Multiple-Vehicle Resource-Constrained Navigation in the Deep Ocean

by

Brooks Louis-Kiguchi Reed

S.B., Massachusetts Institute of Technology (2009)

Submitted to the Joint Program in Applied Ocean Science and Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

September 2011

Author .....
Joint Program in Applied Ocean Science and Engineering
Aug 5, 2011

Certified by .....
Franz S. Hover
Finmeccanica Career Development Professor of Engineering, MIT
Thesis Supervisor

Certified by .....
Dana R. Yoerger
Senior Scientist, WHOI
Thesis Supervisor

Accepted by .....
David E. Hardt
Chairman, Department Committee on Graduate Students
MIT

Accepted by .....
James C. Preisig
Chairman, Joint Committee for Applied Ocean Science and Engineering
WHOI

Multiple-Vehicle Resource-Constrained Navigation in the Deep Ocean

by

Brooks Louis-Kiguchi Reed

Submitted to the Joint Program in Applied Ocean Science and Engineering
on Aug 5, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

This thesis discusses sensor management methods for multiple-vehicle fleets of autonomous underwater vehicles, which will allow for more efficient and capable infrastructure in marine science, industry, and naval applications. Navigation for fleets of vehicles in the ocean presents a large challenge, as GPS is not available underwater and dead-reckoning based on inertial or bottom-lock methods can require expensive sensors and suffers from drift. Due to zero drift, acoustic navigation methods are attractive as replacements or supplements to dead-reckoning, and centralized systems such as an Ultra-Short Baseline Sonar (USBL) allow for small and economical components onboard the individual vehicles. Motivated by subsea equipment delivery, we present model-scale proof-of-concept experimental pool tests of a prototype Vertical Glider Robot (VGR), a vehicle designed for such a system. Due to fundamental physical limitations of the underwater acoustic channel, a sensor such as the USBL is limited in its ability to track multiple targets—at best a small subset of the entire fleet may be observed at once, at a low update rate. Navigation updates are thus a limited resource and must be efficiently allocated amongst the fleet in a manner that balances the exploration versus exploitation tradeoff. The multiple vehicle tracking problem is formulated in the Restless Multi-Armed Bandit structure following the approach of Whittle in [108], and we investigate in detail the Restless Bandit Kalman Filters priority index algorithm given by Le Ny et al. in [71]. We compare round-robin and greedy heuristic approaches with the Restless Bandit approach in computational experiments. For the subsea equipment delivery example of homogeneous vehicles with depth-varying parameters, a suboptimal quasi-static approximation of the index algorithm balances low landing error with safety and robustness. For infinite-horizon tracking of systems with linear time-invariant parameters, the index algorithm is optimal and provides benefits of up to 40% over the greedy heuristic for heterogeneous vehicle fleets. The index algorithm can match the performance of the greedy heuristic for short horizons, and offers the greatest improvement for long missions, when the infinite-horizon assumption is reasonably met.

Thesis Supervisor: Franz S. Hover
Title: Finmeccanica Career Development Professor of Engineering, MIT

Thesis Supervisor: Dana R. Yoerger
Title: Senior Scientist, WHOI

Acknowledgments

First of all, I would like to acknowledge Prof. Franz Hover, who I first met during my sophomore year and has been a privilege to work with ever since. I owe much of my current research path to the inspiring mentorship from Prof. Hover, as well as Dr. Jim Bellingham of MBARI, with whom I had two summer internships. I'd also like to thank my co-advisor Dr. Dana Yoerger. While Dana's stories are always entertaining, his advice to always focus on the real problem at hand and to not shy away from things that get the job done has stuck with me.

Our collaborator Dr. Julio Guerrero helped shape the VGR project from the beginning, and I thank him for always providing an upbeat presence to our meetings. Charlie Ambler helped introduce me to the VGR project and passed on a capable prototype vehicle. I'd like to thank Jordan Stanway, who always helps me track down Dana, and who gave very helpful feedback on this thesis. Additionally, I am grateful for the advice of Dr. Jerome Le Ny on the more theoretical aspects of this work.

It has been an exciting couple of years to be part of the Hovergroup. I'd first like to thank Rob Hummel, who has been a close friend since my freshman year and cuts me no slack when I bounce my ideas off him. Josh Taylor explained bandits to me, and was a major influence in getting me interested in optimization. I thank Josh Leighton for recommending a good set of headphones. Brendan Englot is always a source of good advice, and his tips as well as Eric Gilbertson's math skills helped me get through quals. I've also had the benefit of help from a number of UROPs, and I'd like to thank Roberto and Jasmine for helping with the VGR prototype, and Alex for taking time out of bigger and better things to do some quality editing for me.

As an undergraduate, I had a great time on the sailing team, and want to acknowledge Franny and the Matts for providing a welcome break from school (and also a great place to test marine robots!). I've been at MIT for long enough now to have met too many close friends to name, but you guys have helped me more than I can express, both in research and in life. However, Marguerite Siboni deserves recognition for helping this non-artistically-inclined engineer make a few decent-looking graphics.

Finally, no acknowledgments section is complete without thanking my family. My father Steve has always encouraged me to take things apart and learn the messy way. My grandpa Sam inspired me to be an engineer from a young age, while my mother Laurie has given me her gift of organization, and is always a source of encouragement. And I thank my brother Connor for his sense of humor.

Work was supported by Schlumberger-Doll Research Company, Award 016898-003, administered through the MIT Energy Initiative, and the Office of Naval Research, Grant N00014- 09-1-0700. This thesis is also the result of research partially supported by the MIT Sea Grant College Programs Doherty Professorship in Ocean Utilization.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The oceans cover 70% of the surface of our planet, yet are one of the final frontiers in terms of exploration and understanding. The interests and needs of ocean scientists and ocean-related industries have driven engineers to develop technologies that allow us to further study and utilize the ocean. The ocean environment is harsh, with extreme pressures, unknown currents, physical impediments to communication and navigation, and many other challenges for engineering reliable and useful systems. The oceanographic community, consisting largely of scientific researchers, the oil industry, and the navy, has made significant progress in underwater capability through the use of marine robotics and autonomy. However, most work to date has been focused on the capabilities of individual vehicles. As vehicle technology matures, large-scale fleets of vehicles can be deployed to create underwater infrastructure for research and industry, enabling more efficient operations in the ocean.

Two primary challenges for underwater operations are communication and navigation. Due to the severe attenuation of electromagnetic waves underwater, GPS navigation and radio frequency (RF) communications are not available underwater. Acoustic methods are regularly used for communication and geo-referenced navigation, which bring many constraints not typically faced on land or in air. Large and expensive acoustic navigation sensors such as Ultra Short Baseline Sonar (USBL) can be based on a surface ship and used to track vehicles underwater [81]. The mobility and convenience of this centralized navigation paradigm makes it attractive for op-

erations, however the fundamental limitations of the sensor bring challenges for use in multiple vehicle fleets. These vehicles may be physically different, have different onboard sensing and control, or be operating in regions of the ocean with differing characteristics. Efficient operations in the ocean drive the need for more productivity per unit ship time (ships can cost up to $500,000/day), and almost all missions underwater benefit from accurate navigation.

This thesis considers sensor management methods for multiple-vehicle deployment of autonomous marine vehicles that share a centralized navigation system. As an example, we consider the problem of subsea equipment delivery—the mission of delivering some payload to a desired location on the seafloor. In this mission as well as many general deployments of heterogeneous fleets of vehicles, the ship-based sensor is a constrained resource which much be effectively allocated among the members of the fleet.

1.1 Motivation and Background

1.1.1 Vehicle Operations in the Ocean

To give some context for this work, we will first cover some basics of underwater vehicle operations: common tasks, technical issues, as well as the vehicle platforms in use today.

Underwater Vehicle Tasks

Vehicles in the ocean have historically been used for a number of tasks. Equipped with various sensors, vehicles are often used to collect oceanographic data, such as salinity, temperature, dissolved oxygen, nitrates, fluorescence, and recently more advanced biological and chemical data such as DNA and mass spectrometry [113]. Additionally, vehicles are often equipped with water sampling capabilities in order to bring samples back for detailed analysis in the lab [22]. These data-collection and sampling tasks are sometimes performed in the mid-water column by vehicles, and are also conducted

at the seafloor using passive landers [109] or undersea observatories [49, 53]. Imaging is useful for documenting new discoveries of underwater life and seafloor formations, as well as documenting archaeological sites and categorizing marine life. Underwater imaging methods include sonar-based methods such as multibeam seafloor mapping, as well as vision-based methods, usually implemented with high power LED strobe arrays and still or video cameras [59]. Underwater vehicles are also used for intervention tasks, which can range from performing maintenance on oil pipelines to taking core samples from the seafloor. The final category of underwater vehicle tasks are specifically related to defense, such as ship hull inspection in harbors [58], surveillance, and mine countermeasures [42, 105].

Technical Scope: Common Challenges Encountered Underwater

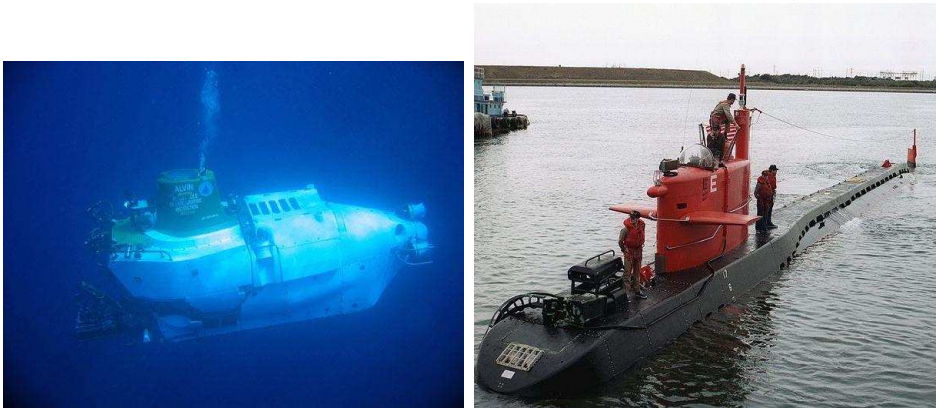
The underwater environment is harsh and unforgiving, and presents numerous challenges for underwater vehicles, including extreme pressures, corrosive saltwater, buoyancy, propulsion, communications, navigation and control. Battery life limits range, making propulsion a challenge for any autonomous vehicle, and thus propulsion efficiency is important. Propulsion underwater is almost exclusively accomplished by propellers, although buoyancy methods [103] and flapping foils [74] have also seen success in certain circumstances. Most vehicles use a combination of propeller thrusters and hydrofoil control surfaces to steer and maneuver.

Communications and navigation are especially difficult underwater, because the severe attenuation of electromagnetic waves in water means that traditional land and air based methods such as wireless RF communication and GPS do not work in the ocean. As will be described further in Sec. 1.1.3, acoustic methods are the primary means for both communication and navigation underwater. These constraints represent one of the primary challenges for advancing the capabilities of underwater vehicles today.

Platforms

Various classes of underwater vehicles exist, spanning the spectrum of size, capability and complexity. Small manned submersibles used for research and industry will be discussed; these submersibles are fundamentally different from large Navy submarines, which are not considered here. Unmanned underwater vehicles (UUVs) include two major classes of vehicles: Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs), which differ in whether the vehicle is tethered to a support ship, and also the amount of, and reliance on, human input to the vehicle.

Apart from military submarines, manned submersibles for research and industry use usually hold 2-10 passengers [65,88]. Manned research submersibles such as the US Navy-owned ALVIN, operated by the National Deep Submergence Facility (NDSF) at the Woods Hole Oceanographic Institution (WHOI), excel at tasks where human scientist firsthand accounts are important, and at intervention tasks such as sample collection, recovery of objects, and undersea repairs [16].

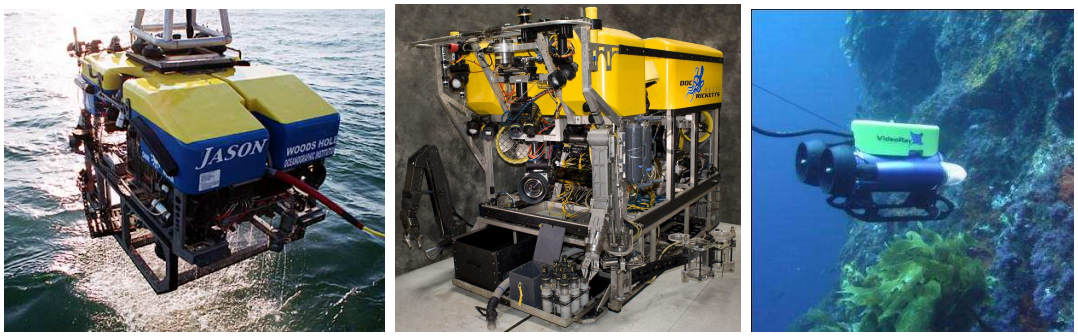


(a) (DSV) Alvin (b) NR-1

Figure 1-1: Manned research submarines. On the left is Alvin, operated by the Woods Hole Oceanographic Institution Deep Submergence Laboratory. On the right is the decommissioned US Navy submarine NR-1. Image credits: a) U.S. Navy photo [Public domain], via Wikimedia Commons, b) ⟨http://www.whoi.edu/page.do?pid=8422⟩

Remotely operated vehicles (ROVs) attempt to have the same capabilities as manned subs, but without the requirement of humans onboard. Instead, ROV pilots are onboard the support ship, where they have access to many different cameras and vehicle sensors allowing them to control the vehicle remotely. Power and data

transfer to the ROV is accomplished through the use of a long tether, which complicates vehicle dynamics but gives the vehicle unlimited endurance and much higher power and data bandwidth compared to AUVs. Various levels of manual versus automatic control exist, from full pilot control of the thrusters to highly capable autopilots that can hold station and servo visually off of features identified by the pilot in the vehicle's camera field of view [86, 107]. ROVs are the workhorses of the underwater vehicle community as their versatility allows them to perform many tasks. The drawbacks to ROVs are the necessary support infrastructure: the surface ship with the tether must remain with the vehicle at all times, and the entire setup can be expensive. ROVs can range from large, powerful work-class vehicles that are often found performing construction and maintenance in the oil and gas industry, to small portable inspection ROVs that can easily be deployed from a small boat [33].



