilities by handling the low-level data indexing and searching. In this experiment, we focus on the idea of foveation, as shown in Fig. 7-8, where the SLAM inferred positions of the robot and landmarks are used as a value-add type of meta information. Having geometric information about poses and landmarks enables efficient server-side spatial searches over potentially large volumes of sensory data.

Figure 8-22: Foveation, see Section 7.4.2, Fig. 7-8, and Pixel-wise segmentation, using SegNet [9]: Three keyframe SLAM-aware images recovered through a *foveate* query. Bottom right shows a repeat image but with segmentation color palette mask overlay. The images show a common scene separated by hours from actual events, and shows how an independent process may now manipulate the persistent database centralized factor graph to introduce new loop closure constraints based on object detections of the painting. The figures also shows a missed painting detection in the top images, which can be used to improve training of the detection and classification algorithms. This figure was modified from [64].

We use the camera image information to illustrate how third party interactions can be facilitated with the centralized SLAM-aware database system (SLAMinDB). There is usually more information available from raw camera image data, but this requires large computational resources and more advanced algorithms.

We use SegNet per pixel image segmentation algorithm from Badrinarayanan et al. [9] on all keyframe images, one per pose of the robot. We also insert the segmentation output image into the data blob store, with its own UUID unique identifier, and co-register the new UUID with the pose in the central starved graph database.

From there, yet another process can visit the updated nodes and factors and see the robot state estimate as well as the image segmentation information. Fig. 8-22 shows four example images from pose keyframes across several multi-session robot trajectories. The images are masked with pixel level segmentation output, showing the 'artwork' label, and was retrieved from the associated data blobs we had discussed above, using the standard SegNet image segmentation.

Extracting Value from Aggregated Data

Visualization is another process, as depicted earlier in Fig. 7-1, which attaches to the database and all relevant multi-session information and gives the user the benefit of aggregated data. Pose state, SLAM solver values of interest, depth point cloud, RGB & segmentation color palette for each pose can then be visualized as required. Again, this process is greatly simplified by the ability to query over all the information and only retrieve (transport) the bits relevant to that specific user query.

In Fig. 8-23, we see pose-slaved point clouds from three Turtlebot trajectories. These multi-session maps contain internal loop closures, and thanks to the common factor graph persistence, can also share information among sessions. By using sightings of AprilTags and current best estimates of robot keyframe pose locations, we can manually introduce new relationships in the graph to effect cross-session loop closures.

The visualizations in Fig. 8-23 shows three robot trajectory sessions. The right-hand side segment shows one session's point cloud with a segmentation color palette for floors, walls, and objects. We can now use the common data persistence to enhance training on the segmentation system, leveraging the SLAM-aware information available in the database.

Now, consider searching for more loop closures to improve the quality of the map or localization, or consider seeking to improve an object detection and classification system. We can use the SLAM-aware system to extract value from larger amounts of data.

Using yet another independent process, we can search for the painting shown both in the keyframe images, Fig. 8-22. The painting is also visible in the point-cloud reprojection shown on the right hand side of Fig. 8-23. Using the foveate query—illustrated with Fig. 7-8 and described in Section 7.4.2—we can search for robot poses on the database which might see a specific point in space. After recovering a list of poses from the starved graph database, we can retrieve the keyframe images and segmentation data for each from the large blob store.

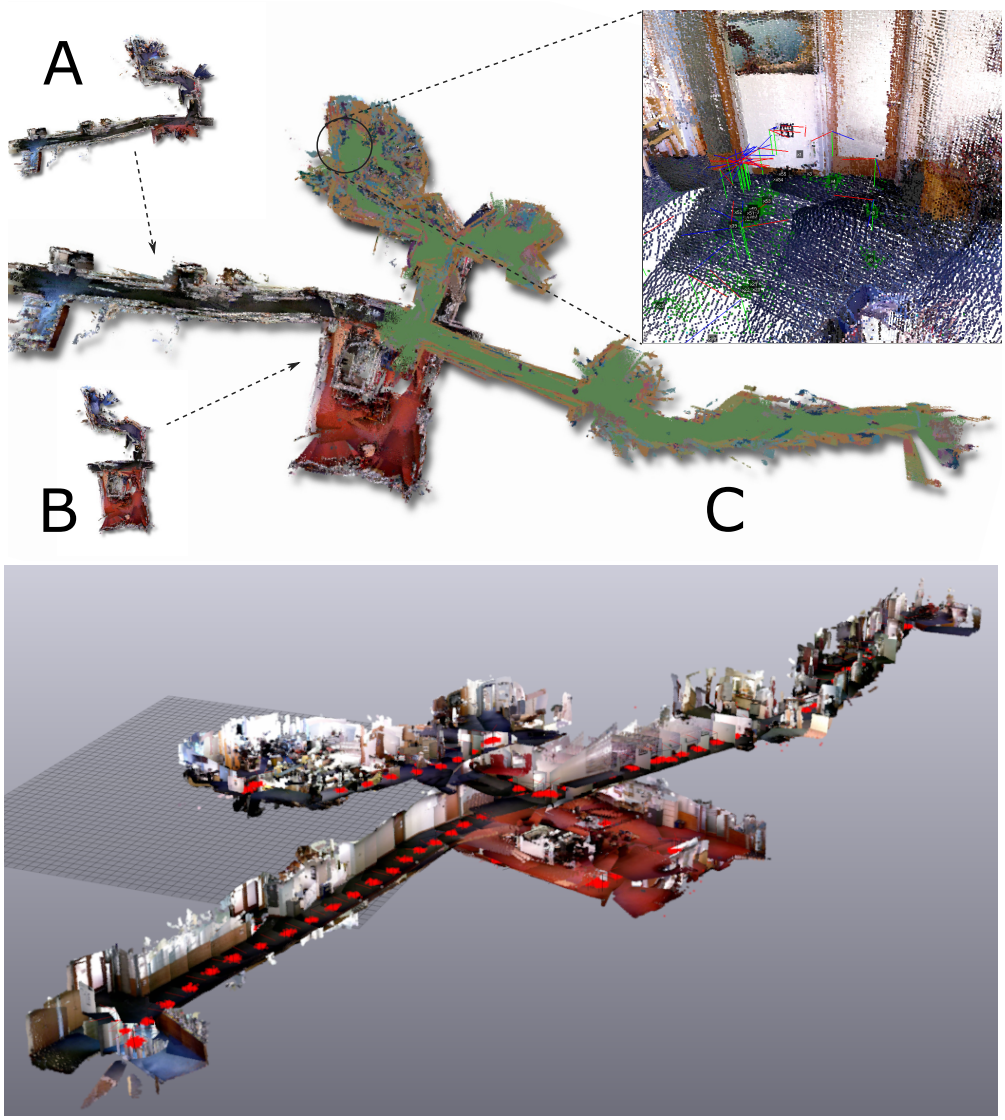


Figure 8-23: Composite 3D reconstruction from a ”decoupled” visualization process (as depicted in Fig. 7-1) from three Turtlebot trajectory sessions. Individual keyframe structured-light point clouds are projected from multi-modal iSAM inferred trajectories directly in the centralized graph database. The session to the right has color labels given by SegNet segmentation [9], showing floors, walls, objects and more. Red dots on the bottom figure are samples from the marginal pose position beliefs. Top figure re-used from [64].

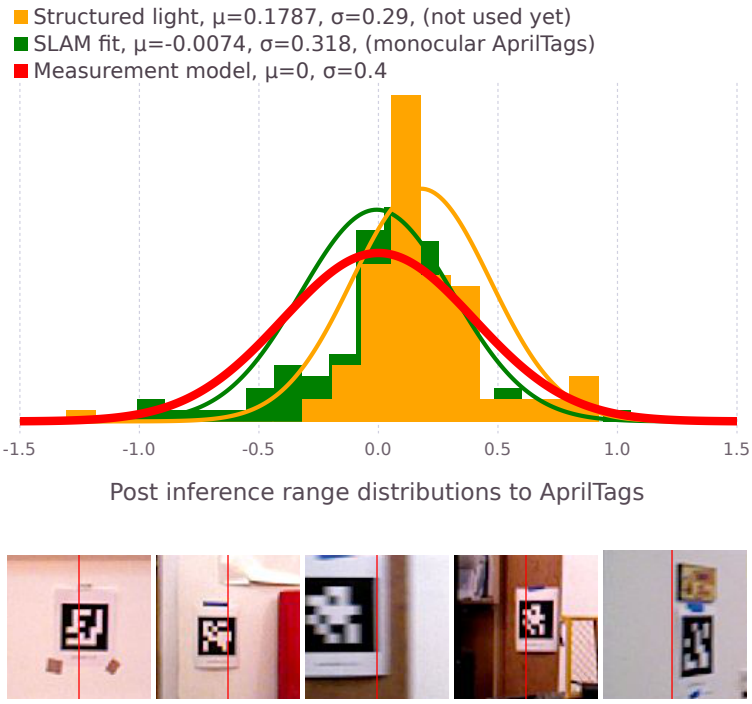


Figure 8-24: Histogram distribution of range measurements from AprilTag sightings, using data from multiple sessions, tied together by the multi-session interface discussed in Section 3.4.3.

The centralized architecture for multiple robots and sessions allows leveraging large amounts of data through dedicated learning operations. In the last part of this experiment, we briefly show the value in understanding and improving calibration errors in a robot directly from data. For this example, we compare the range values from i) monocular AprilTag detections, ii) post inference SLAM pose to landmark distance, and iii) structured light data (which is not yet used as a measurement in the current SLAM solution).

The three Turtlebot trajectories shown in Fig. 8-23 contain camera and structure light depth sightings to AprilTags across the three sessions. Fig. 8-24 shows data from all AprilTag sightings, drawing histograms of the residual between AprilTag detection range and SLAM or structured light distances. The assumed measurement noise model is also shown in red. Fitting a normal distribution to each of these residuals shows the SLAM solution to be consistent with zero mean and conservative covariance. The structure light data has roughly the same covariance, but

a mean offset of $0.18\ m$ indicates a clear calibration error that can now be rectified.

8.5 Fully Nonparametric Likelihoods (Group I & III)

We progress to the last experiment in this chapter, using nonparametric measurement likelihood factors from acoustic beam forming data from an autonomous underwater vehicle. We show how the *Multi-modal iSAM* solver can directly operate with nonparametric Group III factors, see Section 3.4.3. This experiment was not conducted on the vehicle, but was post-processed using the centralized architecture described in Chapter 7.

8.5.1 Underwater Acoustic Navigation (Beam Forming)

Work in this section was done in collaboration with Nick Rypkema et al., using data from the inverse Ultra-short Baseline (iUSBL) acoustic system discussed here [163]. Note that our results here are not representative of their system performance. We use purposefully degraded data — by reducing the resolution of data used — to demonstrate how highly non-Gaussian measurement likelihoods can be used directly in a nonparametric factor graph framework. We refer the reader to Schmidt, Rypkema, & Fischell's work [163] for a real-time filtering based solution, and post filtering parametric SLAM solution.

With this experiment, we introduce the ability of replacing the separate particle filtering and parametric optimization steps, used for navigation in [163], into a single nonparametric factor graph based inference framework. THe example shows is that a *sum-product* style inference solution is possible for larger problems, and demonstrate this by constructing measurement likelihood factors with data from as near the analog-to-digital acquisition as possible. Although we were not able to recover the raw analog-to-digital sampling data from the iUSBL system, we were able to retrieve slightly preprocessed version.

Beam Forming for Nonparametric Likelihoods

The iUSBL system is a vehicle mounted system that samples four acoustic transducers placed in a tightly packed tetrahedron. Data from the four sampling tracks are processed with a beam forming mechanization, see Van Trees [233], to recover an intensity map along range, bearing, and elevation dimensions. The beam forming mechanization searches for a correlation between received signals and a template waveform stored on-board the vehicle.