(a) WHOI Jason II (b) MBARI Doc Ricketts (c) VideoRay

Figure 1-2: WHOI's Jason ROV (left) is purpose-built for oceanographic research. MBARI's Doc Ricketts ROV (center) is a modified commercial work class ROV by SMD. The VideoRay ROV (right) is a small inspection AUV. Image credits: a) ⟨http://www.whoi.edu/page.do?pid=8423⟩, b) ⟨http://www.mbari.org/dmo/vessels_vehicles/Doc_Ricketts/Doc_Ricketts.html⟩, c) ⟨http://www.molchanmarine.com/news/Default.shtm⟩

Autonomous underwater vehicles (AUVs) are vehicles that operate with independence from the support ship. Onboard computers execute missions without human input and the vehicle operates under its own battery power. AUVs are primarily suited to survey and monitoring tasks, where they often execute preplanned or adaptive missions to obtain oceanographic data, images or sonar-based maps. However, some AUVs have additional capabilities such as hovering [58] that allow them

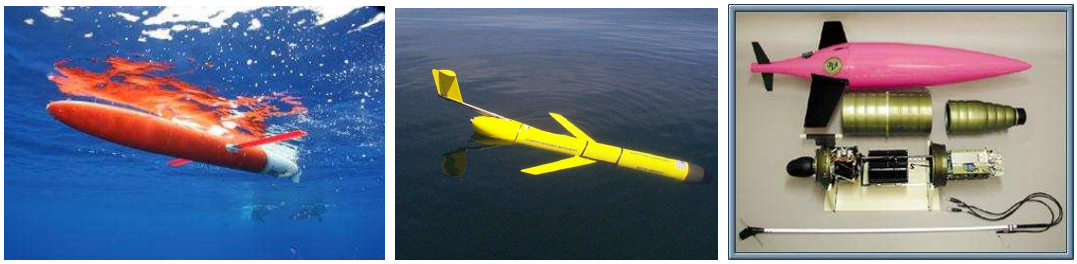
to perform tasks in more complex environments, as well as intervention tasks [75]. AUVs can vary in size from specialized large vehicles designed by the Navy [43] which can weigh up to ten tons, to the moderately sized but highly capable WHOI NDSF Sentry vehicle [114], to small 'man-portable' survey class AUVs that can easily be operated from a small boat, such as the Kongsberg/Hydroid REMUS 100 [12], or OceanServer Iver2 [39]. AUVs also have varying levels of autonomy, which will be discussed further in Sec. 1.1.4, but the basic capabilities include navigation, a low-level vehicle controller, and some sort of mission controller that executes high-level planning. AUVs rely on battery power which limits their range and endurance, and thus AUVs tend to be much more streamlined than ROVs.

(a) WHOI Sentry AUV (b) OceanServer Iver2 AUV (c) Bluefin-MIT HAUV

Figure 1-3: WHOI's Sentry AUV is used for mapping, imaging and sampling, and carries an extensive suite of scientific sensors. The OceanServer Iver2 AUV is a small commercially available survey class vehicle. The Bluefin-MIT Hovering AUV (HAUV) is a highly maneuverable vehicle used for ship hull inspection. Image credits: a) Chris German, Woods Hole Oceanographic Institution, b) ⟨http://www.naval-technology.com/contractors/electronic/oceanserver/oceanserver4.html⟩ c) ⟨http://oceanexplorer.noaa.gov/explorations/08auvfest/background/auvs/media/slideshow/gallery/08auvfest_album/large/hauv.jpg⟩

A specific type of AUV with a special means of propulsion is the underwater buoyancy glider. These vehicles do not have propellers, and instead move by adjusting buoyancy and gliding in a vertical yo-yo pattern using wings attached to their body. Due to their means of propulsion and direction of travel, to distinguish this vehicle from the Vertical Glider, to be discussed later, these gliders will be referred to as horizontal buoyancy gliders. In deep water especially, these gliders are effective at covering large distances in order to collect oceanographic data, although they move very slowly. Three successful designs for horizontal buoyancy gliders are the

Spray glider originally developed at Scripps Institute of Oceanography [90] and now produced by Bluefin Robotics, the Slocum glider originally developed through collaboration with WHOI [103] and now produced by Teledyne Webb Research, and the Seaglider vehicle developed at the University of Washington [44] and now produced by iRobot.



(a) Spray Glider (b) Slocum Glider (c) Seaglider

Figure 1-4: Three models of underwater buoyancy gliders in use today. These three models were originally developed at academic institutions, and have now been transferred to commercial products. Image credits: a) Robert Todd, Scripps Institute of Oceanography b) ⟨http://www2.sese.uwa.edu.au/~pattiara/slocum/⟩ c) ⟨http://www.apl.washington.edu/projects/seaglider/summary.html⟩

The final class of underwater vehicle that is relevant for subsea equipment delivery is the passive lander. These vehicles have very few capabilities and are a simple solution to the need to deliver sensors and other equipment to the seafloor. The landers are dropped above their desired location and fall passively without control, and thus suffer from drift. Current profile estimates can be used to account for drift, but passive landers are largely used in applications where a cheap simple solution is needed, and accurate placement is not required. Passive landers in the context of subsea equipment delivery are discussed more in Sec. 1.2.1.

1.1.2 Applications: Subsea Equipment Delivery

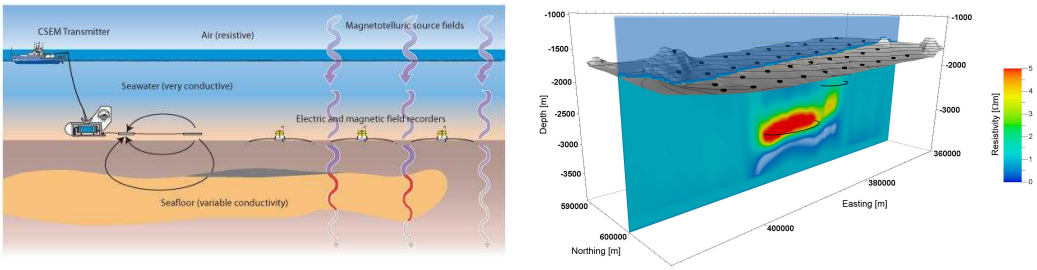
The task of subsea equipment delivery is useful for a variety of applications, including oil exploration, scientific monitoring, defense, and support infrastructure. In all of these scenarios, a payload (which in certain cases is integral to the vehicle itself), must be delivered to specific locations on the seafloor in the deep ocean. Scientific

applications of subsea equipment delivery include environmental monitoring [26, 37], such as placing chemical sensors next to a seep or measuring chemical interactions at the sediment-water interface [50], or geophysical applications including placement of seismic sensors in specific arrays [31, 56]. Various defense applications exist as well, such as defusing mines [98], or data collection/surveillance [38]. Additionally, subsea equipment delivery can be used to set up support infrastructure, ranging from acoustic network nodes [24, 56], collection baskets for deep-sea archaeology or other sampling tasks [17, 25] to underwater observatory docking equipment [57, 91, 99].

A specific application is oil exploration. The oil and gas industry would like to deploy electromagnetic sensors in a large precise grid at 4,000 m or deeper in order to map subsea rock formations [35]. This grid is on the order of a 7 km x 7 km square, with sensors every 1 km (49 total), as shown in Fig. 1-6. The method used is known as Controlled-Source Electromagnetics (CSEM), where an EM source (usually a dipole) is towed in the vicinity of the array, and the EM sensors in the grid pick up variations in the electric field caused by varying resistivity of different subsea materials (e.g. different types of rock, gas, or oil), as shown in Fig. 1-5. The 3-D reconstruction of the subsea formations from the sensor data relies on grid-based PDE reconstruction techniques, which perform better when the sensors are very accurately placed. Operationally, the fleet of CSEM receivers is deployed from a ship, the dipole is towed above the array, measurements are recorded onboard the receivers, and then the receivers are released and ascend back to the surface, where they are recovered [36]. This process is repeated in different locations, so it is desirable to have quick and accurate deployment of the system, as shown in Fig. 1-7.

Vertical Glider Robot (VGR) Concept

Subsea delivery is achieved with powered underwater vehicles (autonomous underwater or remotely-operated vehicles; AUV's or ROV's) or unguided landers; a full review of prior methods is given in Sec. 1.2.1. Powered vehicles can accomplish precision delivery with high performance because they can make repeated attempts to reach a given specification. But capital and operating costs of these vehicles can be



(a) CSEM Concepts (b) CSEM reconstruction

Figure 1-5: On the left is an overview of CSEM concepts. On the right is an example of grid-based CSEM reconstruction of subsea formations. Image credits: a) Scripps EM laboratory ⟨http://marineemlab.ucsd.edu/resources/concepts/CSEM_MT.html⟩ b) a) Electromagnetic Geoservices ⟨http://www.emgs.com/content/598/Modelling⟩

orders of magnitude larger than the cost of the sensor being deployed; in the case of many packages to be delivered, these costs and the risk to major assets may be too high.

Oceanographic researchers and the offshore oil and gas industry regularly use passively dropped landers to deploy sensors to full ocean depth of up to six kilometers. This is achieved by positioning the surface vessel so that predicted ocean currents cause the lander to free-fall to the desired target. Over the length of the drop, these landers accumulate significant drift; 1% of depth is a typical value reported in deep water when a good current measurement is made *a priori* (J. Guerrero, personal communication). Due to drift, passive landers sometimes have to be recovered so that another attempt can be made. In oil exploration, such as the example shown in Fig. 1-6, operating costs of the support vessel can be up to $500,000 per day, so precise and timely delivery of equipment is important. To reduce ship time and the associated costs, it is desired to allow all of the landers to be deployed from a single ship location near the center of the grid, with simultaneous or rapid sequential deployment to minimize the time needed to drop the entire fleet to the seafloor. The grid application also motivates the need for better horizontal transit capabilities than passive landers so that the vehicles headed to the outside of the grid can reach their targets from the central ship location.

To meet these challenges, we have been developing a unique system for this mis-

sion, which is aimed for multiple-vehicle deployment of equipment to be delivered to the seafloor. The individual lander vehicles are designed to be simple and economical, so the system is scalable, with the expensive components shared by the whole fleet. To keep cost and complexity low, we retain the free-falling lander concept that uses potential energy instead of a powered propulsion system. Building on the steerable elevator concept described in Sec. 1.2.1, we propose to add fully autonomous navigation and active control, and to streamline the vehicle in order to add horizontal transit capabilities as well as reduce the large drift forces from large-scale hydrodynamic separation. To distinguish our work from existing elevators and gliders as used in the ocean today, we refer to our device as the Vertical Glider Robot, or VGR. The VGR is designed to have its principal orientation nose-down, with negative buoyancy to provide a nominally constant dive rate. Most crucially, the vehicle is marginally stable in the open loop, allowing it to operate at extreme angles of attack and thereby move at glide angles greater than 60 degrees from vertical, satisfying the need for moderate horizontal transit capability.

The initial VGR system was developed in a previous thesis by C. Ambler [13], which included concept generation, hardware design and early control simulation for a single-vehicle system navigated by USBL. The work presented in this thesis aims to extend the single vehicle concept to multiple-vehicle fleet deployment, and considers the associated navigation and control problems that arise.

1.1.3 Underwater Communication and Navigation

Currents and drift due to hydrodynamic disturbances make navigation important—the vehicle cannot stop in one place to determine its location as is often the case on land. Communication is necessary for data transfer, mission commands, and is an integral component of distributed navigation systems. Consistent navigation and communication in the underwater environment is a perennial challenge because of our reliance on the acoustic channel [96] (due to the attenuation of electromagnetic waves in seawater, methods such as GPS and RF communication do not work underwater). The acoustic channel underwater is notoriously difficult, and subject to delays, fading,

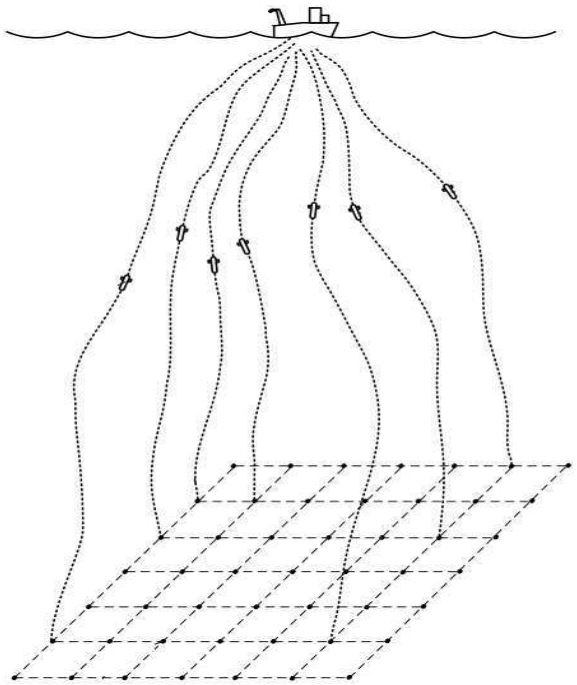


Figure 1-6: Overview of a Vertical Glider mission scenario to deploy 49 pieces of equipment on a 7 km x 7 km grid on the seafloor from a single ship on the surface.

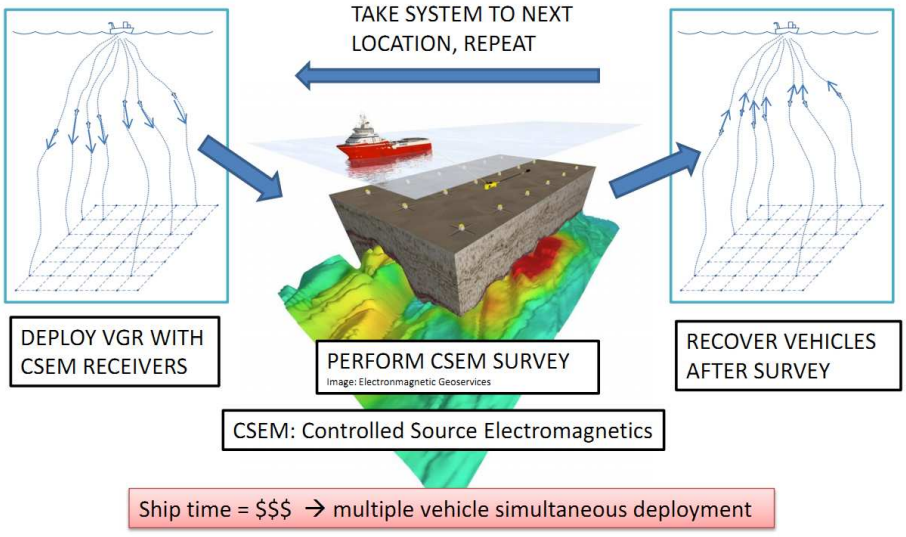


Figure 1-7: Vertical Glider operation cycle for oil prospecting with CSEM methods. The vehicles are deployed in the accurate grid, the survey is performed, and the vehicles are recovered. The ship drives to the next location and the process is repeated. Image credit: Electromagnetic Geoservices ⟨http://www.emgs.com/content/595/Data-acquisition⟩

frequency-dependent path loss, non-Gaussian noise, and multipaths, which all provide challenges and constraints to underwater navigation and communication. Some non-acoustic methods exist, however as will be discussed they have their drawbacks as well.

Non-acoustic Navigation Methods

Depth, magnetic heading, and orientation are relatively easily obtained underwater in the open ocean, however methods for accurately determining geo-referenced position are challenging. The most crude navigation involves dead reckoning based off of a compass and some sort of speed measurement such as counting prop turns, or some other open-loop model. These methods are not very accurate due to drift and no sensing of position. However, more advanced odometry-based navigation can be quite accurate. Navigation systems relying on inertial measurement units (IMU) and Doppler velocimetry (DVL) are frequently used in the underwater environment [64]. These systems have been reported to give sub-meter navigational accuracy, and also work well when combined with low frequency updates from a global navigation system (such as the acoustic methods described in the next section). However, these systems have significant drawbacks. A high-end IMU costs $150,000, while a DVL costs $30,000 or more depending on depth-rating, and Doppler velocimetry is only useful within range of a solid boundary. DVL bottom-lock range is frequency-dependent and is inversely proportional to the accuracy of measured velocities. Normally this range is on the order of tens of meters, although there have been some recent developments advertising 500m range [8]. As with very high-end IMUs, these units are prohibitively expensive and large in size for use in small, economical AUVs. Price and form factor aside, inertial and Doppler methods suffer from drift over time—errors accumulate as acceleration and velocity are integrated to give position. The latest high performance inertial and Doppler methods have drift rates as low as 0.1% of distance travelled, a 'good' system could have drift on the order of 0.5%, and obviously, as cheaper and smaller components are used, performance degrades further.

Drift-free Acoustic Navigation

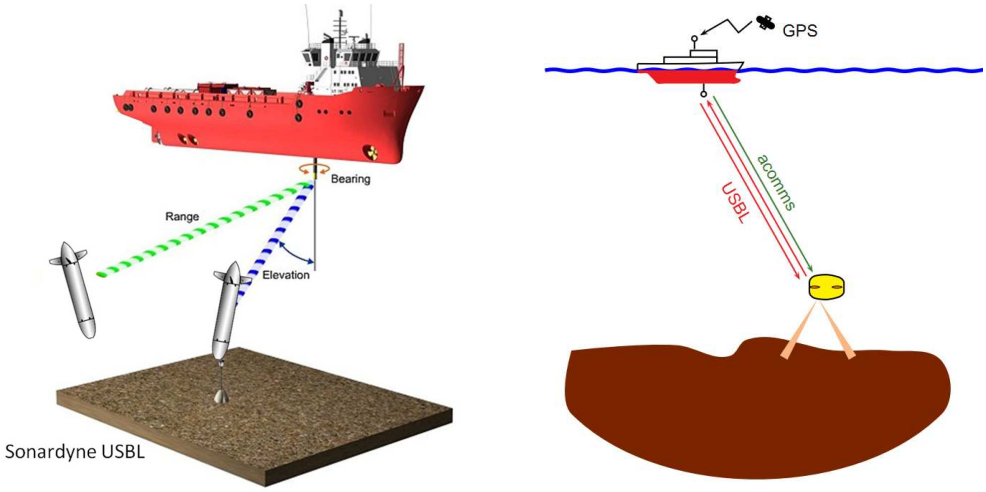
Acoustics can provide GPS-like drift-free globally referenced navigation underwater, albeit with other limitations. There are two main classes of acoustic navigation underwater that provide drift-free global reference: Long baseline (LBL) [77] and Ultra-short baseline (USBL) [101]. These systems use the travel time of sound in water to determine distance and therefore track acoustic pingers.

Long-Baseline Sonar (LBL) LBL systems include a GPS-like array of acoustic nodes, which are usually set up on the seafloor, separated by distances on the order of 100 m to 5 km for conventional 12 kHz systems. In most configurations the vehicle uses a pinger to send sonar ping to the beacons, which then respond with a return ping. Two-way travel times from the vehicle to the beacons obtains estimates of the distances from each beacon to the vehicle, which are used along with a precise survey of beacon locations to trilaterate the vehicle's position. If accurately synchronized clocks are used, one-way travel times can be used which reduces the delays involved and increases the update rate [104]. Because of the seafloor deployment and large spacing between beacons, the performance of LBL systems is largely depth-independent. With special care in the protocol, LBL systems can also be adapted to be used with multiple vehicles [46]. Additionally, there has been work with 'moving LBL,' or 'GPS intelligent buoys (GIB),' where the beacons are on moving platforms, usually autonomous surface crafts or buoys, equipped with GPS [11,41]. The moving LBL beacons sends down their locations along with the ranging ping, and with that information the vehicle can determine its location. While promising, moving LBL systems have not seen widespread adoption in ocean operations, likely due to the complicated infrastructure needed for multiple surface craft deployment from a large research vessel, as well as seaworthiness concerns. While accurate, conventional static LBL systems are not well suited to portable operations. Due to the large amount of time invested in setting up and calibrating the network of LBL beacons, LBL systems today are usually set up only when operations will be performed in the same area for a long time, such as multiple days.

Ultra-Short Baseline Sonar (USBL) In contrast to LBL systems, USBL systems use a single transceiver mounted below a support ship (with GPS and an IMU) that has multiple transducers in a compact (baseline on the order of 10 cm) array [3, 4, 6], shown in Fig. 1-8. First, an 'interrogation ping' is sent to the vehicle whose position is to be measured. This vehicle then sends a return ping to the ship transceiver. The travel times of the return ping from the measured pinger to the ship give the range, and the arrival times of the return ping at the different transducers in the array are compared using phase-differencing techniques to determine the direction of the return ping. The receiver includes an inertial measurement unit, and is also integrated with the ship's dynamic positioning system and GPS. This installation (not always permanent on ships) must be well-calibrated, however it usually represents a more convenient solution than deploying an LBL network. The direction and distance from the USBL unit are able to give precise 3D measurements in a globally-referenced Cartesian frame. The position can then be sent down to the vehicle in the next ping using an acoustic modem (often integrated into the USBL unit).

One aspect of USBL systems that demands special attention is their angular error characteristic, which will lead to a linear increase of noise on the Cartesian space estimate as slant range increases. Additionally, for LBL and USBL, position updates are delayed many seconds as components move apart; the speed of sound in water is around 1500 m/s. Recent advances such as ping stacking [6] allow for consistent updates at 1 Hz with USBL systems, and delayed-state filtering methods can help alleviate the additional error due to the age of the measurement when the update finally reaches the vehicle (after up to three trips along the slant range from ship transceiver to vehicle) [94]. However, due to the limited frequency band available for effective acoustic communication, current acoustic communication and navigation systems can only receive signals one-at-a-time. There have been some recent developments with both code and frequency-based multiplexing that allow for multiple vehicle tracking [6], but due to the limited frequency bands that can be used underwater, the number of vehicles that can be tracked at once is still much smaller than the overall size of the fleet of vehicles which are to be deployed. The extreme

difficulties and constraints of underwater acoustics suggest that these constraints will remain restrictive for the foreseeable future, especially as vehicle technology matures and fleet sizes grow.



(a) USBL Measurement (b) USBL for Control

Figure 1-8: In (a), a USBL unit mounted on a ship, measuring range, bearing and elevation [6]. Image modified from ⟨http://www.sonardyne.co.uk/Products/PositioningNavigation/systems/fusion_usbl.html⟩. In (b), cartoon showing the use of USBL for real-time control. Position updates are sent back down to the vehicle via acoustic modem (acomms). Image courtesy M. J. Stanway

The most effective underwater navigation is achieved using drift-free acoustic systems combined with IMUs and DVLs to achieve accuracy on the order of one meter [64, 69, 85, 106]. With multiple-vehicle fleets, collaborative navigation using inter-vehicle ranging can help improve position estimation accuracy [47], but is still subject to drift due to clock drift, and requires specialized equipment and processing onboard each vehicle (as well as time/frequency allocation in the network multiple-access scheme—see next section).

However, as mentioned previously, IMUs and DVLs are expensive, and LBL systems are time-consuming to deploy and calibrate. USBL systems have the most expensive component (the transceiver) mounted onboard the ship, and only require a small pinger coupled with an acoustic modem onboard each vehicle. Thus, the USBL represents the easiest single means for maintaining drift-free global reference with low infrastructure onboard the individual vehicles—well-suited for large-scale

multiple-vehicle deployments with short times onsite (seafloor observatories can also easily support multiple vehicles through permanent LBL installations). For vehicles with more advanced onboard navigation, addition of the USBL to the system can improve navigation further, and help bound errors due to drift. For the purposes of this thesis, we will focus on a sensing mode akin to a ship-mounted USBL, with limited sensors and navigation capability onboard the vehicle.

Underwater Communications

Radio-frequency wireless communications, the workhorse of terrestrial systems, are infeasible underwater due to severe attenuation. Attenuation is less dramatic at low frequencies, however systems running as low as 433 MHz have only been reported to propagate just over one meter underwater [10]. Transmissions at extra low frequencies (30-300 Hz) can propagate through conductive seawater, and are commonly used for communications by US Navy submarines [55], however transmission at these frequency bands requires large antennas and high power, making it impractical for use by small autonomous vehicles. Optical communications using lasers or LEDs have also been considered for high-bandwidth underwater communications [70] and can offer high throughput in certain conditions (several Mbits/sec at ranges up to 100m [48]), however optical links are affected by high scattering due to particles in the water and can require high precision in directionality, making them infeasible for most general applications underwater.

Similarly to navigation, underwater communications are primarily accomplished through acoustic links. Various technologies exist for acoustic modems, usually operating in the 10-30 kHz range. Performance of acoustic modems varies significantly based on the modulation type used and the channel characteristics. Frequency shift keying is a simple noncoherent modulation technique which is relatively reliable and low-power, but offers low communication throughput. Phase-shift keying (PSK) is a more complex coherent modulation method that requires more processing, is more fragile, but offers the possibility of orders of magnitude higher throughput [96]. Channel characteristics can vary in different ocean applications based on the water depth,

bottom topography, oceanographic water properties, sea surface conditions, ambient noise, and the direction of communication [97]. Deep water vertical channels offer the best conditions for acoustic communication due to low ambient noise and scattering in the mid-water column, less difficulty with multipaths, and lower variance on delays. The shallow water channel is much more difficult due to multipaths from surface and bottom effects, high delay spreads, and a high Doppler spread [10]. A rough performance limitation for vertical channels in deep water is 100 $km \cdot kbps$ for the range-rate product [63], while in shallow horizontal channels achievable bandwidths can be as low as 80 bps, and sometimes channel availability can completely vanish for tens of minutes [79]. Recent work has focused on signal processing (multiple input-multiple output channel estimation and spread-spectrum techniques for improving the performance of phase-coherent methods) as well as research into multiple access protocols and network routing for acoustic communication networks [32, 34].

There are a number of commercial off-the-shelf acoustic modems available [9], such as the WHOI micromodem [52], models by Teledyne Benthos [7], LinkQuest [5], EvoLogics [2] and DSPComm [1]. Additionally, USBL navigation units include acoustic modem capabilities integrated into the transceiver and transponders, such as with the Sonardyne Ranger USBL system used with the NDSF vehicle Sentry [6]. These USBL units support transmission of position data obtained by the USBL interleaved with short data or control packets.