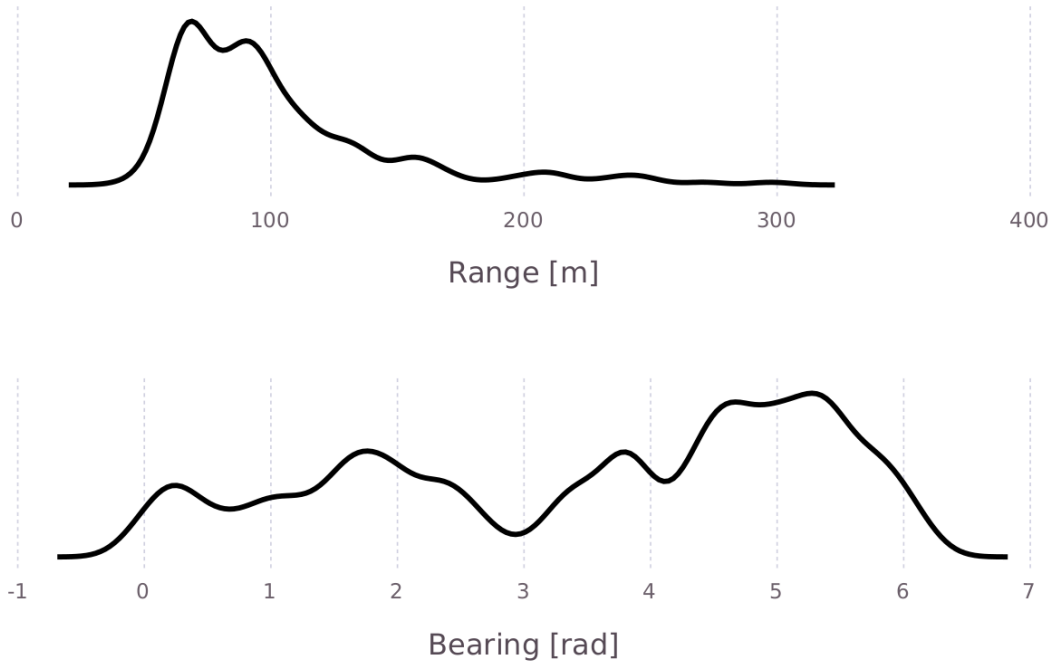


Figure 8-25: A sub-sampled and smoothed range and bearing estimate from a beam forming acoustic system, excluding convolution with an acoustic channel model.

By correlating (convolving) a template waveform—given sender and receiver clock synchronization—, we can estimate the time-of-flight range to the pinger source. Additionally, even without sender receiver timing synchronization (hyperbolic navigation), the time difference between when acoustic wavefronts are detected by each transducer can be used to extract the bearing and elevation of where an acoustic transmission originated. Strictly speaking, in USBL systems the time delay between different transducers is within one wavelength, and therefore we prefer to use phase angle of arrival – also considering convolutions are computed as products in the frequency domain. By convolving the range, bearing, elevation intensity map with an assumed acoustic channel model, we arrive at a probability likelihood of where a time synchronized acoustic ping originated.

The top plot in Fig. 8-25 shows the combined correlator outputs of all four transducers against the time-of-flight as range. Again, this plot is not representative of the iUSBL performance, as the data has been sub-sampled and partially processed for plotting. The range measurement scale is recovered by multiplying the in water sound speed with the time-of-flight delay. The lower plot shows the bearing

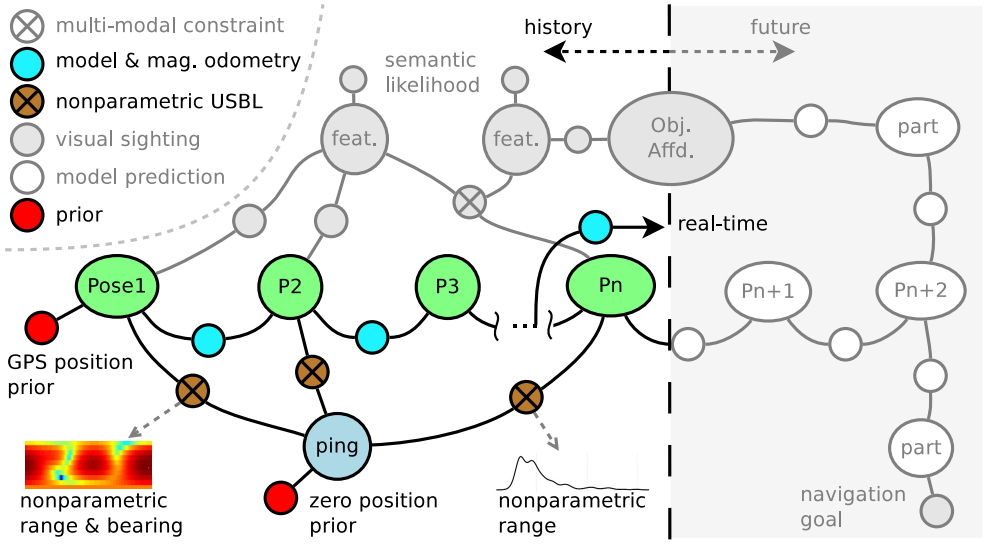


Figure 8-26: Joint probability factor graph representation for underwater nonparametric USBL measurements, using factors from Group I and III.

angle, relative to the vehicle body, where the ping originated. The bearing angle shown includes body roll and elevation which has been marginalized into the one dimensional bearing estimate.

One important point on the bearing correlator output shown in Fig. 8-25 is that range information has also been collapsed into the bearing estimate. Unlike the marginalization of roll, pitch, and elevation data into the bearing estimate, only the maximum range likelihood information is used to assemble the bearing likelihood. This maximum range point selection is the only part of the measurement pipeline which does not constitute an end-to-end *sum-product* approach. We do have access to the combined raw range data for deep likelihood representation there.

Assembling the Nonparametric Factor Graph

We assemble the nonparametric factor graph, as illustrated by Fig. 8-26, using an odometry and iUSBL measurements. Odometry is assembled from a combination of heading and propeller turn count based distance. Heading is extracted form an off the shelf MEMS based, low-cost magnetometer and gravity (accelerometer) aided attitude heading reference (AHRS), using gyroscopes to predict platform orientation. We conduct this experiment in a horizontal, two dimensional mechanization.

A front-end process was developed to upload odometry and acoustic range and bearing information to centralized server architecture discussed earlier. Odometry is used to extract unimodal parametric rigid transforms between poses. Poses are time triggered each second to coincide with iUSBL measurements which are also expected each second. Acoustic measurements are all considered nonparametric, using kernel density estimation to represent the likelihood belief over range and bearing, when available, and then let the *Multi-modal iSAM* solver find marginal posterior beliefs over all the variables. We add one landmark position for the pinger, with a $(0, 0)$ normally distributed prior. All (accepted) acoustic measurements from each pose are connected via their nonparametric measurement likelihood to the one pinger landmark node in the graph.

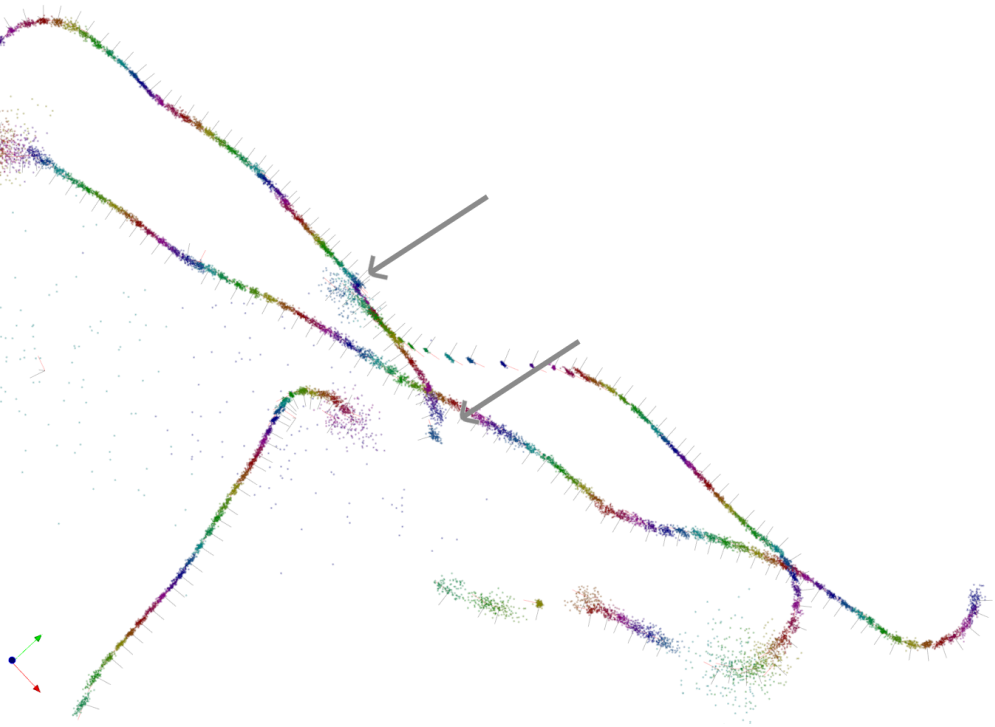


Figure 8-27: Trajectory estimate from beam formed acoustic data from a test in the Charles river, Boston, MA. This trajectory segment spans about $70\ m$ from East to West. Top right corner is towards North, as shown by green arrow in the triad.

Not all acoustic measurements are valid, and multiple USBL measurements are discarded using mean and covariance based consistency checks. Several partial USBL measurements are used, where range information is relatively visible,

but bearing measurements are nearly uniform and discarded as a result. We note that the previously discussed maximum range point selection for constructing the bearing measurement complicates its use during poor signal to noise measurements. In future, we would like to avoid this maximum selection step and let the combined inference search for the best posterior likelihood in combination with all other navigational data.

Fig. 8-27 shows the top down trajectory estimate from the joint probability factor graph described above. This trajectory segment starts at the bottom right, with true north along the green reference arrow diagonally torwards the top right of the image. The vehicle travels back and forth, west and then east, finally turning south at the bottom left of the image. Each pose is represented as consecutive color of dots, denoting samples from each pose position's marginal distribution.

In Fig. 8-27, we see some dispersion in the pose position estimates at the most westerly turn, but not sufficient to completely loose lock on vehicle position. Later in the trajectory, there is a loss of lock on the navigation solution for a couple of poses during the last westward travel section, and see the missing pose samples are widely dispersed to the lower left of the image. The navigation lock is then reacquired as a jump to the center of the image, as shown by the two arrows. Navigation lock is then lost again, before the solution returns to the correct point just before the last southward turn.

The loss of navigation lock is inherent to the data, where the odometry measurements are no longer able to constrain the solution into a continuous stream of poses, being over powered by the USBL bearing and range measurements. We note that this problem can be overcome by relying more heavily on the relative odometry chain and increasing the general uncertainty of the bearing and range measurements to each of the affected poses.

The key take-away we would like to emphasize with this experiment is, to the best of our knowledge, this is the first batch nonparametric navigation solution directly from sensor likelihood models using the nonparametric kernel density representation. The batch solution shown in Fig. 8-27 is computed by the SLAM-aware database system, where a full solution iteration takes on the order of $15\ minutes$ to compute. The computation time varies, based on the fidelity used to represent the nonparametric measurement likelihoods, and number of poses used to represent the trajectory.

We also note that the computation time of the algorithm is adversely affected by all acoustic measurements being connected to just one pinger landmark node in the factor graph. We can improve the sparsity by duplicating several landmarks with a duplicate $(0, 0)$ position prior, thereby reducing the clique dimension containing

the pinger landmark in the Bayes tree. We have not yet investigated that avenue further. The computation time of the algorithm is also influenced by the amount of history we consider. By operating in increment mode, we can reduce the cliques being updated, assuming multiple landmark nodes are used.

The next major option, and likely avenue — if we were to deploy the current solution — is to use a fixed lag version of the algorithm. By marginalizing out older variables and only keeping a fixed window of variables in the active portion of the factor graph, we can limit the size of the inference while still being able to extract the value of nonparametric smoothing over multiple variables.

In this experiment, we are able to show that parametric odometry can be combined with nonparametric factors in the common factor graph and that representative marginal posterior beliefs can be recovered from each of the variables in the system. This experiment shows an entry point to a whole new class of multi-sensor data fusion methods, where highly uncertain measurements, in very raw form, can be combined in a nonparametric framework and common consensus be found by the inference algorithms. This concludes the experiments reported in this chapter.

8.6 Critical Analysis

This chapter looked at several experiments that show how different combinations of parametric and nonparametric measurement likelihoods could be combined through a factor graph. We showed how the multi-modal iSAM algorithm is able to recover both consensus and alternative solutions given the available measurement data.

The hand-held visual-inertial experiment shows reliable sensor bias recovery and free trajectory estimation. These tests give us confidence that the inertial odometry factors are usable and add value to the overall joint probability factor graph based method for state estimation. Furthermore, we were able to show an improvement by including loop closures – a process that is significantly more involved in a hidden Markov type system, such as a Kalman filter.