Due to collisions of acoustic packets at the receiver, great care must be taken with acoustic modem systems if communications with multiple nodes must be achieved. Research is being conducted with multiple access (MAC) schemes, however the most widely used method in practice is simple Time Division Multiple Access (TDMA), where a time slot is allocated for each transponder to communicate. This approach obviously scales poorly as the number of vehicles rises. As mentioned in the context of USBL transducers, frequency or code based multiple access form the basis of other MAC schemes (FDMA or CDMA); but while possibly offering benefits over TDMA, these schemes do not eliminate the multiple-access problem as fleets become large and frequent communications are required [40].

1.1.4 Underwater Autonomy

Autonomy underwater in general is similar to the autonomy required by land robots, with special consideration of the unique navigational and communication constraints encountered in the ocean, as well as the specific requirements of the mission goals. Sensor fusion and state estimation are usually accomplished by conventional Kalman Filter or Extended Kalman Filter implementations, and certain underwater applications have successfully used Simultaneous Localization and Mapping (SLAM) [45]. Onboard flight control is developed and tuned specific to the vehicle design, and ranges from simple PID controllers to highly nonlinear MIMO control systems for vehicles with complex dynamics. Above the low-level controller there is some form of an autonomous decision-maker. This software ranges from simple modules that execute preplanned missions (for example, visiting a series of waypoints), to powerful adaptive mission planners running onboard artificial intelligence algorithms [19, 76]. Additionally, due to acoustic links to a ship, many AUVs rely on some aspect of human-in-the-loop decision making for low-frequency high-level planning, leveraging the economical mobility and data-gathering capabilities of the AUV combined with the experience and knowledge of human scientists [21, 89, 112].

1.2 Prior Work

1.2.1 Hardware For Subsea Equipment Delivery

There has been work using AUVs or ROVs to deploy equipment on the seafloor, as well as ROV deployment of benthic lander vehicles for oil operations monitoring [27]. However, apart from specialized deployments requiring the specific maneuvering or manipulation capabilities of these complex vehicles, subsea equipment delivery is normally accomplished using passive landers. WHOI frequently uses passive elevator vehicles to support ROV and ALVIN sampling operations [25], landers have also been used to track fish using sonar [84], and the design of a vertical/horizontal AUV for deep ocean sampling was addressed in [29]. Passive landers have been used extensively

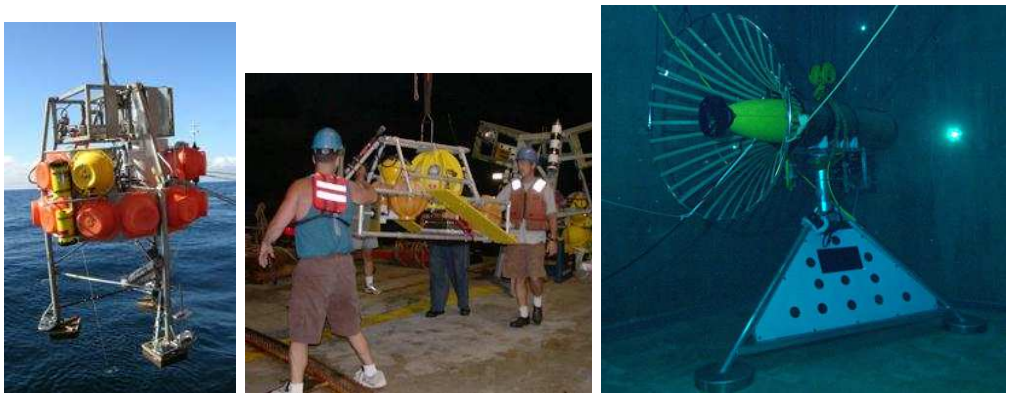
in the Autonomous Lander Instrument packages for Oceanographic Research (ALIPOR) Programme [82], which cites radii of hundreds of meters for accuracy of passive deployments. Our conversations with colleagues at Schlumberger-Doll Research (J. Guerrero, personal communication), have indicated that with *a priori* current profile measurements (e.g. from a ship-mounted ADCP), this can be reduced to roughly 50 m over 4,000 m descents.

To address the poor accuracy when using passive landers, there has been some prior work on steerable elevators at WHOI (D. Yoerger and A. Bradley, personal communication). These elevators consisted of passive elevator frames retrofitted with wings, which spiraled down in a helix trajectory and could be steered manually in a rough manner via a single rudder and an acoustic link to the surface ship. While this project was sidelined in the early 2000's, there has been some recent work with model tests of steerable elevators [87], which focused on the glide angle capabilities when a conventional elevator was outfitted with small angled lifting surfaces and did not address the problems of automatic control, navigation, or multiple-vehicle deployment.

There has been considerable effort to address the similar problem of terminal guidance of AUVs. In addition to the AUV docking systems described in Sec. 1.1.2, docking of AUVs is considered using visual servoing in [72], and delivery of a fiber optic communications cable to an undersea node via optical terminal guidance is discussed in [38]. Control strategies for terminal guidance of an underactuated AUV using a nose-mounted USBL homing to a beacon are considered in [18], while [42] uses a similar approach for mine countermeasures.

1.2.2 Multiple-Vehicle Navigation and Sensor Management

Even with USBL technology there are still many challenges for multiple vehicle deployments. Currently, USBL systems are rarely used to measure more than one vehicle during a mission. Moving to multiple vehicle fleets presents challenges—for example, with 50 vehicles using a simple round-robin scheme, each individual vehicle will receive a measurement update every 50 seconds. Combined with the increasing



(a) IFM Geomar Lander (b) WHOI Steerable Elevator (c) MBARI AUV Docking vehicle

Figure 1-9: Prior hardware for subsea equipment delivery. On the left is the IFM Geomar lander, a passive lander that is part of the ALIPOR Programme. In the middle is a WHOI elevator modified to be steerable, for use with ALVIN and JASON. On the right is the MBARI AUV docking system. Image credits: a) ⟨http://www.ifm-geomar.de/index.php?id=1200&L=1⟩ b) ⟨http://www.whoi.edu/atlantis117/feb8.html⟩ c) ⟨http://www.mbari.org/auv/dockingvehicle.htm⟩

noise and delays with depth, this will not result in good landing error performance.

This problem could be approached a few ways - the first would be to improve the hardware capabilities of the USBL itself, the second is to use inter-vehicle communication and ranging, and the third is measurement allocation algorithms using existing USBL technology. It is theoretically possible to devise methods to compute positions from multiple returns at once. However, current commercially available USBL systems are still limited to 1 measurement every second, and larger fleet sizes will always be desired, so as we look ahead towards advances in sensor technology, the measurement allocation constraints we have posed will be still be relevant. There has been considerable work recently in the fields of multiple vehicle collaborative control and coordination [15, 61, 73]. These methods require that the vehicles have methods of communicating and ranging with each other, such as with an acoustic modem, and various algorithms exist for improving the position estimates of the entire fleet of vehicles based on inter-vehicle range measurements. However, while these methods are promising if the absolute best accuracy is desired, they do not fit well into the simple and scalable goal for the VGR fleet, and USBL updates can still help

bound the error due to drift over time. Inter-vehicle communication and navigation adds another layer of complexity and cost to the individual vehicles, as well as a considerable amount of network protocol design and computational overhead to the navigation system. The end-users of the VGR have indicated that a simple drop-in solution where the number of vehicles can easily be changed is desirable. Thus, we have chosen to focus on navigation using only simple onboard instruments augmented by global position updates from a single USBL on the ship. The USBL represents a single highly constrained sensing resource that must be allocated in a smart manner in order to give the best navigation results for the entire fleet. This brings us to the problem of effective scheduling of measurements from the USBL to different vehicles in the fleet, a problem belonging to the field known as sensor management.

In practice, the current state-of-the-art for marine systems involves simple heuristic methods, the most common being a basic round-robin scheme where every vehicle is measured equally often in a periodic manner [28]. However, as fleets become larger and the desired performance and task complexity increases, better approaches are needed. There has been considerable work from the control theory and operations research community in algorithms for sensor management, often tightly coupled with tracking and estimation problems. The 'information state' or 'reward' in the setup of vehicle tracking problems is usually defined as the state estimation error uncertainty. Thus, state estimation is an integral component of any smart tracking system. As described in Sec. 1.1.4, state estimation is a crucial component for onboard control systems; and the extension to the target tracking case simply requires the decision-making module to run a state estimator for the relevant states of all of the vehicles in the fleet. For the purpose of this thesis, the standard linear Kalman Filter will be used, although other approaches are possible. The standard implementation of the Kalman Filter assumes complete observation updates at each time step, however for the target tracking problem, this will not be the case as the sensor may only observe one vehicle each time step. Kalman filtering is often used in practice with intermittent observations, which uses the simple intuitive result that the optimal method for handling missed measurements is to propagate the prediction updates open-loop (no

update or innovation due to zero Kalman gain through setting the measurement noise to infinity) when no measurement is available [92].

A greedy heuristic based on the tracking error uncertainty is the first step towards an algorithm smarter than a standard round-robin. However, optimization-based methods that are 'non-myopic' have the potential to leverage better vehicle, sensor and environmental models in order to best utilize limited sensing resources. Maximizing the utility of measurements for an underlying detection or estimation problem can be addressed by brute-force enumeration of scheduling policies for very small problems, and can also be formulated as a dynamic program. However for large fleets (large or infinite state spaces) and many decision epochs, the curse of dimensionality makes traditional decision-making approaches computationally intractable. Due to the large scale and difficulty of these problems, greedy or myopic approaches are commonly used and have seen success, as well as possess some performance bounds. However, non-myopic information-theoretic approaches are theoretically more elegant and offer promise for better performance in many cases, especially where special problem structures can be exploited [110].

A general sensor management problem is considered in [30], involving classification of multiple unknown objects. Using an approximate dynamic programming approach, this work formulates the resource management problem as a constrained dynamic program, and solves the Lagrangian relaxation optimally. Solution is through standard partially-observable Markov decision process (POMDP) algorithms, which puts significant constraints on the size of the state spaces that may be considered. In [111], the authors consider the problem of tracking multiple targets with a single steerable sensor, such as phased array radar. The sensor constraints are very similar to the USBL model—only one target can be observed at a time, and the multiple target processes are evolving independently and dynamically. The problem formulation is limited by the assumption that the vehicles have identical dynamics, however vehicles may have heterogeneous process and sensor noise models. The authors formulate a general stochastic estimation problem and use an auction approach to solve the open-loop feedback control problem optimally over a constrained set of policies. A

Bayesian mutual information method is used to incorporate new measurements; any prior distribution (measurement model) is possible. This is a distinct advantage over classical Kalman Filter based methods such as those in [71], which assume Gaussian noise distributions. A finite planning horizon is considered, within which each target may only be measured once, which allows for tractable computation of the combinatorial optimization problem. This constraint presents significant limitations as it cannot handle targets with vastly different characteristics (an example could be a case where one vehicle requires significantly more updates than others due to very high noise). Interestingly, this work includes a bound that says a greedy measurement allocation policy is guaranteed to be within a factor of 2 of the optimal sequence.

Much of the work in non-myopic sensor management relies on the fundamental notion of submodularity, an intuitive property of diminishing returns [66], which can be used practically to design algorithms as well as to derive performance bounds [67]. The most basic explanation of submodularity is that adding a sensor to a small deployment helps more than adding a sensor to a large deployment, or taking a measurement of a vehicle with high uncertainty helps more than measuring a vehicle with low uncertainty. More specifically, submodularity is a notion similar to convexity, but for set functions. The property of submodularity is inherent in a special class of problems, known as the Multi-Armed Bandit (MAB) problem, which is especially promising for vehicle tracking with constrained sensing resources. This field will be explained in more depth in Chapter 4, but briefly, bandit problems involve a situation where the goal is to make sequential decisions between a number of choices in order to maximize some cumulative reward. Information, or a model, of the process evolving is used to inform the decision-maker, however the decision made at each time step influences the new information that becomes available after the decision is made. In a sensor management problem, the cumulative reward in question is some metric of desirable tracking performance, and the decision to be made each time step is which vehicle to measure. A seminal paper by Gittins in 1974 [54] demonstrated that the MAB can be solved with a series of one-dimensional problems using a priority index policy—an index can be computed for each process which represents the intrinsic value

of observing that process, taking expected current and future rewards into account. Then the process with the highest index value is chosen for measurement at the given time step.

An early attempt to address the sensor management problem via the Gittins index is found in [68], where the problem is to find an optimal solution to tracking multiple independent objects using a Hidden Markov Model. The application is radar beam scheduling, however the use of the standard (passively static) MAB requires the inappropriate assumption that the states of the targets do not change when they are not being observed. In [80], the authors study a slightly different situation than vehicle tracking but attempt to study the effects of unknown dynamics within the bandit framework. The reward is a time-varying linear function of the covariate vector of each system, and the system dynamics are unknown. The covariates and consequences of actions are observed, so the goal is to learn the association between actions and covariates. A more relevant study of the dynamic target tracking problem in the MAB framework is given in [102], which evaluates round-robin, myopic and MAB approaches to the tracking of Brownian motion targets. Although the analysis is not comprehensive, the authors find that in many situations the classical MAB gives a good suboptimal solution even when some of its assumptions are violated.

An extension of the MAB problem known as the Restless Bandit problem [108] extends the structure to cases where the system evolves whether a decision is made or not. Since a vehicle is still moving (affected by control and/or process noise) whether a measurement is taken or not, this scenario describes the VGR measurement problem well. The MAB and Restless Bandits, as well as the Restless Bandit Kalman Filter scheduling algorithm in [71] are described in detail in Chapter 4.

1.3 Summary and Objectives

The goal of this thesis is to study methods for multiple-vehicle deployment of autonomous vehicles using a constrained, centralized sensing resource for global navigation, primarily focusing on non-myopic sensor management methods for allocating navigation hits among vehicles with different noise or dynamic characteristics. As a specific case-study we will consider the subsea equipment delivery mission described earlier, and briefly discuss development of a model scale prototype Vertical Glider vehicle which serves as a proof-of-concept for a scalable multiple-vehicle deployment application in the deep ocean. Experimental tests of this prototype are presented in Chapter 2. We focus on tracking large fleets of vehicles using a USBL-like sensor mounted centrally on a ship, which can measure one vehicle at a time at a finite update rate. In Chapter 3, we discuss multiple-vehicle operations in the ocean using this navigation method, focusing on the system architecture and problem formulation for Kalman Filter-based multiple vehicle tracking. We develop simple models for two mission scenarios which are suitable for use in the tracking algorithms we investigate in Chapters 4 and 5. We consider USBL augmented navigation for vehicles with two commonly used onboard sensor suites: onboard compass and attitude sensors, and vehicles equipped with a DVL. In Chapter 4 we give a tutorial of the Multi-Armed Bandit problem, its applicability to multiple vehicle tracking, as well as an extended explanation of Restless Bandits and the Scheduling Kalman Filters algorithm. In Chapter 5, we show the usefulness of these algorithms through computational experiments on examples with heterogeneous vehicle fleets, as well as the specific multiple-vehicle subsea equipment delivery application.

In summary, the objectives of this thesis are:

1. Describe a vehicle hardware concept suited to economical multiple vehicle deployment for subsea equipment delivery, and present experimental results from a single-vehicle prototype system. (Chapter 2)

2. Describe the high-level design of a multiple vehicle control system that uses centralized navigation from a single constrained sensor, develop simple models that are suitable for use in tracking algorithms for two onboard sensor suites, and formally state the multiple vehicle tracking problem (Chapter 3)

3. Investigate non-myopic algorithms for multiple-vehicle sensor management, and give an explanation of the Restless Bandit Kalman Filters (RBKF) scheduling algorithm. (Chapter 4)

4. Present computational results comparing the performance of non-myopic algorithms with commonly used heuristics. (Chapter 5)

Chapter 2

Prototype Vehicle

This chapter discusses work with a single vehicle prototype system suitable for economical large-scale multiple vehicle deployments. This work is a continuation of the work by C. Ambler, following initial concept generation for the Vertical Glider Robot for subsea equipment delivery. The goal of the prototype vehicle work is to demonstrate a proof-of-concept for vertical deployment of the VGR using surface-based navigation. The prototype vehicle system consists of the physical Vertical Glider prototype vehicle, which takes the form similar to traditional torpedo-shaped survey AUVs, however in a vertical orientation; as well as the navigation and control system and associated software. We give a brief overview of the physical vehicle design here; a more comprehensive description is given in [13]. Navigation and control methods are discussed, and experimental testing results in the MIT swimming pools are presented, which demonstrate the successful proof-of-concept.

2.1 Prototype Vehicle Physical Design

A prototype vehicle has been built to explore the behavior of vertically-oriented streamlined vehicles, including the effectiveness of control fins and achievable glide slopes. The vehicle has a simple, streamlined shape with control fins at the tail in the traditional cross configuration, as shown in Fig. fig:explodedview. Table 2.1 lists some of the vehicle's physical characteristics. For control and data logging purposes,

the vehicle's sensor suite includes an onboard tilt-compensated compass, pitch, and roll sensor (Ocean Server OS5000), a pressure sensor used to measure depth (Measurement Specialties M86), and angular rate gyros (Invensense IDG1250). An Arduino Mega microcontroller is used to read in sensors, compute control commands, drive servos and log data. Onboard data logging is handled by a 4D systems μDrive microSD data logger. We use the CMUCam3 camera system for global navigation.

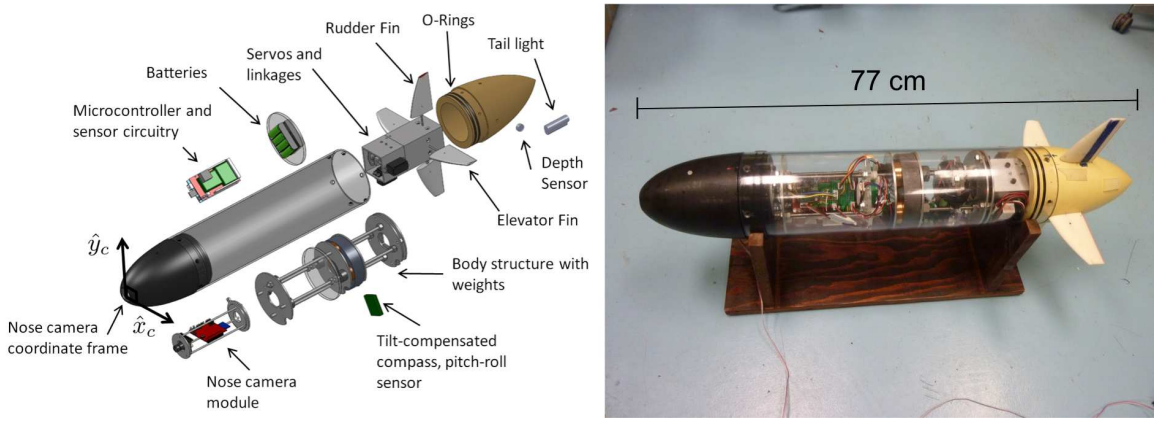


Figure 2-1: An exploded view of the vehicle is on the left, including the onboard camera reference frame $\hat{x}_c$ and $\hat{y}_c$. A photograph of the vehicle with the communication tether attached to the side of the nose is on the right. Note the large lead weight near the center of the vehicle which was located to place the center of mass very slightly below the center of buoyancy, resulting in a marginally stable vehicle that can fly at high angles of attack.

Table 2.1: Vertical Glider Physical Parameters

Length	77 cm
Diameter	12.7 cm body, 30 cm at tips of fins
Volume	8040 cm^3
Weight	8.05 kg
Weight in Water	98 g
Fin Profile	NACA-0020
Design Dive Rate	55 cm/s
Max Depth	5 meters
Servos	HiTec HS-322HD (x2)
Power Source	8xAA NiMH batteries (1.2 V each, 9.6 V total)

2.2 Prototype Vehicle Navigation and Control

2.2.1 Navigation Methods

A camera tracking system is used in pool testing to emulate angle-based tracking methods used in the ocean. Two major modes of operation using a camera are possible, as shown in Fig. 2-2. One mode consists of the camera mounted in the nose of the vehicle. A flashlight is placed on the bottom of the pool to serve as the target; the camera tracks the light and the control system guides the vehicle towards the target. This method is different than the proposed surface ship navigation using a USBL, but has obvious applications in missions such as docking or homing towards an existing target [91, 99]. This capability is completely self-contained within the vehicle.

The second mode of operation matches deployment with a USBL on a ship more closely. A light is placed on the tail of the vehicle, and a surface raft holds a camera that tracks the light. The error in vehicle position is computed on a connected laptop at the surface, and this is combined with heading and attitude information received from the vehicle through a 2 mm diameter tether to compute commands for the vehicle's control surfaces. Matlab software is used for communication, control and logging on the laptop.

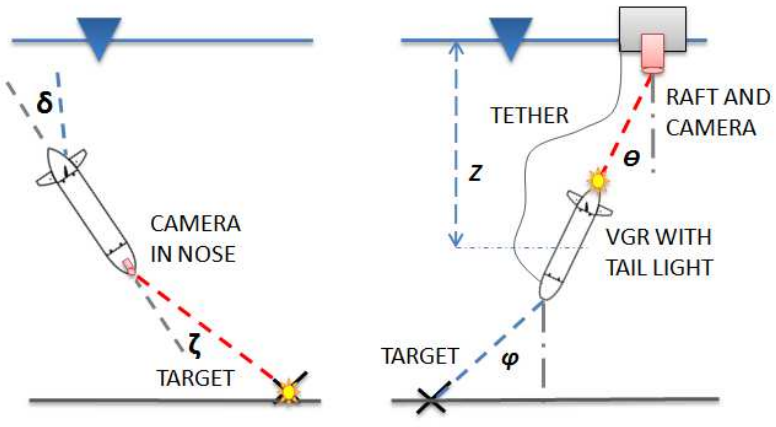


Figure 2-2: Prototype vehicle testing configurations. The nose camera configuration is on the left, and the surface camera configuration is on the right.

2.2.2 Flight Control with Onboard Camera

One primary advantage to the onboard camera is that flight control is very simple because measurement, control and actuation are all kept in the vehicle body-referenced frame. No information about the vehicle's orientation is needed for the controller. The elevators correct for errors in the camera's y axis, $\hat{y}_c$, and the rudders correct for errors in the x axis, $\hat{x}_c$, as diagrammed in Fig. fig:explodedview. A simple proportional controller maps the target location in the camera's field of view (x_c and y_c, measured in pixels) to fin commands, attempting to keep the target in the center of the camera's field of view:

$$\left\{ \begin{array}{c} \theta_{elevator} \\ \theta_{rudder} \end{array} \right\} = \left\{ \begin{array}{c} -Ky_c \\ -Kx_c \end{array} \right\} \tag{2.1}$$

If the camera loses the target, the fins are both held at their previous position, which in practice allows the vehicle to recover from large oscillations that cause the target to temporarily leave the field of view of the camera.

2.2.3 Flight Control with Surface Camera

The surface camera is located at the origin of a global North-East-Depth inertial coordinate frame, which is represented by $\hat{x}_g$, $\hat{y}_g$ and $\hat{z}_g$ in Fig. 2-3. We use the depth of the vehicle, z, and the camera target pixel locations to find the tail location in global coordinates: x_g and y_g. We subtract the target, x_g^{des} and y_g^{des}, from the tail's location in the global frame to get a global horizontal-plane error vector, e_g^x and e_g^y. The vehicle's body-referenced frame $\hat{x}_v$, $\hat{y}_v$ and $\hat{z}_v$ is aligned with $\hat{z}_g$ but is rotated in the horizontal plane by the vehicle's compass heading. The compass heading ψ is the angle of rotation of the body-referenced frame from magnetic North (set to equal $\hat{x}_g$ in Fig. 2-3), which is computed onboard the tilt-compensated compass sensor using data from magnetometers and accelerometers on all three axes. We transform the global error vector into a vehicle body-referenced error, e_v^x and e_v^y, through a rotation matrix that uses ψ:

$$\begin{Bmatrix} e_v^x \\ e_v^y \end{Bmatrix} = \begin{bmatrix} cos(\psi) & -sin(\psi) \\ sin(\psi) & cos(\psi) \end{bmatrix} \begin{Bmatrix} e_g^x \\ e_g^y \end{Bmatrix} \tag{2.2}$$

Vehicle pitch is a rotation about the vehicle's body-referenced x axis, $\hat{x}_v$, and is actuated by the elevators. Vehicle roll is a rotation about the vehicle's body-referenced y axis, $\hat{y}_v$, and is actuated by the rudders. Using the depth of the pool, D, the vehicle's current depth, z, and the vehicle body-referenced errors, angles to the target about the vehicle's x and y axes, θ_x and θ_y, are calculated:

$$\begin{Bmatrix} \theta_x \\ \theta_y \end{Bmatrix} = \begin{Bmatrix} atan(e_v^y/(D-z)) \\ atan(e_v^x/(D-z)) \end{Bmatrix} \tag{2.3}$$

Since the vehicle's pitch and roll dynamics are faster than its dynamics in the horizontal plane, a closed-loop pitch and roll controller commands the fins to angles $\theta_{elevator}$ and θ_{rudder} to attempt to drive the vehicle to the desired angle to the target, using proportional control with gain K:

$$\begin{Bmatrix} \theta_{elevator} \\ \theta_{rudder} \end{Bmatrix} = \begin{Bmatrix} -K(Pitch - \theta_x) \\ -K(Roll - \theta_y) \end{Bmatrix} \tag{2.4}$$

A block diagram of the distributed control system used for the surface camera tests is shown in Fig. 2-4.

2.3 Prototype Experiments in Pool

Testing was conducted in the MIT Alumni Pool (4m depth) and the MIT Z-Center Pool (4.25m depth).