We note that during development of the visual-inertial system, many failure modes were encountered and a large amount of 'parameter tuning' was required to make the system function. Difficulties with engineering a front-end solution—that were required to generate Gaussian only constraints for parametric inference with parametric iSAM1.7—provided the motivation for finding more robust inference, which resulted in the multi-modal iSAM work.

In our multi-modal inference approach, where data association happens at

inference time, the structure of previous factors in the factor graph remain unchanged. By contrast, null hypothesis methods modify the graph structure in preprocessing, or introduce penalty slack variables to disable the factors. Maintaining the topological structure of the factor graph allows us to exploit the full structure within the joint probability using the Bayes tree, which in turn maintains the incremental inference capability discussed in Chapter 5.

The results from multi-modal experiments in more challenging data association situations, as demonstrated by the Victoria Park dataset, or nonparametric measurements as demonstrated by the acoustic beam forming example, show that nonparametric factor graph inference is possible and computationally feasible. The theory and development at this stage point to major advances on the horizon for multi-sensor navigation and general automated *location awareness*.

8.7 Conclusion

Once we have a robust solution in place, as offered by multi-modal inference, we can work to build more automated methods that parse large volumes of data and improve the overall system performance. We can use a SLAM-aware database system to help find common features in the world, and feed back more loop closures or remove obviously bad ones – this post-hoc refinement is only possible if we can trust that a stable and robust inference solution is independently working away at the current best joint probability description offered by the centralized factor graph. We note this abstraction stretches beyond just a single platform navigation session, into multi-robot, multi-session situations.

In turn, information learned from an independent process, such as in Fig. 8-24, can be used to refine calibration or reconstruction from the data. Separation of concerns through the centralized graph architecture simplifies the tractability, concurrency and implementation of such learning processes.

Chapter 9

Discussion & Conclusion

Chapter 1 shows four cutting edge robotic platforms, and notes that they are not using a factor graph based state estimation (localization) and mapping process. Instead, these platforms use Kalman filtering approaches in a more rigid operational regime. Trajectories and behaviors are either preemptively known or skilled support teams are required to operate the vehicle.

We argue that a factor graph [129] based navigation solution should replace Kalman filter style solutions as the primary navigation source; Multi-modal factor graphs and nonparametric (multi-modal) inference techniques can provide the required robustness and flexibility which hamper Gaussian only SLAM systems that exist today. Existing parametric simultaneous localization and mapping solutions place an onerous restriction on the front-end process designer, who is limited by having to fit all measurements into a Gaussian uncertainty model.

To allow robotic systems more operational freedom for ad-hoc style interaction, one key requirement is the ability to deploy a more thorough *location and dynamic awareness* capability that is robust to unpredictable measurements and situations. We are motivated by the philosophy of *perfect five* autonomy, coined by Mindell [151], which is defined as the sweet spot in human/machine interaction: between low-level actuator control as a 'one' and some hypothetical high level unsupervised task level autonomy as a 'ten'.

9.1 Core Contributions

This thesis identified an open problem in robotic navigation relating to how an autonomous navigation system, that spans wide timescales, is need for a robotic future. We broke the down into high-bandwidth aspects and introduced inertial

odometry factors. Next, we developed the multi-modal iSAM algorithm for robust, non-Gaussian inference. This algorithm is the first sum-product style inference solution using the Bayes tree. The algorithm allows the front-end navigation system designer much greater freedom for representing non-Gaussian measurement uncertainties with more powerful probability models.

Our theoretical development relied on Markov Chain Monte Carlo theory to ensure that the inference would converge towards and stabilize around some stationary posterior. We require the user to supply measurement likelihood models which are reversible and only depend on variable states contained within the factor graph. We also require that samples can be generated from the noise models used for each likelihood. This approach allows for non-Gaussian inference over any combination of likelihoods that meet the requirements.

Through a series of canonical examples and real-world experiments, we were able to demonstrate a variety of navigation solutions that simply are not possible in other existing SLAM inference systems. Each of the examples could potentially be cast (in some special way) in existing SLAM solutions to compute a result, but our approach allows a generic, modular, and unified approach to all the examples. Our approach leveraged known factor graph and Bayes tree idea, but harnessed newer nonparametric inference components in a novel combination. This combination allows inference over problems having around 10000 variables in a sparsely connected factor graph.

In each of the example problems, we were able to show non-Gaussian, multi-modal posterior density distributions as the output result from the inference process. The key idea is that uncertainty is propagated into a static factor graph model and consensus is found by the multi-modal iSAM inference procedure, where dominant modes are tracked with minimal approximation error. The algorithm allows more or less particles per marginal to improve or speed up estimation of marginal posterior beliefs.

As a last piece, this thesis briefly looked at combining different aspects of a navigation system through a centralized factor graph approach. We used a starved graph data base and large blob store approach to separate concerns between each of the process. In hindsight, we promote the idea of a having an ecosystem of navigation related processes which are all working around a transactional navigation system model.

The remainder of this chapter discusses problem validation (rather than verfication) aspects, and tries to identify the bigger picture of where navigation research is going. We also try and understand whether the projected future development aligns with the needs and expectations of society.

9.2 Hindsight Requirements for Robotic Navigation

In hindsight, we revist the key characteristics required by robotic navigation system:

1. Predict a real-time state estimate with relatively low error;

2. Separation of concerns, by defining a framework in which different aspects of the navigation solution can be developed, tested, and integrated into a larger system as a collection of individual processes who each perform a dedicated and conceptually manageable task;

3. Flexibility in the manner which measurements are fused together, by allowing the system to maximally exploit a variety of opportunistic information gathered by the robot;

4. Modularity in the theoretical underpinning and available implementations to assemble different navigation systems for different robots using the same tools;

5. Dynamic calibration of critical sensor measurement errors, such our retroactive calibration strategy for gyroscopes and accelerometer biases;

6. Using a blend of sensing modalities with a broad synergistic reach in terms of long term stability and high bandwidth sensing;

7. We require a strong theoretical foundation to predict how a particular system will perform using a general toolkit software and sensor components;

8. Combine measurements over a wide timescales, starting from highly dynamic motion in the millisecond range through loop closures across multiple robot sessions over the span of hours, days or more;

9. Avoid algorithms that will later require a large amount of workarounds to deal with a unexpected real-world events – rather maximize flexibility first and then reduce the algorithm down to a particular application;

10. A degree of automation and fault tolerance in interpreting sensor data will always be required, and not all uncertainty can just be passed to the back end inference system;

11. We must forgo the conclusion that robots can only operate when predictable measurements are available (such as GPS) and instead be able to rely more heavily on opportunistic information where the robot is operating;

12. An inference mechanism to fuse sensor data together into a consensus estimate that is more robust and accurate than the sum of individual parts;

13. Existing hidden Markov methods (such as Kalman Filtering) have been extensively explored, but are not expected to provide the modular, flexible, efficient, or robust data fusion approaches. New methods are required and deeper research into factor graph approaches is most likely to yield the best results;

14. Concurrency (transactions) between different navigation processes are inevitable, as the complexity of the navigation systems increase, and we should rather think about building ecosystems of processes that deliver a navigation and state estimate solution;

15. Navigation is a means to an end, allowing third parties to utilize an automated system to achieve their goals; and

16. We also identify that navigation systems operate from within a community of people, of which the population is always in flux, and it is worth thinking about how designers and developers interact with the system during the research and development process – it should be easy for a new person to start interacting with an operational robotic navigation system and be able to quickly start developing and testing new features or fixing problems. We identify this as one of the separation-of-concerns, where training an productivity of new developers are considered a vital part of a navigation system's survival.

These characteristics pose both a scientific and technological challenge on the navigation system designer. We can work to solve these challenges through improved theoretical abstraction and better use of available technology.

9.3 Vital Aspects of Existing Solutions

We follow existing literature from the community in using factor graphs as a common framework for modern navigation systems – to describe how information

from multiple sensors are combine into a joint probability density estimate. State-of-the-art understanding of the navigation problem among the SLAM community, suggests that factor graphs are used as a language with which a bespoke front-end process can describe and transfer the inherent inference problem to a third party back-end solver. This separation between different robots and algorithms is well established and accepted by the community, we feel that is the best abstraction going forward.

The iSAM2 algorithm from Kaess and Dellaert et al. [115] is the current state-of-the-art SLAM back-end solver and a special case of the more general belief propagation algorithm discussed in Chapter 5. iSAM2 is an implementation of a *max-product* style inference technique, also discussed in Chapter 5. The combination of *max-product* style and Gaussian factors leads to sparse, non-linear least squares approaches, Chapter 2. We discuss the limitations of the Gaussian and *max-product* assumptions in the next section.

Critically, the iSAM2 algorithm introduces the Bayes tree symbolic refactorization of the factor graph. Although the iSAM2 algorithm assumes all factors to be Gaussian, both the factor graph and the Bayes tree are more general representations, which go well beyond Gaussian distributions. In addition to the flexibility and parallelism gained from the Bayes tree representation, it also enables recycling of previous computations through an incremental update scheme. Going forward, we expect to see incremental inference on the Bayes tree to become a common fixture in future SLAM techniques.

An objective assessment of SLAM suggest that many applications do not necessarily require a high-speed SLAM solution, but does require a robust map building capability to guide the long term state estimation process required for real-time operation. Concurrent smoothing and filtering work by Kaess et al. [117] makes a deeper assessment on distributing large SLAM solutions, and finds an elegant analytical method to separate larger (and older) measurements from a smaller (and younger) problem by means of the Bayes tree and sharing of a common root clique. Their work show a viable method to separate low latency state estimation from the large multi-sensor data fusion. This line of reasoning allows us to investigate slower, but much more versatile data management strategies, such as a database-centric approach.

Some work in the SLAM community has been breaking away from the Gaussian only measurement assumption. The histogram style Bayesian inference approaches, such as in Thrun et al. [224], approximate solutions in a finite and discretized fixed world grid. The penalty for a simplified discretized world is that the navigation system must keep all individual elemental beliefs about the state

current, requiring large amounts of processing, memory, and systems challenges in finding where computation should be focused.

Literature outside robotics have also studied different inference techniques in great detail. We discussed many of these methods, and contribute by bringing *sum-product* style thinking into state estimation and robotic navigation community.

The statistics community have long developed several tree algorithms—analogues to the Bayes tree—for re-factoring any factor graph into a more computationally tractable form. Clique trees, cluster trees, rake-and-compress trees and most notably the junction tree, all re-factorize a "loopy" factor graph into an acyclic representation [127]. Most of the existing work consider trees much smaller than typically found in SLAM problems.

9.4 What Needs to Change

At the highest level, development of modern navigation systems for ad-hoc operations in GPS-denied environments—where a user can easily guide the system in unpredictable ways—requires a robust navigation solution that is able to cope with all measurement and processing uncertainties; while simultaneously delivering a real-time and position, velocity, and orientation estimate.

Current workhorse Kalman filter based inertial navigation and GPS (INS/GPS) systems have the luxury of a confined operating regime and predictable measurement behavior. The operating regime assumption effectively dictates measurement certainty, such as valid GPS reception, is required before any vehicle operations can start – this is a common requirement with current INS/GPS systems.

In order to relax the operational regime to a more ad-hoc approach, we need to rethink manner in which navigation state estimation is performed. To some extent, we must forgo the predictable measurement requirement and be able to rely more heavily on opportunistic features and references in the world. Increasing the variety of sensing modalities a navigation system can robustly incorporate, significantly increases the robotic usefulness of such a system.

Similarly, deep sea remotely operated vehicles (ROVs) have various inertial and acoustic Doppler navigation aides (Kalman filter style INS/DVL) but still rely on highly skilled pilots, vast amounts of energy, and a high-bandwidth connection to the mother ship. As impressive as ROV's are, for example work done during the Deepwater Horizon oil rig disaster, ROVs still require full ships and crews from which to operate. Even for trivial inspection tasks still have steep operating costs and safety risks. Again, these requirements on the operating regime limit the scope

and feasibility of more general robotics.

Using the Deepwater Horizon disaster as a further example, it took almost three months to get the deep sea oil spill under control. Not because of a limit in the human operator's skill or the ROVs strength and capacity, but rather in the complexity of getting the ROV machine to perform tasks the users and pilots intended.

As a sort of symptomatic argument, we can see the changes required to ROV operations as a proxy into what has to change in the navigation system solution: ROV pilots are highly skilled and trained, but getting the machine to do the required task is a challenge. We argue that the current ROV piloting operations are cutting edge, but do not yet operate through a system that achieves *perfect five* autonomy.

Towards achieving this goal, there is a progressive logic trap in the development of navigation type multi-sensor data fusion systems. Chapter 2 discusses at length how INS/GPS and visual-inertial filtering systems progressed from initially being loosely coupled comparators of position; but over time various challenges and subsequent development resulted in tighter centralization of the prediction and measurement models.

While the progression of tighter coupling and better centralization of measurement models seem obvious enough, the way in which the system comes together is extremely important. The tighter coupling has a negative impact on separation-of-concerns. The further trap is to avoid the lure of a Gaussian-only assumption for computational convenience. In this these, we introduce both a centralized navigation framework that separates concerns, tied together with high-bandwidth inertial odometry state prediction and *multi-modal iSAM* robust inference.

We argue that the logic should be centralized around factor graph data structure, and all other processes are separated from there. Chapter 7 discusses the value of using a client-server model, built around graph database technologies, where a new "thin" robot front-end process[1] does not pull in the back-end inference code, but rather interfaces with an already operational SLAM-aware server.

One of the critical, and often understated, questions on combining data is whether to pursue a distributed or shared scheme. Shared processes (such as a centralized factor graph server) present the single point of failure issue, while distributed approaches limit the amount of information sharing. As argued throughout this thesis, the trend—as with the biological world—is to go for a centralized approach and leverage consensus amount various pieces of partial information.

[1] "Thin" implies a process that interprets sensor information into factor graph updates, but does not perform the SLAM inference.

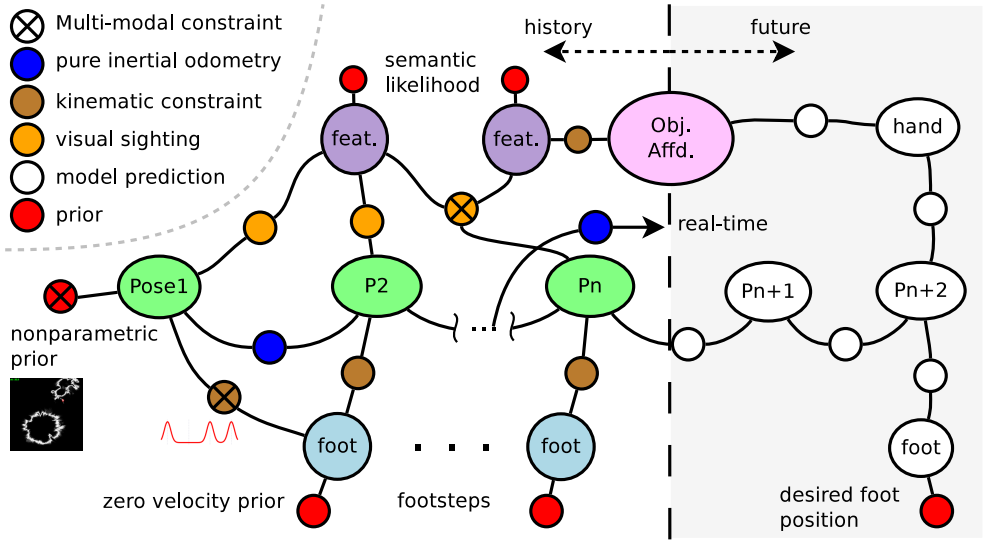


Figure 9-1: Overview illustration of a joint probability nonparametric, multi-modal factor graph representing for the humanoid robot navigating towards some manipulation task.

9.4.1 Humanoid Robot Illustration

In Chapter 1, surrounding the process illustrated in Fig. 1-2, we made the argument that a humanoid robot navigation problem perfectly illustrates the needs for an autonomous *location and dynamics aware* navigation system. We reiterate the humanoid navigation example here, and try represent all the major elements that a autonomous navigation and state estimate system.

Consider a robot approaching an object to be manipulated, and needs to guide its trajectory towards a specified goal. A state estimate in the relative relationship should smoothly converge towards truth as the robot approaches the intended goal, and not diverge as is common with separated odometry (which drift) and mapping techniques.

In Chapter 3, we developed the intuition for assembling a navigation-type factor graph well suited to this goal, since state estimation, map estimation, and dynamic guidance in the environment can be combined into a centralized estimation process. We reproduce the joint probability factor graph here in Fig. 9-1.

Fig. 9-1 shows the future robot trajectory planning directly coupled to the current state estimate, towards a specific user-defined objective. Consider that the legged robot should walk up to a desirable position in front of an identified ob-

ject in the world. The desired standing foot position can be introduced through a desired foot position prior, shown as a prior.

Second, we desire the hand position at or near some affordance object. Using the affordance abstraction allows a known model of an object—with known interaction rules—to be fit to an estimated part of the generated map. The future trajectory plan can then be connected to this affordance estimate and closed loop state estimate and trajectory planning can be achieved. Detection, classification and pose estimation of objects in the world may well be part of the mapping system. Direct interaction between affordance pose estimates and robot motion planning is desirable, since vehicle trajectory is directly affected by observations made.

We argue that common inference generalization does exists, in part based on the arguments above and on a biological system comparison argument: We as humans are able to pick up the controls of a wide variety of platforms—such as cars, planes, helicopters, space ships, or submarines—and are able to learn the control strategies, but rarely question our ability to resolve a navigation state estimate from available data, or our ability to build an intuitive feeling where the dynamics might take the vehicle in the near future based on our actions.

9.4.2 Current Limitations

Thus far, the community have generally assumed that measurement likelihoods used as factor in the graph should be Gaussian, given the computational convenience of existing optimization routines. The Gaussian assumption is entrenched, and satisfies most, but not all measurements. The cost of a highly inconsistent measurement being introduced as a Gaussian term is disastrous for system performance.

Any limitations enacted by the back-end process—such as Gaussian only factors—is transmitted up the navigation process stack, which usually requires a more complicated front-end process to deal with various difficult real-world measurement situations. From a research perspective, there is a risk in assuming that Gaussian only measurement errors are a sufficient assumption, since real-world problems then get pushed over onto the front-end developer and system integrator. A sort of avoidance posture that implies unsatisfactory navigation performance is a limitation on the front-end side of the solution.

Several authors have worked to expand the capability of back-end SLAM optimizers, and improve robustness. In Chapter 2, we discussed several predominant null-hypothesis style approaches, which adapted existing *max-product* style, para-

metric optimization techniques—such as iSAM2—to deal with so-called ”outlier” factors. Our opinion is the probabilistic interpretation of ”outliers” is misunderstood in this context.

The *max-product* style approach of iSAM2 produces a fundamentally different inference result that would be retrieved by a belief space approach using a *sum-product* style approach. We believe that the restrictions due to Gaussian likelihood functions or heuristic based null-hypothesis approaches in many real-world situations outweight the benefits using existing parametric optimization techniques.

As discussed in Chapter 2, null-hypothesis methods exist but usually require careful parameter setup. This requires the front-end processes to dictate how likely a measurement should be ignored. Switch variables [219], as example, require a penalty definition, which dictates a hard decision on whether a measurement factor should be discarded. As discussed in Chapters 3 and 8, many situations are ambiguous associations rather than erroneous measurements and we would like to be able to defer the uncertainty to back-end resolved consensus.

A strong synergy exists between belief interpretations and consensus among multiple hypotheses, but literature suggests little work has been done on *sum-product* style approaches for finding probabilistic consensus with navigation-type factor graphs.

A back-end inference solution that allows the navigation system designer more options, even if the computational cost is higher, could in many situations be a far more attractive option. We stress that the increases computational cost is only palatable under the separations-of-concerns umbrella, where real-time state estimation is separated out as a conjoined operations (as discussed around Fig. 3-6 in Chapter 3).

We believe that a large part of separate-of-concerns can be solved with improved data management technology and interprocess communication. Our work on using a database-centered approach to SLAM follows from, among others, three major requirements on a centralized SLAM system.

Firstly, we do not know ahead of time which data or situations may be of interest. As a result, robot systems capabilities are limited in our ability to utilize and infer traits, patterns, trends or things from gathered sensor data. Secondly, our abilities are constrained by available computational power, ultimately resulting in less robust ’artificial intelligence’. Thirdly, large volumes of data and robust inference techniques require larger computation resources, such as multi-processor architectures, which in turn places large concurrency demands on the design of a robotic computational system.

9.5 Our Contributions

We decompose the problem—through separation-of-concerns—into three major areas: (i.) Robust multi-sensor inference (discussed in Chapter 5); (ii.) Access to high rate odometry (with dynamic calibration, such as inertial in Chapter 4); and, (iii.) A systems approach to combining the fast and robust elements over varying timescales using the duplication method, as discussed in Chapters 3 & 7.

Our premise is that existing (Gaussian only) measurement likelihood models are too restrictive, and in this thesis we introduce a more robust *sum-product* style inference solution. We also introduce new nonparametric likelihood functions to better lever the *sum-product* style inference.

In the first section below, we look at how a user might assemble a nonparametric factor graph in a transactional-navigation approach to interact with the SLAM service. Next we discuss contributions in *sum-product* inference that makes the more robust SLAM service solution possible. Lastly, we discuss the novel continuous time inertial odometry factors that would bring this client-server type system in line with existing INS/GPS type state estimation.

9.5.1 A New Kind of Front-end Process

By separating the real-time, high-bandwidth state estimation operations from the robust multi-sensor fusion operations in a duplication framework—as outlined in Fig. 3-6, page 94—we increase the available computation time per inference cycle.

Using our approach, the robot navigation designer now has the freedom to build a "thin" front-end process in a client-server model, using a wider class of multi-modal and nonparametric measurement likelihood models. These models are used in a factor graph to produce non-Gaussian marginal posterior estimates. The *Multi-modal iSAM* robust inference solution allows this new variety likelihood models and associated of sensing modalities to extract a consensus posterior belief state estimate.

For example, ambiguity in measurement data association can be modeled as multi-modal belief between multiple variables in the system, which, to the best of our knowledge, is novel method of performing data association. This would allow back-end solution to internally find Bayesian consensus amongst ambiguous data without modifying the structure of the factor graph. This is a vital difference to previous approaches.

Our break from the common Gaussian assumption also allows more general nonparametric measurement likelihood models into the joint probability factor

graph, which is a significant departure from the current Gaussian status quo. Relaxing the Gaussian constraint requirement allows the user significantly more freedom in representing ambiguous uncertainties from real world measurements and deferring consensus to the inference algorithm. We showed through practical experiment how beam-forming type measurements with truly nonparametric measurement likelihoods can be included into the navigation and state estimation process.

9.5.2 Generalized *Sum-Product* Multi-modal Inference

Rather than trying to retrofit previous unimodal optimization style SLAM solutions for multi-hypothesis solutions, we implemented a new algorithm to propagate multi-modal beliefs on the Bayes tree factorization of the navigation style factor graphs. Unlike Thrun et al.'s [224] histogram (fixed grid approaches), our approach instead works freely on the infinite continuous set, maintaining only a select set of variables to define where probabilistic belief is concentrated.

The algorithm is based on the Bayes tree symbolic refactorization which was first developed by iSAM2[2], allowing for incremental nonparametric updates to the belief states of all variables; and we believe this might also be a first in *sum-product* style algorithms that can deterministically recycle computations for stable regions of a nonparametric joint probability distribution.

Our algorithm is called *Multi-modal iSAM* and scratch developed and exists in an software implementation using the Julia programming language. We believe this may be the first larger scale *sum-product* style inference algorithm in robotics (thousands of variable processed in a non-Gaussian, multi-modal manner). The algorithm is able to do general posterior recovery for arbitrary user-defined functions satisfying mild conditions. The generalized inference algorithm can compute non-Gaussian marginals from the full posterior belief of any variables the joint probability factor graph model. Our approach is based on nested, multi-scale Gibbs sampling which uses the user-defined measurement likelihood models as proposal distributions, thereby ensuring that all modes have reasonable possibility of being selected.

In our multi-modal inference approach, the structure of the factor graph remains unchanged throughout the inference process. By contrast, some null hypothesis methods modify the graph structure during a preprocessing step. Maintaining the topological structure of the factor graph allows us to exploit the full

[2]incremental Smoothing and Mapping

structure within joint probability by using the Bayes tree, which in turn maintains the incremental inference capability discussed in Chapter 5.