2.3.1 Onboard Camera

We conducted several experimental runs to a flashlight target on the bottom of the swimming pool with the onboard camera configuration. A plot showing the camera's

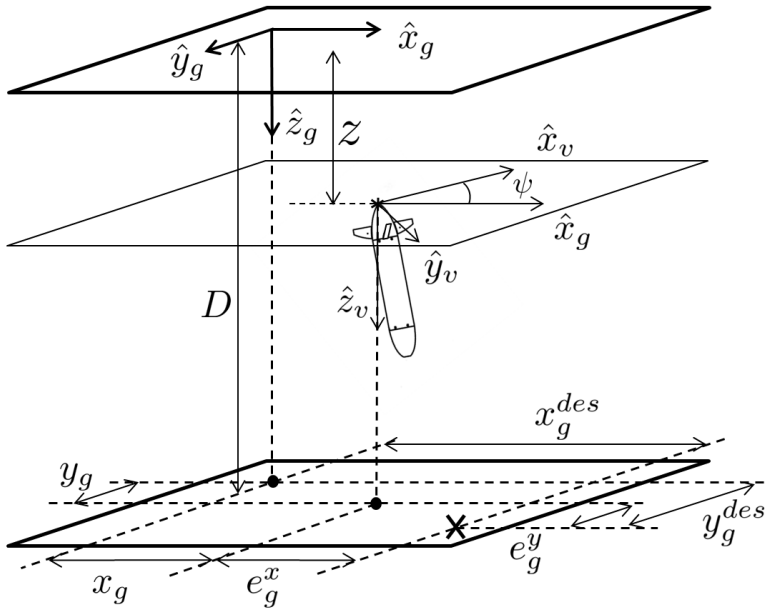


Figure 2-3: 3D coordinate frames used for flight control with surface camera. The global frame $\hat{x}_g$, $\hat{y}_g$, $\hat{z}_g$ is centered at the location of the surface raft. The body-referenced frame $\hat{x}_v$, $\hat{y}_v$ and $\hat{z}_v$ is aligned with $\hat{z}_g$ but is rotated in the horizontal plane by the vehicle's compass heading, ψ.

adjusted target over the course of a run is shown in Fig. 2-5. Starting from a variety of initial positions and angles, the vehicle hit the target within 25 cm 26 times and veered off course due to loss of the target in the camera field of view 3 times. The times when it veered off track were due to testing the limits of extreme initial conditions. During these closed-loop tests, we noted the vehicle was able to reach targets that required a trajectory of 45 degrees from the launching point. Detailed analysis of the onboard camera testing is discussed by C. Ambler in [13].

2.3.2 Surface Camera

To test the surface camera, we placed and surveyed a target on the bottom of the pool that was 3 ft directly to the East of the surface camera. To show the vehicle's control capabilities, we started the vehicle in different orientations – both the angles in the E-Z and N-Z planes and the rotation about the vehicle's axis. We observed some runs where the vehicle rotated a full 360 degrees about its primary axis, showing

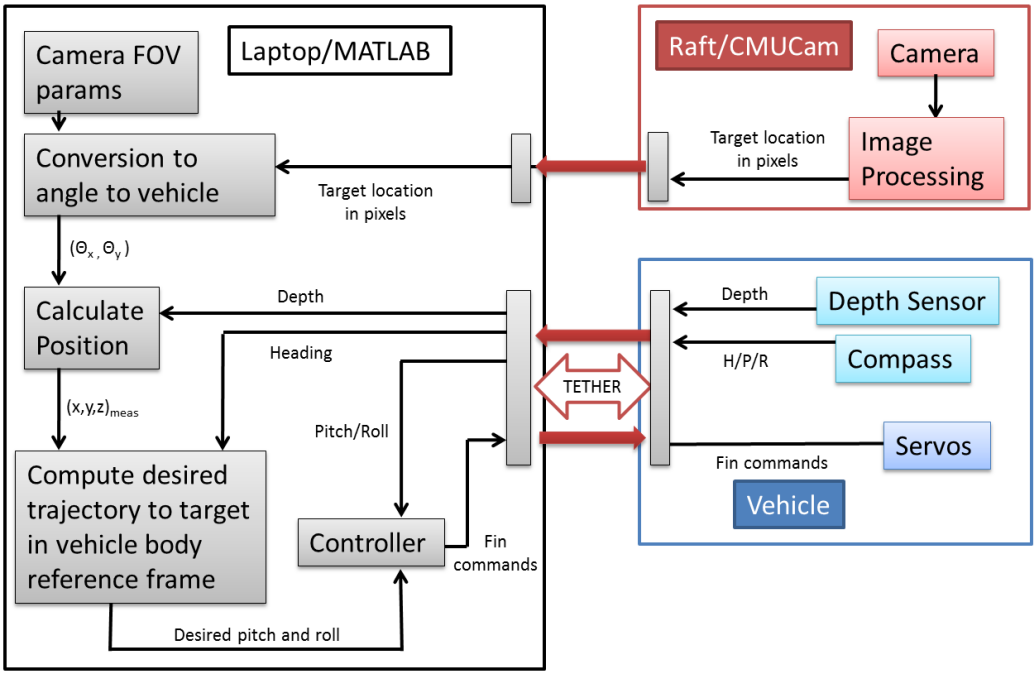


Figure 2-4: Control system block diagram for prototype vehicle with surface camera configuration

that our transformation from global to vehicle frame based on heading was working correctly.

Plots showing the vehicle's trajectory for three runs to the target with different initial conditions are shown in Fig. 2-6. The vehicle corrects for drift in the N-Z plane over the course of the run. The vehicle tracks the desired angle to the target in the E-Z plane well, but due to inaccuracies in the system and a simplified controller, it overshoots the target slightly, by an amount proportional to its initial angle towards the target.

One major limitation on this test was the camera's field of view. The CMUCam has a field of view of 49 degrees in x and 37 degrees in y, which limits the 'cone' in which the vehicle can be seen by the camera. USBL systems in the ocean also have a limited cone of detection, due to attenuation of the signal to reduce noise from the ship machinery at shallow angles. While the CMUCam's field of view is a tighter constraint than typical USBL detection cones, we were able to learn about the effects of this constraint on operations through our testing. The limited cone means that

we could not command the vehicle to go to targets very far away, and the margin for testing initial vehicle orientations was limited.

Additionally, the surface raft that holds the camera was designed to resist wave disturbances; however, some pitch and roll oscillations were observed that added noise onto the measurements. Adding a pitch and roll sensor to the raft could remove this noise, just as is done with a real USBL system on a ship. Regarding the control system, the vehicle had some backlash and calibration errors on the fins, which can add errors. For the tests shown, the controller computes control actions based off the position of the light at the tail, not the vehicle's center of gravity (CG). This introduces angular error and accentuates nonminimum phase aspects of the measured system. An improved controller would account for the difference between the measurement and vehicle's CG and also attempt to drive the vehicle directly over the target first, and then drop straight down. These issues were ignored for our initial tests, and explain some of the overshoot observed in the results.

2.4 Summary

We have shown a physical prototype vehicle and navigation system that demonstrates the concept of a vertically-oriented vehicle with no thrusters and active steering using a terminal guidance system. This vehicle serves as a proxy for an ocean vehicle navigated by a USBL on a ship. Results show that control can be used to guide the vehicle towards a target on the bottom using only basic onboard sensors (depth, heading and attitude) and a position sensor at the surface. This navigation method results in economical individual vehicles, enabling operations with large fleets. This thesis is focused towards multiple-vehicle deployment, and due to the pool constraints as well as the complexity of testing multiples of this prototype vehicle, the decision was made to terminate the physical VGR prototype testing at this stage in order to focus on the multiple vehicle sensor management methods, which will be the subject of the remaining portions of this thesis. Future experimental work will utilize a multiple surface raft testbed currently in development.

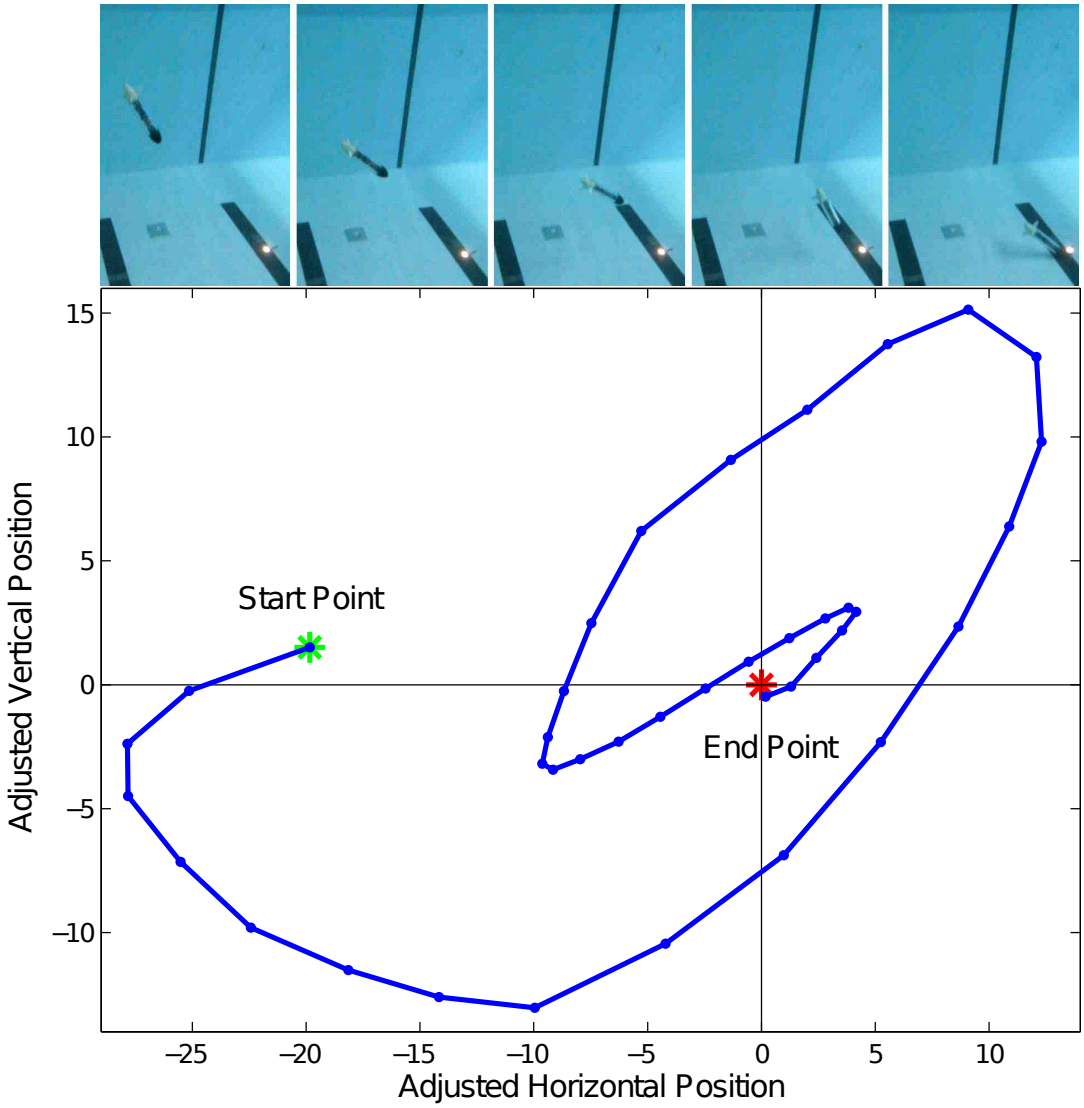


Figure 2-5: Scatter plot showing the adjusted target location as seen by the vehicle's onboard camera during pool testing. For this plot, the camera's output in pixels is scaled by the radius of the target as seen by the camera, which adjusts for the angle-accentuating effects of the vehicle's distance to the target. The vehicle was launched from a point 3 m horizontally away from the target.

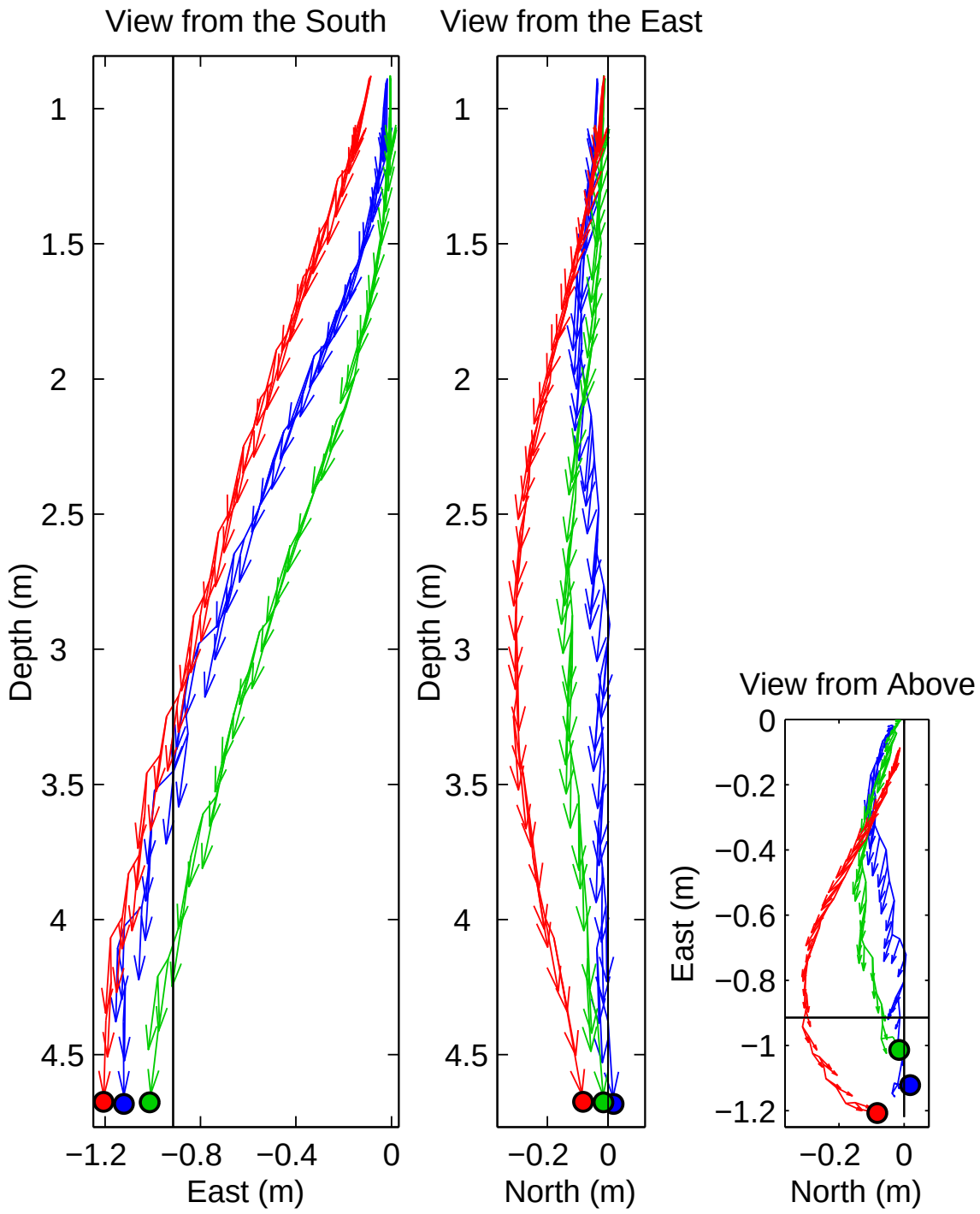


Figure 2-6: Trajectory results from pool experiments. The target was 0.9 m directly to the West, as shown by the black lines.

Chapter 3

Multiple Vehicle Sensor Management

Having shown a proof-of-concept of a model scale vehicle hardware platform that is suitable for scalable multi-vehicle operations, the focus now shifts to techniques for deploying large fleets of autonomous agents, which is seen as one of the the next big steps in advancing autonomous capability in the ocean. Due to the extreme challenges of navigation and communication underwater, acoustic methods are a primary enabling technology, and present some unique constraints for multiple-vehicle fleets. This chapter will outline the system architecture for multiple-vehicle deployment of AUVs using a centralized global navigation system. We explain the control loops, the division of capabilities between the individual vehicles and the support ship, and the use of Kalman Filters to decouple tracking from control, as well as provide information for tracking algorithms. We describe two ocean vehicle mission scenarios and develop simple kinematic vehicle models which are suitable for use with the tracking algorithms to be considered. We then set up the tracking problem in a formal mathematical framework and explain simple heuristic approaches through the exploration versus exploitation tradeoff. We conclude with an explanation of the curse of dimensionality for multiple-vehicle sensor management, motivating the computationally tractable theory for non-myopic sensor scheduling to be introduced in the next chapter.

3.1 Motivation and Operations

The majority of applications involving multiple vehicle fleets in the ocean can benefit from drift-free acoustic tracking. As mentioned in Sec. 1.1.3, simple onboard sensors such as heading, attitude and depth cannot detect drift due to process noise, and vehicles with more capable inertial or Doppler sensors for dead-reckoning suffer from drift over time, a fundamental property of integrating a noisy signal. Global position updates can provide drift-free measurements that allow for vehicles to accurately localize. Due to a combination of convenience, economics and performance, the current trend for measuring position in a global reference frame is to use a USBL sonar unit mounted on the support ship to provide tracking of vehicles, and if needed, to send position updates down to the vehicle through an acoustic modem. Because of the constraints of the underwater acoustic channel, the entire system must be designed to make best use of the limited navigation resource provided by the USBL. The underlying goal of a multi-vehicle navigation system is to provide position tracking that will best help the fleet execute its mission. For context, we briefly outline two types of missions—the Vertical Glider Robot (VGR) example of subsea equipment delivery, and missions with teams of heterogeneous vehicles.

3.1.1 VGR Mission Example

An operational example describing the Vertical Glider mission for subsea equipment delivery helps illustrate the general system architecture of an outer tracking loop using USBL that corrects for drift, coupled with some limited amounts of autonomy and control onboard the individual vehicles. The Vertical Glider mission is the delivery of sensor packages to specific locations to form a grid on the seafloor. For the 49-vehicle grid sensing application discussed in Sec. 1.1.2, the mission will take roughly 50 hours total with vehicles falling 4000 meters at 1 m/s, which equals roughly 1 million dollars in ship operating costs (assuming costs $500,000 per day, which is standard in the offshore industry). Multiple vehicle simultaneous deployment has the potential to drastically reduce ship time while still meeting mission goals satisfactorily.

The vehicles will be dropped from a single stationary platform (such as a ship) and use their horizontal transit capability to reach the targets anywhere on the grid. The individual vehicles have limited onboard autonomy and attempt to drive to their targets. The USBL on the ship is used to correct for drift that the vehicles themselves cannot detect. Again, the USBL is a sensor with significant constraints because it nominally measures 1 vehicle every second—so when it measures one vehicle, it must ignore all the others also dropping to the seafloor. A lower bound on USBL sensor noise is the manufacturer spec of 0.1 degrees, which results in a standard deviation of over 7 m at 4000 m depth—so measurements every 50 seconds with a naive round-robin scheme will not give enough averaging (reduction in uncertainty) to achieve desired landing accuracy. Therefore, the fundamental problem we will now consider is how to best allocate these limited measurements of the USBL with the limited information we have available.

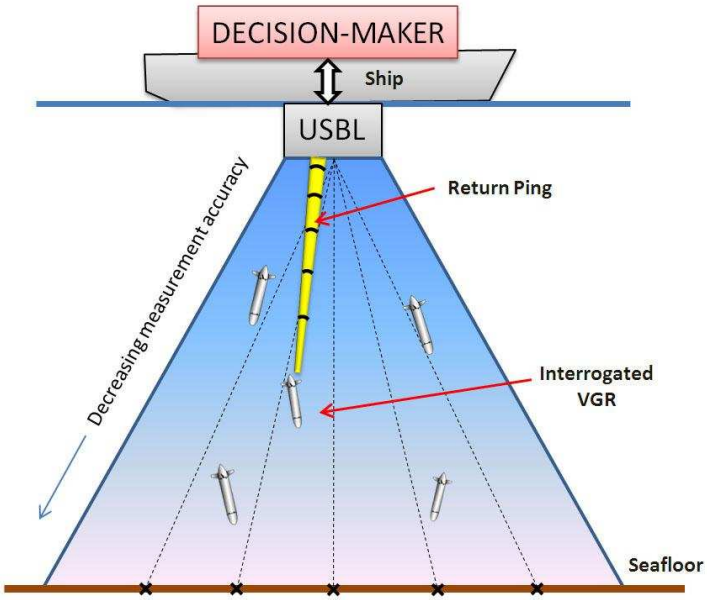


Figure 3-1: VGR navigation with USBL on ship. The USBL broadcasts an interrogation message indicating which vehicle to be measured. The indicated vehicle returns a ping, which is received by the USBL and used to calculate position. The navigation update is then broadcast back down to the vehicle via an acoustic modem.

The vehicles have identical dynamics and are subject to independent process noise with consistent statistics, however the sensor noise due to the USBL angular error

characteristic varies with depth. Operationally, individual Vertical Glider vehicles will be deployed sequentially in time from the ship. As the individual vehicles drop to the seafloor, they will each be at different depths at any given time. This variation in vehicle depth results in different noise parameters for each vehicle, which can be leveraged by the measurement allocation policy. Additionally, priority weighting for vehicles closer to the bottom can help achieve final landing accuracy; see Sec. 5.2.3 for a detailed discussion of this. The length of time in between individual vehicle deployments will vary depending on operational constraints as well as ship time economics, however it is a reasonable assumption that the spacing will be much shorter than the time it takes for an individual vehicle to reach the bottom, and long enough that there will be significant differences in vehicle depths at a given time. The gaps in time between vehicles are short (on order of a few minutes) so overall ship time is reduced compared to one at a time deployment. Performance of various algorithms as a function of spacing (and therefore ship time required to complete a mission) is given in Sec. 5.2.5. In summary, the sensor management problem for the Vertical Glider mission considers homogeneous vehicles with noise and priority weighting parameters that vary with depth, and desires the best way to allocate USBL hits for optimal navigation and thus control system performance to achieve landing accuracy.[1]

3.1.2 Heterogeneous Vehicles Mission Example

The second mission example is more general. For various reasons, future multiple-vehicle operations in the ocean will likely include heterogeneous fleets of vehicles. Some mission scenarios could require a mix of vehicles with different capabilities, all working together to achieve a certain objective. Due to the reliance on support ships for oceanographic research, other scenarios may include shared cruises with simultaneous deployment of multiple missions, possibly with each sub-mission con-

[1] A similar sensor management problem can be formulated for the recovery of the vehicles—guidance of the vehicles from locations on the grid back to the location of the ship for easy recovery. This problem is not considered in this thesis, but would be a straightforward modification.

sisting of fleets of homogeneous or heterogeneous vehicles, all which desire accurate geo-referenced navigation from the ship based sensor.

As mentioned in Sec. 1.1.3, onboard navigation sensors on underwater vehicles range from very simple compass and attitude, to more precise dead-reckoning based on Doppler Velocimetry (DVL). Inertial measurement units can be used to aid navigation, but dead-reckoning based on an IMU alone gives poor performance. As with the VGR application, simple vehicles with very basic onboard navigation are economical for use in large fleets, and with the help of USBL navigation may find significant use as part of heterogeneous vehicle teams. For vehicles with more capable onboard navigation, the fact still remains that no onboard navigation sensor can provide absolute geo-referenced position[2]—so measurement updates from the USBL on the ship can greatly improve navigational accuracy over long missions. Additionally, DVL based odometry is only useful when in range of a solid boundary, such as the seafloor. For many vehicles which operate at the bottom of the ocean, little navigation is available on the descent to the seafloor, although ADCP water profile dead-reckoning methods have the potential to improve this [95]. Once the vehicle reaches the bottom, it requires averaging of many USBL hits to obtain a good position estimate on which to initialize DVL-based dead-reckoning. Other missions may require vehicles to operate in the mid water column, away from DVL range to the seafloor.

It is easy to envision many combinations of these types of vehicles operating at once, and sharing the ship-based drift-free navigation sensor. In these cases, the vehicles have potentially different dynamics, onboard sensors, noise parameters, and priorities. Methods which balance these differences in order to optimally allocate measurement updates across the fleet can greatly improve navigational performance, enabling new complex missions and increasing the efficiency of ship-based vehicle operations at sea.

[2]Terrain-relative or visually-augmented navigation can observe relative drift, but require very reliable historical data and a stationary environment in order to provide absolute geo-referenced position

3.2 Overview of Navigation and Control

Navigation is often divided into two components: realtime navigation and postprocessed navigation. Postprocessed navigation includes acausal filtering and smoothing, and is often used to match data to position accurately. Realtime navigation relies on causal filtering for position estimation. The majority of vehicles underwater feed realtime navigation into guidance and control systems for localization purposes. The simplest way to control vehicles using a USBL navigation system would be to simply feed the USBL measurements into a controller for position. However, with the relatively slow update rate of the USBL as well as relatively large sensor noise, precise localization is impractical with this method. The overall control system performance can be greatly improved by adding some elements of onboard autonomy and control to the individual vehicles, with the USBL navigation as a supplement. This leads to USBL-aided dead-reckoning, much like land-based GPS-aided inertial navigation. We consider the vehicle control system at two levels, or at two time scales. The lower level attempts to address the following task: given a desired trajectory, what actions should the thrusters and/or control surfaces take in order to drive the vehicle in that trajectory? This level is concerned with fast vehicle dynamics, which are highly dependent on the particular vehicle design and hydrodynamics. Some examples of low-level controllers are pitch/roll control, hovering control, bottom-following control, etc. Vehicle control is an extensive subject, see [51] for a survey. The higher level of control, often known as guidance, is for positioning in a global reference frame. The USBL is a measurement input into a state estimator for vehicle position, and a simple position controller generates error commands that are input to the low-level controller.

The basic operation of the USBL is as follows. First the USBL sends out an 'interrogation ping' which specifies which vehicle should reply in order to be measured. Next, the interrogated vehicle sends a return ping back to the USBL transceiver, which is able to measure the range and bearing to the vehicle based on reception of the return ping. The USBL sends down a small data packet with the location

of the previously measured vehicle via an integrated acoustic modem, while sending out the next interrogation ping. Onboard the ship, there is a 'decision-maker' which tells the USBL which vehicle to interrogate. The measurement algorithms discussed in the remaining portions of the thesis primarily consider what happens inside this ship-based 'decision-maker.'

Update rates of the USBL are on the order of 1 Hz, and when the USBL is shared among multiple vehicles, update rates will be slower. Thus, vehicle dynamics which are handled by low-level control are much faster than the USBL update rate. Since we aim to study the general problem of sensor management for a centralized global sensor such as the USBL, we will idealize this controller, and assume that the vehicle in question is able to control itself well enough that we can approximate its dynamics as a kinematic particle—we will see in subsequent sections that the appropriate use of an idealized kinematic model enables applications of powerful theory for sensor scheduling.

3.2.1 Individual Vehicle Onboard Autonomy

The use of a Kalman Filter or similar estimator for vehicle position onboard the vehicle decouples the onboard control system from the USBL navigation updates.[3]. As we will see, this is an important property when dealing with multiple-vehicle deployments, as the USBL navigation updates may not be allocated to vehicles in an easily predictable manner. By using an estimator, the onboard control system drives to the desired position based on the position estimate, given by the estimator which is *always* running, incorporating USBL hits when they are available. In this way, we avoid interacting 'outer' and 'inner' feedback loops, which can cause problems when drastically different and nondeterministic update rates are used. The vehicle benefits greatly from USBL position updates, which correct for drift that cannot be detected

[3]Alternatives to the Kalman Filter such as the Extended Kalman Filter, Unscented (or Sigma Point) Kalman Filter, or deterministic observers (Luenberger, etc.) can more accurately handle nonlinear vehicle dynamics and non-Gaussian noise when estimating vehicle states [64]. However, for the purposes of this thesis we stick to the classical linear Kalman Filter, as we will use purely kinematic vehicle models in our development of sensor allocation algorithms

by the onboard control system, however the vehicle does not completely rely on the USBL updates—it does as well as it can based on whatever updates it receives.