We have chosen to approximate the marginal posterior densities of each clique in the Bayes tree with a set of samples. These samples can be used to construct belief propagation messages for inference across the entire tree. We choose to use kernel density estimation to smooth the belief estimates and naturally encapsulate multi-hypothesis characteristics in a selective multi-modal framework.

Furthermore, the Bayes tree symbolically encodes the multi-hypothesis belief of an otherwise obscure variable interaction shown by the factor graph alone. We limit the number of samples according to available computational resources and thereby concentrate computation around the prominent modes in the system. The approximation and inference scheme is designed to have minimal approximation error around the dominant modes that are being tracked.

In Chapter 5 Table 5.1, we identified nine computational principles which reduce the computational complexity of the *sum-product* style algorithm. The principles were either exact symbolic re-factorizations, or asymptotic approximations of the true belief. The loss of information is a vital aspect in obtaining a computationally tractable algorithm and corresponds to discarding low likelihood modes while focusing computation on dominant modes, with very minor approximation errors.

The key advantage of our approach is that when more information becomes available, the inference on the clique is repeated and a new set of samples are generated which may now focus computation on an altered subset of dominant modes, retrieving modes that may have been lost before. Thereby data association is deferred into the back-end inference process and assignment is available through individual variable beliefs rather than factor graph structure.

In Chapter 6, we evaluated several canonical examples nonparametric and non-singular examples. We showed that consensus on continuous domains are possible with the three door example. We showed that multi-modal posteriors are possible with the four door example. We shows how the ambiguous (non-singular) SLAM-e-donut example is able to recover multiple correct modes form Gaussian only likelihood models. We also illustrated three dimensional inference is possible with basic acoustic structure from motion examples.

In Chapter 8 we showed how ambiguous data association (loop closures) could be modeled and realistically computed for the Victoria Park dataset, representing a model of many thousands of dimensions. We also noted that our factor graph construction theoretically had around 2^{1700} captured in the uncertain associations. We note that during *sum-product* style inference, all theoretical modes in the system

have some probability of occurring in the posterior belief, however, nearly all the modes are highly unlikely and are therefore ignored. We believe this is a first.

We also investigated a purely nonparametric measurement likelihood factor graph, using raw beam formed acoustic measurement data. We were able to recover a trajectory estimate without having to perform *max-product* data preprocessing, and belief this approach is also a first.

9.5.3 Real-time Odometry and Inertial Odometry

In Chapters 1 & 2, we identified the need for a measurement factor suited for factor graph based navigation. In Chapter 4, we derived a novel continuous time, second order inertial odometry residual function. Validation test in that chapter was able to confirm that the measurement model is working.

Real data experiments with a hand-held visual-inertial system, as well as a humanoid robot, show that the inertial odometry factor does infact work and is able to use navigation information from loop closures, or an other measurement likelihood models we can integrate into the factor graph.

9.6 Future Direction

We believe that the major theoretical development that is required—as we discussed in Section 5.4.3—is to connect contractive mapping theory from functional embeddings to detailed balance and stationary distributions in Markov Chain Monte Carlo theory. Once we know the criteria for our inference operator $T_{\mathcal{Q}}$ to be a contractive mapping, we can instead focus on how to satisfy that criteria for developing newer and better inference algorithms.

In addition, in Chapter 5 we showed through frequency embeddings how most computation should be focused on 'mid-frequency' computations. We would like to investigate this mid-frequency phenomenon more.

Staying with Hilbert embeddings, recent work on the connection between reproducing Hilbert space embeddings and conditional distributions [17, 214] offers a different and potentially more efficient mechanism for approximate Bayesian computations. Further methods, such as progressive Bayes [204] might prove very powerful indeed.

We would also like to investigate the non-singular system, encountered with the SLAM-e-donut example in Chapter 6, in more detail. It might be that the non-singular systems, which originate from ambiguous Gaussian systems in fact have

non-Real posterior distributions. We are interested to see if complex values distributions have anything to do with the observed multi-modality.

From a more practical standpoint, we intend to further exploit the Bayes tree structure by combining hybrid Gaussian mixture models for portions of the tree which involve distinct multi-modality, and switching to more costly smooth non-parametric methods (such as kernel density estimates) when is other non-Gaussian beliefs are encountered – as we discussed in Section 5.5.5. We anticipate the use of Gaussian mixture models, with epsilon exact belief approximation accuracy is a reasonable step towards reducing computation while tracking dominant modes.

We continue to work towards a central inference task which simultaneously localizes & maps the environment (sparsely), and can also plan future motions through predictive models.

We can also consider newer deep learning based state estimation with our non-Gaussian approach. We could train neural networks to produce beliefs over measurement variables (represented in mixture models for example), and then incorporate these as likelihood factors in a factor graph model. The acoustic beam forming example is closest example presented in the thesis. Combining likelihoods from "learned" sensor perception systems is certainly an interesting avenue for future research.

9.7 Conclusion

Increased computational throughput coupled with lower cost sensing and promise of improved robotics have spurred rapid development navigation technology. More people today are working on state estimation development than ever before, building systems that service a variety of new and improved automated tasks.

The algorithm and framework presented in this thesis provides a route to implement a real-time state estimate that interacts through a friendly centralized database architecture, providing robust multi-modal inference. The intention is to build a notion of *location and dynamic awareness* for use by robots. Many challenges remain to ensuring that a SLAM style state estimate solution is robust and computationally feasible, but we expect this work to contribute towards our general understanding and help future endeavors find bigger and greater discoveries.

Appendix A

Identities

A.1 Notation

We follow the convention

$$\begin{aligned} \mathbf{R}\left({}^{b_i}_{w}\mathring{\mathbf{q}}\right) = {}^{b_i}_{w}\mathbf{R} \qquad {}^{w}_{b_i}\mathring{\mathbf{q}} = \begin{bmatrix} \cos\frac{\theta}{2} \\ \hat{\omega}\sin\frac{\theta}{2} \end{bmatrix} \qquad |\theta| < \pi \\ {}^{b_i}\mathbf{v} = {}^{b_i}_{w}\mathbf{R}\,{}^{w}\mathbf{v} \end{aligned} \tag{A.1}$$

with regular conversion between quaternions and rotation matrices.

A.2 Mathematical constructs

This work makes extensive use of the matrix exponential, which is defined by Taylor series:

$$\mathbf{e}^{M} = \sum_{k=0}^{\infty} \frac{\mathbf{M}^k}{k!} \tag{A.2}$$

such that $\mathbf{M} \in \mathbb{R}^{N \times N}$. The matrix exponent provides an elegant formulation, but care must be taken with practical floating point computation of $\mathbf{e}^{\mathbf{M}}$. A (numerically) accurate matrix exponential, such as the Pade approximation, should be used [154]. Implementations of matrix exponentiation by Pade' approximation are common.

Our derivation exploits the manifold geometry of $SO(3)$ around the current best estimate of a vehicle's trajectory in 3 dimensions. We parameterize perturba-

tions on this manifold with Lie algebra $[\varphi_\times] \in so(3)$. The members of $so(3)$ consist of minimal parameters, $\varphi \in \mathbb{R}^3$, in skew-symmetric matrices:

$$[\varphi_\times] = \begin{bmatrix} 0 & -\varphi_z & \varphi_y \\ \varphi_z & 0 & -\varphi_x \\ -\varphi_y & \varphi_x & 0 \end{bmatrix} = \varphi_x \hat{\mathbf{E}}_x + \varphi_y \hat{\mathbf{E}}_y + \varphi_z \hat{\mathbf{E}}_z$$

with usual Lie algebra basis elements $\{\hat{\mathbf{E}}_x, \hat{\mathbf{E}}_y, \hat{\mathbf{E}}_z\}$:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.3}$$

The Vee operator, $[\varphi_\times]^\vee = \varphi$, vectorizes the coefficients on the associated orthogonal basis.

General rotations are operated by rotation matrices as part of the Lie Group ${}^b_a\mathbf{R} \in SO(3)$, where the matrix exponent is used to map members of the algebra onto the group, $\mathbf{e}^{[{}^b_a\varphi_\times]} : so(3) \to SO(3)$. Rodrigues' formula gives a closed form expression:

$$\mathbf{R} = \mathbf{e}^{[\varphi_\times]} = \mathbf{I} + \frac{\sin\|\varphi\|}{\|\varphi\|}[\varphi\times] + \frac{1-\cos\|\varphi\|}{\|\varphi\|^2}[\varphi\times]^2 \tag{A.4}$$

Practical implementations should take care as $\|\varphi\| \to 0$, and we suggest using eq. (A.2) directly when $\|\varphi\|$ drops below some reasonable threshold.

The logarithm map is used to convert elements from the Lie Group to Lie algebra, and is taken from Proposition 2.2 in [184], subject to $\|\varphi\| < \pi$ and $\text{Tr}\left({}^b_a\mathbf{R}\right) \neq -1$:

$$\log_{SO(3)} {}^b_a\mathbf{R} = \frac{\phi}{2\sin\phi}\left({}^b_a\mathbf{R} - {}^b_a\mathbf{R}^T\right) \tag{A.5}$$

where $1 + 2\cos\phi = \text{Tr}\left({}^b_a\mathbf{R}\right)$.

We use singularity free positive scalar quaternions, ${}^{b_i}_b\mathring{\mathbf{q}}$, to store the each pose's orientation in the world frame. Body orientation at time t_i is denoted as ^{b_i}. Rotations may be directly composed with quaternions by using the Hamiltonian product.

Appendix B

Code Snippets

B.1 Three Doors Canonical Example

B.1.1 Mathematica Code for Three Doors example

```
z2 = 5;
phi12 = Simplify[E^(-1/2*((x2 - x1 - z2)^2))];

Na = E^(-1/2 ((x1)^2));
Nb = E^(-1/2 ((x1 - 10)^2));
Nc = E^(-1/2 ((x1 - 30)^2));

phi1 = Na + Nb + Nc;
M12 = phi1 * phi12;
m2 = Simplify[Integrate[M12, {x1, -Infinity, Infinity}]];
Plot[m2, {x2, -10, 50}]

z3 = 5;
phi23 = Simplify[E^(-1/2*((x3 - x2 - z3)^2))];

N3a = E^(-1/2 ((x3)^2));
N3b = E^(-1/2 ((x3 - 10)^2));
N3c = E^(-1/2 ((x3 - 30)^2));

phi3 = N3a + N3b + N3c;
```

```
M23 = Simplify[phi23 * phi3 * m2];
pX3 = Expand[Integrate[M23, {x2, -Infinity, Infinity}]]
Plot[pX3, {x3, 0, 50}, PlotRange -> All]
```

B.1.2 MCMC code snippet

Complete code for the leaf clique:

```
# measurement functions at the leaf node
h1(z1) = z1; # landmark measurement
h2(x1,z2) = x1 + z2; # odometry measurement

# The Measurements for this experiment
Z = zeros(4,1) #  odo measurements
Z[2] = 5.0; # odo
Z[3] = 5.0; # odo

# Define the potentials
# three modes for the first measurement
function phi1(z1)
  w = zeros(3,1);
  w[1] = 1.0/3.0;
  w[2] = 1.0/3.0;
  w[3] = 1.0 - w[1] - w[2];
  mu = zeros(3,1);
  mu[1] = 0.0;
  mu[2] = 10.0;
  mu[3] = 30.0;

  SEL = rand();
  W = cumsum(w);
  for i in 1:3
    if (W[i]>=SEL)
      z1 = mu[i];
      break;
    end
  end
  return Normal(h1(z1), 1.0)
end
```

```
phi3(z3) = phi1(z3);

# single mode for this odometry measurement between X1, X2
phi12(x1,z2) = Normal(h2(x1,z2), 1.0);

#Compute the first message m2
# MCMC -- Following substitution and Gibbs sampling examples in Cassela1992 and Gelfand1
k = 1; # convergence time (burn-in) for Gibbs sampler (Markov chain length)
M = 1000; # number of independent trials (as per Gelfand and Cassela)

x1 = zeros(k,M);
x2 = zeros(k,M);
m2 = zeros(M);

for m = 1:M
  for i = 1:k
    # generate sample from phi1 potential (Multimodal)
    # Gibbs sampling -- sequentially sample a single variable at a time
    x1[i,m] = rand(phi1(Z[1]));
    x2[i,m] = rand(phi12(x1[i,m], Z[2]));
  end
  m2[m] = x2[k,m];
end
p1 = plot(x=m2, Geom.histogram(bincount=100))
```

B.1.3 Belief Propagation

Julia code for incorporating the measurements potentials, and marginalizing X_2 at the root clique, is presented below.

```
# single mode for this odometry measurement between X2, X3, Z3
phi32exp(x3, x2, z) = exp(-0.5*((h3(x2,z)-x3)^2));

x3 = linspace(0,50,100);

function approxG(x3, m2, Z)
  M = size(m2,1)
  acc = 0.0
```

```
  for l = 1:M
    acc += 1.0/M * phi32exp(x3, m2[l], Z[3])
  end
  return acc
end

# include the measurement Z4
pz4(x3) = exp(-0.5*((x3-0.0)^2)) + exp(-0.5*((x3-10.0)^2)) + exp(-0.5*((x3-30.0)^2

X3 = zeros(M,1);
i = 0;
for val = x3
  i += 1;
  X3[i] = pz4(val) * approxG(val, m2, Z);
end

p4 = plot(x=x3,y=X3,Geom.line)
draw(PNG("pX3_Z1234.png", 6inch, 3inch), p4)
```

Appendix C

Example Intraclique operation

An example intraclique operation, as illustrated in Fig. 5-7 and Fig. 5-10.

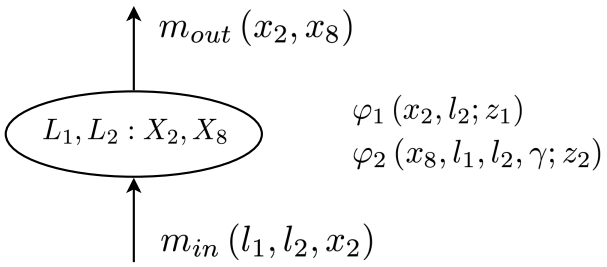


Figure C-1: Canonical example of clique belief propagation operation

Consider a canonical example shown in Fig. C-1, two likelihood models associated with this clique are used to capture data association uncertainty between robot poses X_i with potential sightings to landmarks L_j:

$$\varphi_1\,(x_2, l_2; z_1) = [\,Z_1 \,|\, X_2, L_2\,] = \mathcal{N}\,(\mu = \ominus X_2 \,\oplus\, L_2, \Sigma) \tag{C.1}$$

and

$$\varphi_2\,(x_8, l_1, l_2) = [\,Z_2, \Gamma \,|\, X_8, L_1, L_2\,] \propto [\,Z_2 \,|\, \gamma, X_8, L_1, L_2\,]\,[\Gamma] \tag{C.2}$$

$$[\,Z_2 \,|\, \gamma, X_8, L_1, L_2\,] = \frac{\gamma}{2}\mathcal{N}\,(\mu_1, \Sigma) + \frac{1-\gamma}{2}\mathcal{N}\,(\mu_2, \Sigma) \tag{C.3}$$

$$\mu_i = \ominus X_8 \,\oplus\, L_i \qquad\qquad \forall i \in \{1, 2\} \tag{C.4}$$

$$\gamma \sim Cat\left(\begin{bmatrix}0.5 & 0.5\end{bmatrix}\right). \tag{C.5}$$

Note that φ_2 represents a bimodal likelihood based on a hyperparameter 0.5 which the user specified.

Step 1: Determine all local potential functions from each unclaimed factor associated with frontal variables, using the tree leaves to root ordering as a guide. For example in clique $L_1, L_2 : X_2, X_8$, variable l_1 is associated with the local potential φ_2 and the incoming message $m_{in}\,(l_1, l_2, x_2)$, as shown in the table below:

	l_1	l_2	x_2	x_8
φ_1		x	x	
φ_2	x	x		x
m_1	x		x	
m_2		x		

Step 2a: Likelihood functions are used to relate desired variables to current beliefs using n sample points

$$\{l_2^{(1:n)}\}^{(k)'} \equiv \underset{i=1:n}{\text{solve root}} \left[\varphi_1 \left(x_2^{(i,k)}, {l_2}^{(i)}; z^{(i)} \right), \qquad z^{(i)} \sim \mathcal{N}\left(z_1, \Sigma\right) \right] \tag{C.6}$$

from where we construct the approximated belief over L_2 using kernel density estimation

$$\left[\hat{L}_2 \,|\, y_z \right]_{\varphi_1}^{(k)'} \propto \text{KDE} \left(\{l_2^{(1:n)}\}^{(k)'} \right). \tag{C.7}$$

Step 2b: Prior beliefs may be marginalized to the desired variable such as $[\, L_1 \,]$ from an incoming message

$$[\, L_1 \,]_m = \int_{\mathbb{R}^2} [\, L_1, L_2, X_2 \,]_m \, dx_2 dl_2, \tag{C.8}$$

Step 3: Compute approximate belief over each variable using available conditional distributions – for example, we compute next iteration $[\, L_1 \,]^{(k+1)}$

$$[\, L_1 \,]^{(k+1)} \propto [\, L_1 \,]_m \times [\, L_1 \,|\, y_z \,]_{\varphi_2}^{(k)'} \tag{C.9}$$

Step 4: Iterate this procedure across all variables in the clique until convergence. At which point the belief over frontal variables $[\, L_1 \,|\, y_{z:} \,]$ and $[\, L_2 \,|\, y_{z:} \,]$ replace previously stored beliefs. Belief over all separator variables are passed along

as the outgoing message:

$$m_{out}\left(x_2, x_8\right) = \left[\, \hat{X}_2, \hat{X}_8 \,|\, \hat{Y}_{z:} \,\right] \tag{C.10}$$

Bibliography

[1] Apache Tinkerpop: The benefits of graph computing. `http://tinkerpop.apache.org/`. Accessed: 2017-02-20.

[2] Google: Project tango. `https://get.google.com/tango/`. Accessed: 2017-07-20.

[3] MongoDB: For giant ideas. `https://www.mongodb.com/`. Accessed: 2017-02.

[4] Neo4j: The world's leading graph database. `https://neo4j.com/`. Accessed: 2017-02-20.

[5] Pablo F Alcantarilla, Olivier Stasse, Sebastien Druon, Luis M Bergasa, and Frank Dellaert. How to localize humanoids with a single camera? *Autonomous Robots*, 34(1-2):47–71, 2013.

[6] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK Users' guide*. SIAM, 1999.

[7] Amit Apte, Martin Hairer, AM Stuart, and Jochen Voss. Sampling the posterior: An approach to non-Gaussian data assimilation. *Physica D: Nonlinear Phenomena*, 230(1):50–64, 2007.

[8] Nikolay Atanasov, Menglong Zhu, Kostas Daniilidis, and George J Pappas. Semantic localization via the matrix permanent. In *Robotics: Science and Systems*, 2014.

[9] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[10] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, Sydney, NSW, Australia, 2002.

[11] T. Bailey and H.F. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part I. *Robotics & Automation Magazine*, 13(2):99–110, June 2006.

[12] T. Bailey, E. Nebot, J. Rosenblatt, and H. Durrant-Whyte. Data association for multiple robot navigation: a graph theoretic approach. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 3, pages 2512–2517, April 2000.

[13] Meïli Baragatti, Agnès Grimaud, and Denys Pommeret. Parallel tempering with equi-energy moves. *Statistics and Computing*, pages 1–17, 2013.

[14] Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[15] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Lecture notes in computer science*, 3951:404, 2006.

[16] Bradley M Bell. The iterated Kalman smoother as a Gauss-Newton method. *SIAM Journal on Optimization*, 4(3):626–636, 1994.

[17] Alexandros Beskos, Frank J Pinski, Jesús Marıa Sanz-Serna, and Andrew M Stuart. Hybrid Monte Carlo on Hilbert spaces. *Stochastic Processes and their Applications*, 121(10):2201–2230, 2011.

[18] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *arXiv preprint arXiv:1411.1607*, 2014.

[19] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[20] AT Bharucha-Reid et al. Fixed point theorems in probabilistic analysis. *Bulletin of the American Mathematical Society*, 82(5):641–657, 1976.

[21] S. Birchfield. Derivation of the Kanade-Lucas-Tomasi tracking equation, January 1997.

[22] Stan Birchfield. Klt: An implementation of the kanade-lucas-tomasi feature tracker, 2007.

[23] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning research*, 3(Jan):993–1022, 2003.

[24] Michael Bloesch, Marco Hutter, Mark A Hoepflinger, Stefan Leutenegger, Christian Gehring, C David Remy, and Roland Siegwart. State estimation for legged robots-consistent fusion of leg kinematics and IMU. *Robotics*, page 17, 2013.

[25] Kenneth R Britting. *Inertial Navigation Systems Analysis*. Artech House, 1971.

[26] D. C. Brown. The bundle adjustment - progress and prospects. *International Archives Photogrammetry*, 21(3), 1976. Paper number 3–03 (33 pages).

[27] Adam Bry, Abraham Bachrach, and Nicholas Roy. State estimation for aggressive flight in gps-denied environments using onboard sensing. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012.

[28] R. Camilli, C.M. Reddy, D.R. Yoerger, B. Van mooy, M.V. Jakuba, J.C. Kinsey, C.P. McIntyre, S.P. Sylva, and J.V. Maloney. Tracking hydrocarbon plume transport and biodegradation at deepwater horizon. *Science*, 330(6001):201–204, 2010.

[29] Gilles Celeux, Merrilee Hurn, and Christian P Robert. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, 95(451):957–970, 2000.

[30] Averil Burton Chatfield Chatfield. *Fundamentals of high accuracy inertial navigation*, volume 174 of *Progress in Astronautics and Aeronautics*. AIAA, 1997.

[31] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[32] Gregory S Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, volume 2. Springer, 2011.

[33] Gregory S Chirikjian and Alexander B Kyatkin. *Engineering applications of noncommutative harmonic analysis: with emphasis on rotation and motion groups*. CRC press, 2000.

[34] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013.

[35] Daniel Edward Clark and Judith Bell. Bayesian multiple target tracking in forward scan sonar images using the PHD filter. *IEE Radar, Sonar and Navigation*, 152:327–334, 2005.

[36] Peter Corke, Jorge Lobo, and Jorge Dias. An introduction to inertial and visual sensing. *The International Journal of Robotics Research*, 26(6):519–535, 2007.

[37] Robert Cowell. Advanced inference in Bayesian networks. In *Learning in graphical models*, pages 27–49. Springer, 1998.

[38] Mark Cummins and Paul Newman. Appearance-only SLAM at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[39] T.A. Davis, J.R. Gilbert, S.I. Larimore, and E.G. Ng. A column approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):353–376, 2004.

[40] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

[41] F. Dellaert. Square Root SAM: Simultaneous location and mapping via square root information smoothing. In *Robotics: Science and Systems (RSS)*, Cambridge, MA, 2005.

[42] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, December 2006.

[43] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.

[44] JF DeNatale, RL Borwick, C Tsai, PA Stupar, Y Lin, RA Newgard, RW Berquist, and M Zhu. Compact, low-power chip-scale atomic clock. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 67–70. IEEE, 2008.

[45] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York, 2000.

[46] Charles S Draper. *Mechanization of Inertial Guidance Systems*. 1963.

[47] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

[48] H.F. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part II. *Robotics & Automation Magazine*, 13(3):108–117, September 2006.

[49] S Ebcin and M Veth. Tightly-coupled image-aided inertial navigation using the unscented kalman filter. Technical report, DTIC Document, 2007.

[50] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *Instrumentation and Measurement, IEEE Transactions on*, 57(1):140–149, 2008.

[51] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.

[52] R. Eustice, M. Walter, and J. Leonard. Sparse extended information filters: Insights into sparsification. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3281–3288. IEEE, 2005.

[53] R.M. Eustice, H. Singh, J.J. Leonard, and M.R. Walter. Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. *Intl. J. of Robotics Research*, 25(12):1223–1242, December 2006.

[54] Maurice Fallon, Scott Kuindersma, Sisir Karumanchi, Matthew Antone, Toby Schneider, Hongkai Dai, Claudia Pérez D'Arpino, Robin Deits, Matt DiCicco, Dehann Fourie, et al. An architecture for online affordance-based perception and whole-body planning. *Journal of Field Robotics*, 2014.

[55] Maurice F Fallon, Matthew Antone, Nicholas Roy, and Seth Teller. Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing. Technical report, DTIC Document, 2014.

[56] Jay Farrell. *Aided navigation: GPS with high rate sensors*. Electronic Engineering. McGraw-Hill New York, 2008.

[57] Jay Farrell and Matthew Barth. *The global positioning system and inertial navigation*, volume 61. McGraw-Hill New York, 1999.

[58] Farhan Feroz and MP Hobson. Multimodal nested sampling: an efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses. *Monthly Notices of the Royal Astronomical Society*, 384(2):449–463, 2008.

[59] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. *Robotics: Science and Systems (RSS)*, 2015.

[60] Thomas E Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Multi-target tracking using joint probabilistic data association. In *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, pages 807–812. IEEE, 1980.

[61] D. Fourie. Optimisation of electric long endurance unmanned aerial vehicles. Master's thesis, University of Johannesburg, 2011.

[62] D. Fourie, J.J. Leonard, and M. Kaess. A nonparametric belief solution to the Bayes tree. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS*, Daejeon, Korea, Oct 2016.

[63] Dehann Fourie. Rome.jl, 2017.

[64] Dehann Fourie, Samuel D. Claassens, Sudeep Pillai, Roxana Mata, and John J. Leonard. Centralized graph databases for mobile Robotics. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Singapore, June 2017.

[65] Dehann Fourie and contributors. Caesar.jl, 2016.

[66] Dehann Fourie, Brent Tweddle, Steve Ulrich, Alvar Saenz-Otero, MC OâĂŹConnor, BE Tweddle, JG Katz, K Makowka, A Saenz-Otero, and DW Miller. Vision-based relative navigation and control for autonomous spacecraft inspection of an unknown object. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013.

[67] Dehann Fourie, Brent E Tweddle, Steve Ulrich, and Alvar Saenz-Otero. Flight results of vision-based navigation for autonomous spacecraft inspection of unknown objects. *Journal of Spacecraft and Rockets*, 2014.

[68] Dehann Fourie, Kenny Uren, and George van Schoor. Heading alignment with summarized inertial pose constraints. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2757–2764. IEEE, 2014.

[69] Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *The Journal of Machine Learning Research*, 5:73–99, 2004.

[70] K.F. Gauss. *Theoria Motus Corporum Coelestium (Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections)*. Little, Brown, and Co., Republished in 1857, and by Dover in 1963, 1809.

[71] Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

[72] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[73] W Morven Gentleman. Least squares computations by givens transformations without square roots. *IMA Journal of Applied Mathematics*, 12(3):329–336, 1973.

[74] Jeff Gill and George Casella. Dynamic tempered transitions for exploring multimodal posterior distributions. *Political Analysis*, 12(4):425–443, 2004.

[75] Clark R Givens, Rae Michael Shortt, et al. A class of Wasserstein metrics for probability distributions. *The Michigan Mathematical Journal*, 31(2):231–240, 1984.

[76] Matthew C. Graham, Jonathan P. How, and Donald E. Gustafson. Robust incremental SLAM with consistency-checking. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[77] Arthur Gretton. Introduction to RKHS, and some simple kernel algorithms. *Adv. Top. Mach. Learn. Lecture Conducted from University College London*, 2016.

[78] M.S. Grewal, L.R. Weill, L.R. Weill, and A.P. Andrews. *Global positioning systems, inertial navigation, and integration*. Wiley-Blackwell, 2007.

[79] Paul D Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech House, GNSS Technology and Application Series, 2008.

[80] Donald Gustafson, John Dowdle, and Karl Flueckiger. A deeply integrated adaptive GPS-based navigator with extended range code tracking. In *Position Location and Navigation Symposium, IEEE 2000*, pages 118–124. IEEE, 2000.

[81] D. Hähnel, D. Fox, W. Burgard, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, 2003.

[82] Eldon C Hall, Joseph D Sabo, Samuel A Forter, Ralph R Ragan, JH Laning, David G Hoag, Wallace E Vander Velde, Daniel J Lickly, and Edward M Copps Jr. Polaris guidance system, September 11 1984. US Patent 4,470,562.

[83] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013.

[84] Ankur Handa, Michael Bloesch, Viorica Pătrăucean, Simon Stent, John McCormac, and Andrew Davison. gvnn: Neural network library for geometric computer vision. In *Computer Vision–ECCV 2016 Workshops*, pages 67–82. Springer, 2016.

[85] Johanna Hansen, Dehann Fourie, James C Kinsey, Clifford Pontbriand, John Ware, Norm Farr, Carl L Kaiser, and Maurice Tivey. Autonomous acoustic-aided optical localization for data transfer. In *OCEANS'15 MTS/IEEE Washington*, pages 1–7. IEEE, 2015.

[86] Kenneth M Hanson. Markov Chain Monte Carlo posterior sampling with the Hamiltonian method. In *Medical Imaging 2001*, pages 456–467. International Society for Optics and Photonics, 2001.

[87] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, volume 15, pages 147–151. Manchester, UK, 1988.

[88] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[89] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.

[90] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Towards consistent vision-aided inertial navigation. In *Algorithmic Foundations of Robotics X*, pages 559–574. Springer, 2013.

[91] Matthew D Hoffman and Andrew Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *The Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[92] Jeroen D Hol, TB Schon, and Fredrik Gustafsson. A new algorithm for calibrating a combined camera and imu sensor unit. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 1857–1862. IEEE, 2008.

[93] Berthold Horn. *Robot vision*. MIT press, 1986.

[94] Berthold KP Horn. Relative orientation revisited. *JOSA A*, 8(10):1630–1638, 1991.

[95] A. Huang, E. Olson, and D. Moore. LCM: Lightweight communications and marshalling. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[96] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Robotics Research*, pages 235–252. Springer, 2017.

[97] G Huang, M Kaess, J Leonard, and S Roumeliotis. Analytically-Selected Multi-Hypothesis Incremental MAP Estimation. In *International Conference on Acoustics, Speech, and Signal Processing*, British Columbia, Canada, May 2013.

[98] Guoquan Huang. *Improving the Consistency of Nonlinear Estimators: Analysis, Algorithms, and Applications*. PhD thesis, University of Minnesota, Dept. of Computer Science and Engineering, 2012.

[99] Tiffany A Huang and Michael Kaess. Towards acoustic structure from motion for imaging sonar. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 758–765. IEEE, 2015.

[100] Alexander T Ihler and David A McAllester. Particle belief propagation. In *International Conference on Artificial Intelligence and Statistics*, pages 256–263, 2009.

[101] Alexander T Ihler, Erik B Sudderth, William T Freeman, and Alan S Willsky. Efficient multiscale sampling from products of Gaussian mixtures. *Advances in Neural Information Processing Systems*, 16:1–8, 2004.

[102] V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *Intl. Conf. on Information Fusion (FUSION)*, pages 2154–2161, Singapore, July 2012.

[103] V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *J. of Robotics and Autonomous Systems*, 61(8):721–738, August 2013.

[104] Shigeru Itoh. Random fixed point theorems with an application to random differential equations in banach spaces. *Journal of Mathematical Analysis and Applications*, 67(2):261–273, 1979.

[105] H. Johannsson. *Toward Lifelong Visual Localization and Mapping*. PhD thesis, Computer Science and Artificial Intelligence Laboratory, MIT, June 2013.

[106] Carl Johnson. Adaptive corrective alignment for a carouseling strapdown ins. In *American Control Conference, 1984*, pages 1856–1861. IEEE, 1984.

[107] Eagle Jones, Andrea Vedaldi, and Stefano Soatto. Inertial structure from motion with autocalibration. In *Proc. IEEE IntâĂŹl Conf. Computer Vision Workshop on Dynamical Vision, Rio de Janeiro, Brazil*, 2007.

[108] E.S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011.

[109] S. Julier. The scaled unscented transformation. In *Proceedings of the American Control Conference*, volume 6, pages 4555–4559. IEEE, 2002.

[110] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[111] M. Kaess. *Incremental Smoothing and Mapping*. PhD thesis, Georgia Institute of Technology, December 2008.

[112] M. Kaess and F. Dellaert. Covariance recovery from a square root information matrix for data association. *J. of Robotics and Autonomous Systems*, 57(12):1198–1210, December 2009.

[113] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: An algorithmic foundation for probabilistic robot mapping. In *Intl. Workshop on the Algorithmic Foundations of Robotics, WAFR*, Singapore, December 2010.

[114] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: Enabling incremental reordering and fluid relinearization for online mapping. Technical Report MIT-CSAIL-TR-2010-021, Computer Science and Artificial Intelligence Laboratory, MIT, January 2010.

[115] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31:217–236, February 2012.

[116] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, December 2008.

[117] M. Kaess, S. Williams, V. Indelman, R. Roberts, J.J. Leonard, and F. Dellaert. Concurrent filtering and smoothing. In *Intl. Conf. on Information Fusion (FUSION)*, pages 1300–1307, Singapore, July 2012.

[118] R. Kalman and R. Bucy. New results in linear filtering and prediction theory. *ASME Journal of Basic Engineering*, 83:95–107, 1961.

[119] Frank P Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 1978.

[120] Jonathan Kelly, Srikanth Saripalli, and Gaurav S Sukhatme. Combined visual and inertial navigation for an unmanned aerial vehicle. In *Field and Service Robotics*, pages 255–264. Springer, 2008.

[121] Jonathan Kelly and Gaurav S Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, 2011.

[122] Peter Kimball, John Bailey, Sarah Das, Rocky Geyer, Trevor Harrison, Clay Kunz, Kevin Manganini, Ken Mankoff, Katie Samuelson, Thomas Sayre-McCord, et al. The whoi jetyak: An autonomous surface vehicle for oceanographic research in shallow or dangerous waters. In *Autonomous Underwater Vehicles (AUV), 2014 IEEE/OES*, pages 1–7. IEEE, 2014.

[123] James C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft*, Lisbon, Portugal, September 2006. Invited paper.

[124] Takashi Kitamura, Sachie K Ogawa, Dheeraj S Roy, Teruhiro Okuyama, Mark D Morrissey, Lillian M Smith, Roger L Redondo, and Susumu Tonegawa. Engrams and circuits crucial for systems consolidation of a memory. *Science*, 356(6333):73–78, 2017.

[125] Uffe Kjærulff. Inference in Bayesian networks using nested junction trees. In *Learning in Graphical Models*, pages 51–74. Springer, 1998.

[126] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 225–234, Nara, Japan, November 2007.

[127] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, MA, 2009.

[128] Twan Koolen, Jesper Smith, Gray Thomas, Sylvain Bertrand, John Carff, Nathan Mertins, Douglas Stephen, Peter Abeles, Johannes Englesberger, Stephen Mccrory, et al. Summary of team ihmc's virtual robotics challenge entry. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 307–314. IEEE, 2013.

[129] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.

[130] Frank O Kuehnel. Robust Bayesian estimation of nonlinear parameters on SE(3) lie group. *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, 735:176–186, 2004.

[131] DP Landau, Shan-Ho Tsai, and M Exler. A new approach to monte carlo simulations in statistical physics: Wang-landau sampling. *American Journal of Physics*, 72(10):1294–1302, 2004.

[132] Yasir Latif, César Cadena, and José Neira. Robust loop closing over time for pose graph SLAM. *The International Journal of Robotics Research*, page 0278364913498910, 2013.

[133] Yasir Latif, Cesar Cadena Lerma, and Jose Neira. Robust loop closing over time. In *Robotics: Science and Systems (RSS)*, Sydney, Australia, July 2012.

[134] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, et al. A perception-driven autonomous urban vehicle. *J. of Field Robotics*, 25(10):727–774, 2008.

[135] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442–1447, Osaka, Japan, 1991.

[136] S. Leutenegger, M. Chli, and R.Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555, November 2011.

[137] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, December 2014.

[138] M. Li and A. I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *Intl. J. of Robotics Research*, 32(6):690–711, May 2013.

[139] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[140] S Lowry, N Sunderhauf, P Newman, J. J. Leonard, D Cox, P Corke, and M. Milford. Visual place recognition: A survey. *IEEE Trans. Robotics*, 32(1):1–19, 2016.

[141] F. Lu and E. Milios. Globally consistent range scan alignment for environmental mapping. *Autonomous Robots*, 4:333–349, April 1997.

[142] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, IJCAI'81, pages 674–679, 1981.

[143] T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Trans. Robotics*, 28(1):61–76, February 2012.

[144] Donald A MacKenzie. *Inventing accuracy: A historical sociology of nuclear missile guidance*. MIT press, 1993.

[145] Richard W Madison, Gregory L Andrews, Paul A DeBitetto, Scott A Rasmussen, and Matthew S Bottkol. Vision-aided navigation for small UAVs in GPS-challenged environments. *The Draper Technology Digest*, page 4, 2008.

[146] Pat Marion. Director: A robotics interface and visualization framework, 2015.

[147] A. Martinelli. Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *Robotics, IEEE Transactions on*, 28(1):44–60, February 2012.

[148] Dan McLeod, John Jacobson, Mark Hardy, and Carl Embry. Autonomous inspection using an underwater 3d lidar. In *Oceans-San Diego, 2013*, pages 1–8. IEEE, 2013.

[149] Justin J Miller. Graph database applications and concepts with neo4j. In *Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA*, volume 2324, 2013.

[150] David A Mindell. *Digital Apollo: human and machine in spaceflight*. MIT Press, 2011.

[151] David A Mindell. *Our robots, ourselves: Robotics and the myths of autonomy*. Viking, 2015.

[152] Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[153] Faraz M Mirzaei and Stergios I Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *Robotics, IEEE Transactions on*, 24(5):1143–1156, 2008.

[154] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1):3–49, 2003.

[155] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and

S. Thrun. Junior: The Stanford entry in the Urban Challenge. *Journal of field Robotics*, 25(9):569–597, 2008.

[156] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[157] A.I. Mourikis, N. Trawny, S.I. Roumeliotis, A.E. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *Robotics, IEEE Transactions on*, 25(2):264–280, 2009.

[158] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3565–3572. IEEE, 2007.

[159] Anastasios I Mourikis and Stergios I Roumeliotis. A dual-layer estimator architecture for long-term localization. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[160] Beipeng Mu, Shih-Yuan Liu, Liam Paull, John Leonard, and Jonathan P How. Slam with objects using a nonparametric pose graph. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4602–4609. IEEE, 2016.

[161] Peter Mühlfellner, Mathias Bürki, Michael Bosse, Wojciech Derendarz, Roland Philippsen, and Paul Furgale. Summary maps for lifelong visual localization. *Journal of Field Robotics*, 2015.

[162] Peter Mühlfellner, Paul Furgale, Wojciech Derendarz, and Roland Philippsen. Designing a relational database for long-term visual mapping. 2015.

[163] H. Schmidt N. R. Rypkema, E. M. Fischell. One-way travel-time inverted ultra-short baseline localization for low-cost autonomous underwater vehicles. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages pre–print, 2017.

[164] M Zuhair Nashed and Gilbert G Walter. General sampling theorems for functions in reproducing kernel Hilbert spaces. *Mathematics of Control, Signals and Systems*, 4(4):363–390, 1991.

[165] Radford M Neal. Probabilistic inference using Markov Chain Monte Carlo methods. 1993.

[166] Radford M Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and computing*, 6(4):353–366, 1996.

[167] Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011.

[168] Eduardo Nebot. The sydney victoria park dataset, 2009.

[169] José Neira and Juan D Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on robotics and automation*, 17(6):890–897, 2001.

[170] Peter Nelson, Chris Linegar, and Paul Newman. Building, curating, and querying large-scale data repositories for field robotics applications. In *Field and Service Robotics*, pages 517–531. Springer, 2016.

[171] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 127–136, Basel, Switzerland, October 2011.

[172] R.A. Newcombe, D. Fox, and S. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2015.

[173] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. DTAM: Dense tracking and mapping in real-time. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2320–2327, Barcelona, Spain, November 2011.

[174] Paul Newman and Kin Ho. SLAM-loop closing with visually salient features. In *proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 635–642. IEEE, 2005.

[175] D. Nistér. Preemptive RANSAC for live structure and motion estimation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 199–206 vol.1, 2003.

[176] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 652–659, June 2004.

[177] Gabriel Nützi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *Journal of intelligent & robotic systems*, 61(1-4):287–299, 2011.

[178] Man-Suk Oh and James O Berger. Integration of multimodal functions by monte carlo importance sampling. *Journal of the American Statistical Association*, 88(422):450–456, 1993.

[179] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2011.

[180] E. Olson and P. Agrawal. Inference on networks of mixtures for robust robot mapping. In *Robotics: Science and Systems (RSS)*, July 2012.

[181] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7):826–840, 2013.

[182] Open Source Robotics Foundation Inc. Turtlebot 2: Open-source development kit for apps on wheels., 2016.

[183] Paul Ozog, Matthew Johnson-Roberson, and Ryan M Eustice. Mapping underwater ship hulls using a model-assisted bundle adjustment framework. *Robotics and Autonomous Systems*, 87:329–347, 2017.

[184] Frank Chongwoo Park. *The optimal kinematic design of mechanisms*. PhD thesis, Harvard University, 1991.

[185] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Intl. Joint Conf. on Artificial Intelligence*, pages 1157–1164, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[186] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. AUV navigation and localization: A review. *Oceanic Engineering, IEEE Journal of*, 39(1):131–149, 2014.

[187] Judea Pearl. Bayesian networks. *Department of Statistics, UCLA*, 2011.

[188] Peyton Z Peebles, Jay Read, and Peter Read. *Probability, random variables, and random signal principles*, volume 3. McGraw-Hill New York, NY, 2001.

[189] Clifford Pontbriand, Norm Farr, Johanna Hansen, James C Kinsey, Leo-Paul Pelletier, Jonathan Ware, and Dehann Fourie. Wireless data harvesting using the auv sentry and whoi optical modem. In *OCEANS'15 MTS/IEEE Washington*, pages 1–6. IEEE, 2015.

[190] Yu V Prokhorov. Convergence of random processes and limit theorems in probability theory. *Theory of Probability & Its Applications*, 1(2):157–214, 1956.

[191] Gang Qian, Rama Chellappa, and Qinfen Zheng. Robust structure from motion estimation using inertial data. *JOSA A*, 18(12):2982–2997, 2001.

[192] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[193] DIB Randeniya, Manjriker Gunaratne, S Sarkar, and A Nazef. Calibration of inertial and vision systems as a prelude to multi-sensor fusion. *Transportation Research Part C: Emerging Technologies*, 16(2):255–274, 2008.

[194] A. Ranganathan, M. Kaess, and F. Dellaert. Loopy SAM. In *Intl. Joint Conf. on Artificial Intelligence*, pages 2191–2196, Hyderabad, India, 2007.

[195] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24(6):84–90, December 1979.

[196] D. Ribas, P. Ridao, J. Neira, and J.D. Tardós. SLAM using an imaging sonar for partially structured underwater environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[197] D.M. Rosen, M. Kaess, and J.J. Leonard. An incremental trust-region method for robust online sparse least-squares estimation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1262–1269, St. Paul, MN, May 2012.

[198] D.M. Rosen, M. Kaess, and J.J. Leonard. RISE: An incremental trust-region method for robust online sparse least-squares estimation. *Robotics and Automation, IEEE Transactions on*, 30(5):1091–1108, October 2014.

[199] Nicholas Rotella, Michael Bloesch, Ludovic Righetti, and Stefan Schaal. State estimation for a humanoid robot. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 952–958. IEEE, 2014.

[200] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2564–2571, Los Alamitos, CA, USA, 2011. IEEE Computer Society.

[201] Paul G Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of guidance, control, and dynamics*, 21(1):19–28, 1998.

[202] Paul G Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control, and Dynamics*, 21(2):208–221, 1998.

[203] GT Schmidt, P Savage, A VanBronkhorst, JR Catford, and E Levinson. Strapdown inertial systems. Technical report, Advisory Group for Aerospace Research and Development Neuilly-sur-Seine (France), 1978.

[204] Oliver C Schrempf, Olga Feiermann, and Uwe D Hanebeck. Optimal mixture approximation of the product of mixtures. In *Information Fusion, 2005 8th International Conference on*, volume 1, pages 8–pp. IEEE, 2005.

[205] Aleksandr V Segal. *Iterative Local Model Selection for Tracking and Mapping*. PhD thesis, Department of Engineering Science University of Oxford Michaelmas Term 2014 This thesis is submitted to the Department of Engineering Science, University of Oxford, 2015.

[206] Aleksandr V Segal and Ian D Reid. Hybrid inference optimization for robust pose graph estimation. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2675–2682. IEEE, 2014.

[207] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593 –600, June 1994.

[208] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[209] Anatolij Vladimirovič Skorohod. *Random linear operators*, volume 1. Springer Science & Business Media, 2001.

[210] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Intl. J. of Robotics Research*, 5:56–68, December 1986.

[211] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, pages 467–474, Cambridge, MA, USA, 1988. The MIT Press, Cambridge, MA.

[212] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.

[213] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pages 13–31. Springer, 2007.

[214] Le Song, Kenji Fukumizu, and Arthur Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *Signal Processing Magazine, IEEE*, 30(4):98–111, 2013.

[215] Le Song, Arthur Gretton, Danny Bickson, Yucheng Low, and Carlos Guestrin. Kernel belief propagation. In *International Conference on Artificial Intelligence and Statistics*, volume 15 of JMLR: W&CP 15, 2011.

[216] Le Song, Arthur Gretton, and Carlos Guestrin. Nonparametric tree graphical models. In *International Conference on Artificial Intelligence and Statistics*, volume 9 of JMLR: W&CP 9, pages 765–772, 2010.

[217] E.B. Sudderth, A.T. Ihler, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2003.

[218] E.B. Sudderth, A.T. Ihler, M. Isard, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. *Communications of the ACM*, 53(10):95–103, 2010.

[219] N. Sünderhauf. *Robust Otpimization for Simultaneous Localization and Mapping*. PhD thesis, Technischen Universitat Chemnitz, 2012.

[220] N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, October 2012.

[221] Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.

[222] Robert E Tarjan and Mihalis Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on computing*, 13(3):566–579, 1984.

[223] R Tedrake, M Fallon, S Karumanchi, S Kuindersma, M Antone, T Schneider, T Howard, M Walter, H Dai, R Deits, et al. A summary of team mit's approach to the virtual robotics challenge. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2087–2087. IEEE, 2014.

[224] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2005.

[225] S. Thrun, D. Koller, Z. Ghahramani, H.F. Durrant-Whyte, and A.Y. Ng. Simultaneous mapping and localization with sparse extended information filters. In J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, editors, *Proc. of the Fifth Intl. Workshop on Algorithmic Foundations of Robotics*, Nice, France, 2002.

[226] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 4(3):380–407, 2004.

[227] RB Tic. On the observability and self-calibration of visual-inertial navigation systems. 2008.

[228] Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

[229] David Titterton and John Weston. *Strapdown inertial navigation technology*, volume 17. IET, 2004.

[230] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[231] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors,

Vision Algorithms: Theory and Practice, volume 1883 of *LNCS*, pages 298–372. Springer Verlag, 2000.

[232] Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric A Wan. The unscented particle filter. In *Advances in neural information processing systems*, pages 584–590, 2001.

[233] Harry L Van Trees. *Optimum array processing: Part IV of detection, estimation and modulation theory*, volume 1. Wiley Online Library, 2002.

[234] Michael J Veth. *Fusion of imaging and inertial sensors for navigation*. PhD thesis, 2006.

[235] T. Whelan, H. Johannsson, M. Kaess, J.J. Leonard, and J.B. McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[236] T. Whelan, M. Kaess, M.F. Fallon, H. Johannsson, J.J. Leonard, and J.B. McDonald. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, July 2012. Available as MIT CSAIL Technical Report MIT-CSAIL-TR-2012-020.

[237] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald. Real-time large scale dense RGB-D SLAM with volumetric fusion. *Intl. J. of Robotics Research*, 34(4-5):598–626, 2015.

[238] T. Whelan, M. Kaess, J.J. Leonard, and J.B. McDonald. Deformation-based loop closure for large scale dense RGB-D SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013.

[239] Wikipedia. Wikipedia: Decca Navigator System. `https://en.wikipedia.org/wiki/Decca_Navigator_System`. Accessed: 2017-08-14.

[240] Wikipedia. Wikipedia: getting to philosophy. `https://en.wikipedia.org/wiki/Wikipedia:Getting_to_Philosophy`. Accessed: 2017-07-18.

[241] Grant Wilkins, Dehann Fourie, and Johan Meyer. Critical design parameters for a low altitude long endurance solar powered uav. In *AFRICON, 2009. AFRICON'09.*, pages 1–6. IEEE, 2009.

[242] G. Wornell and P. Golland. Mit Course 6.437, Inference and Information. Lecture notes, 2015.

[243] Walter Wrigley. Schuler tuning characteristics in navigational instruments. *Navigation*, 2(8):282–290, 1950.

[244] Paul Zarchan and Howard Musoff. *Fundamentals of Kalman Filtering: A Practical Approach, 3e.*, volume 232 of *Progress in Astronautics and Aeronautics*. AIAA, 2009.