The Kalman Filter running onboard each vehicle makes the vehicle agnostic to measurement updates—the Kalman Filter incorporates the information it receives and provides the best estimate of vehicle position to the control system at any moment based on the available measurements. Simple modifications of the standard Kalman Filter optimally handle missed measurements. One approach, taken in [92], is to set the measurement noise to infinity when no measurement is available, resulting in zero Kalman gain for that measurement and thus no contribution of the innovation. Alternatively, the measurement equation(s) can be changed each time step [94]. The intuition is that for a system with a single input measurement, the best estimate when no measurements are available is simply the open-loop propagation of the system model. For systems with random walk or double integrator dynamics (typical of vehicles which cannot compensate for drift), the position tracking error covariance thus increases linearly when no measurements are received, as expected. These methods easily handle multiple measurement scenarios, as measurements from various sensors can be incorporated at different update rates

The vehicle's onboard controller takes the estimate from the Kalman Filter as the input, making the control problem independent of the tracking problem to first approximation. Thus, as explained earlier, we leave the design of the onboard controller as a separate problem specific to individual types of vehicles, and consider vehicle dynamics as seen by the outer tracking loop as an abstraction which represents the dynamics of the vehicle including it's onboard control. Fig. 3-2 shows the control loop onboard the individual vehicle, illustrating the use of the Kalman Filter onboard to incorporate intermittent measurements from the USBL. In this example, the vehicle and its onboard controller does the best it can to steer based on the KF estimate, but cannot correct for drift. Thus, the vehicle dynamics as seen by the USBL loop are a simple scalar kinematic drift model: $\frac{1}{s}$ in the frequency-domain, $A = 0$ in continuous time, or $A = 1$ in discrete-time. We note that while tracking and control in the ocean includes complicated geometry, we consider the one-dimensional case here in order to

gain intuition with a simple framework that captures the fundamental aspects of the sensor management problem. Extensions to high-fidelity dynamic models as well as three dimensional geometry depend on mission and vehicle scenarios and are possible, but require vector process models and are left for future work.

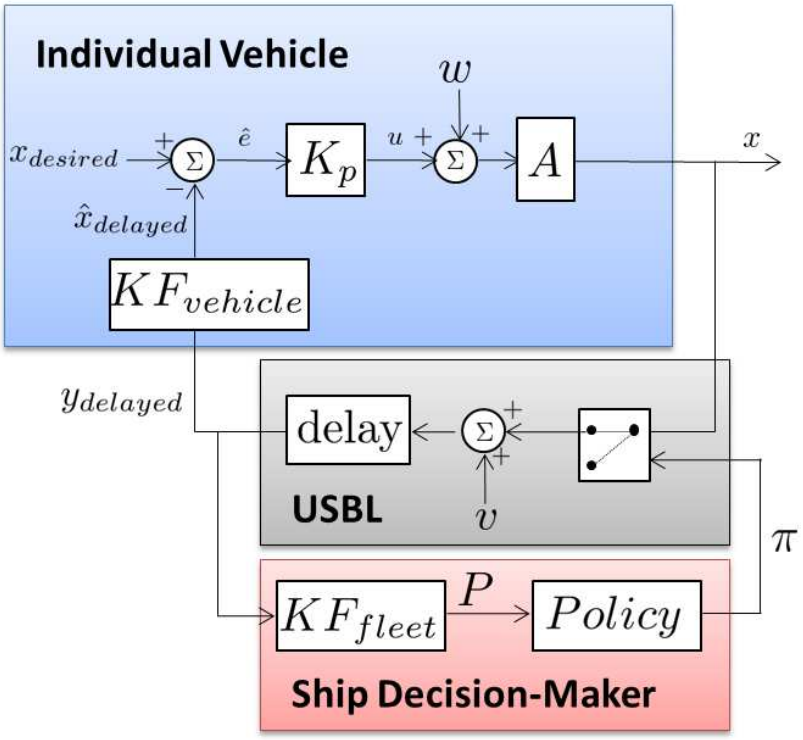


Figure 3-2: Control loops onboard each individual vehicle. The ship-based decision maker governs the measurement update from the USBL, which is input into a Kalman Filter running on the vehicle. The vehicle's onboard state estimate is then fed into a proportional controller for position. In this example, the vehicle and its onboard controller does the best it can to steer based on the KF estimate, but cannot correct for drift. Thus, the vehicle dynamics as seen by the USBL loop are a simple scalar kinematic drift model, $A = 0$ in continuous time

3.3 Tracking Problem Formulation

The measurement updates from the USBL help individual vehicles correct for drift, however, the 'decision-maker' onboard the ship must decide how best to allocate the USBL interrogations. Fig. 3-2 illustrates the use of the *policy* π to control operation of the USBL. For each vehicle at each decision step, π is an indicator variable, set

to 1 if that vehicle is to be measured, and set to 0 if not (for the full fleet, π is a vector of indicator variables, one for each vehicle). To decide which vehicle to measure, a Kalman Filter tracking the entire fleet runs onboard the ship. This fleet KF essentially runs a KF for each vehicle in parallel (a bank of low-dimensional filters), and the tracking error covariance is used as the 'information state' that is input into measurement allocation algorithms.[4]

Here, we provide the formal problem statement for the multiple-vehicle Kalman Filter tracking problem, adapted from Le Ny et al. in [71], which builds on the general problem outlined by Whittle in [108]. Due to the lengths of underwater missions relative to the time scales of underwater vehicle dynamics, we use an infinite-horizon formulation. The algorithms we will use require scalar linear time-invariant (LTI) systems with Gaussian noise, so we formulate the problem to satisfy these assumptions. For ocean systems, Gaussian process noise is a reasonable assumption because disturbances are largely due to bluff body hydrodynamics and small-scale turbulence. Slowly-varying and non-Gaussian or correlated process noise due to large-scale ocean currents can be mostly corrected for through the use of *a priori* current profiles or predictions. We note that the problem formulation and approach given in [71] has extensions to multidimensional systems, however for simplicity we stick to the scalar formulation here as it is what we will analyze in Chapter 4. Additionally, while the approaches of [71] and [108] allow for multiple sensors m (assuming the number of vehicles is significantly larger than the number of sensors), we restrict our formulation to the $m = 1$ case for notational clarity (and because operational limitations usually result in use of a single USBL transceiver).

The sensor management task is to provide state estimates for all targets that minimizes the weighted mean-square error on the system states plus additional measurement costs. The targets to be tracked are N independent Gaussian linear time-

[4]We note that the term 'information state' is used here to refer to the state which is relevant for the decision-making problem, as is common in operations research and decision theory literature. This is not to be confused with Fisher information in filtering literature, which is the inverse of the covariance. The Fisher information matrix is used in the maximum information formulation of estimation problems, which is the dual of the standard minimum covariance formulation used in this thesis.

invariant (LTI) systems whose dynamics evolve according to

$$\dot{x}_i = A_i x_i + B_i u_i + w_i, \quad x_i(0) = x_{i,0}, \quad i = 1, \ldots, N \tag{3.1}$$

where A_i describes the dynamics of vehicle i, $B_i u_i$ is the control input, and the driving process noise w_i is a stationary white Gaussian noise process with zero mean and a known continuous-time power spectral density W_i, i.e. $\mathrm{Cov}(w_i(t)w_i(t)') = W_i\delta(t-t')$, $\forall t, t'$. If the sensor observes target i, a noisy measurement is obtained according to

$$y_i = C_i x_i + \nu_i \tag{3.2}$$

where C_i is the system measurement model for target i and ν_i is a stationary white Gaussian noise process with power spectral density V_i, assumed to be positive-definite. We note that while Le Ny et al. consider the continuous time case, the implementation of sensor scheduling in a real system is inherently a discrete-time process and a finite sample period must be chosen. The continuous-time description of the problem allows for powerful analysis methods, and real-world system dynamics of course evolve in continuous time, so this method allows true continuous-time dynamics to be used in the solution. For the specific analytic solution for LTI scalar systems, any discretization of the system will in fact give the exact states of the continuous-time equivalent system at the sample times.

The goal is a measurement policy, which is denoted by π. Define

$$\pi_i = \begin{cases} 1 & \text{if vehicle } i \text{ is observed at time } t \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

The sensor operates under two constraints. The sensor can observe at most one system at each instant

$$\sum_{i=1}^{N} \pi_i(t) \leq 1, \quad \forall t, \tag{3.4}$$

and each system can be observed by at most one sensor at each instant:

$$\pi_i(t) \leq 1, \quad \forall t, \tag{3.5}$$

The problem considered is an infinite-horizon average cost problem to design an observation policy $\pi(t) = \{\pi_i(t)\}$ satisfying the constraints 3.4 and 3.5, and a state estimator $\hat{x}_\pi$ of the state of all targets x that *it depends only on the past and current observations produced by the observation policy (causal)*, such that the average weighted error covariance over all targets, plus measurement costs are minimized. The cost function γ is thus

$$\gamma = \min_{\pi, \hat{x}_\pi} \overline{\lim_{\tau_f \to \infty}} \frac{1}{\tau_f} E\left[\int_0^{\tau_f} \sum_{i=1}^{N} \left((x_i - \hat{x}_{\pi,i})' T_i (x_i - \hat{x}_{\pi,i}) + \kappa_i \pi_i(t)\right) dt\right] \tag{3.6}$$

where $\kappa_i \in \mathbb{R}$ is the measurement cost per unit time when target i is observed, the T_i's are positive semidefinite weightings (how important a low error covariance is for a given target compared to another), and $\overline{\lim}$ denotes the upper limit, or lim sup.[5]

The Kalman-Bucy filter gives an unbiased state estimate, $\hat{x}_{\pi,i}$ in continuous time, with $\hat{x}_{\pi,i}$ for all vehicles $i = 1, \ldots N$ updated in parallel following

$$\frac{d}{dt}\hat{x}_{\pi,i}(t) = A_i \hat{x}_{\pi,i}(t) + B_i(t)u_i(t) - P_{\pi,i}(t)\left(\pi_i(t)\frac{C_i}{V_i}\left(C_i \hat{x}_{\pi,i}(t) - y_i(t)\right)\right) \tag{3.7}$$

We note that since $B(t)$ does not factor in the evolution of the tracking error uncertainty, it is allowed to be time-varying. For scalar systems, the error covariance matrix $P_{\pi,i}(t)$ for system i satisfies the algebraic Ricatti differential equation

$$\dot{P}_{\pi,i}(t) = 2A_i P_{\pi,i}(t) + W_i - \pi_i \frac{C_i^2}{V} P_{\pi,i}(t)^2 \tag{3.8}$$

The dependence on the policy is evident in that the terms having to do with a new

[5]The formal statement uses lim sup because the covariance is inherently periodic (or at least has intermittent jumps downward) due to the switching observations—so lim sup means the upper limit of those cycles (since there is no true steady-state). Since the limit is as $T_f \to \infty$, as T_f gets longer, T_f could fall at different points in the measurement cycle, so the time average will move up and down, requiring the use of the supremum.

observation are switched on and off by the policy indicator function $\pi_i(t)$. Thus, we refer to this as the *conditional* Ricatti equation.[6] Note that while the covariance evolution is dependent on the policy, due to the use of the Kalman Filter, it does not depend on the actual observation values—only if a measurement is taken. This means that the Kalman Filter handles the stochastic aspects of the system, and the problem of finding the optimal policy becomes a *deterministic* optimal control problem, described by the cost function

$$\gamma = \min_{\pi} \overline{\lim_{\tau_f \to \infty}} \frac{1}{\tau_f} \left[\int_0^{\tau_f} \sum_{i=1}^{N} \left(T_i P_{\pi,i}(t) + \kappa_i(t) \pi_i(t) \right) dt \right] \tag{3.9}$$

subject to the constraints 3.4 and 3.5, where $E((x_i - \hat{x}_i)' T_i (x_i - \hat{x}_i)) = (T_i P_{\pi,i})$ and the dynamics of the error covariance are given by 3.8.

3.3.1 Simple Vehicle Model Development for Tracking Algorithms

Here, we consider simple analysis of basic models of onboard control. The motivation is to develop simple but useful models that can be used in scalable, computationally tractable multi-vehicle sensor scheduling algorithms, which will be explained further in Chapter 4. The use of LTI systems greatly simplifies the mathematical analysis, and is a reasonable assumption for the idealized kinematic vehicle models we desire. However, we will see certain situations where significant approximations must be made to meet the LTI assumption; we note the limitations of our approach and mention some possible approaches. The rigorous extension of sensor scheduling algorithms (and the associated vehicle models) to time-varying and non-Gaussian formulations are subjects for future work.

We consider two cases discussed earlier and commonly encountered in the ocean:

[6] We note that the use of the binary indicator π to denote the policy is redundant with the convention that measurement noise covariance is set to ∞ when there is no measurement. This convention allows for time-invariant measurement models. The important aspect of this conditional algebraic Ricatti equation is that it cannot be solved by conventional means, because it is time-varying in a unique way due to the switching of the policy.

vehicles with no dead-reckoning capabilities, and vehicles with a DVL (and compass). Two possible measurements are available to the vehicle: y_{USBL} from the USBL with noise covariance V_{USBL}, and y_{DVL} from the DVL with noise covariance V_{DVL}

$$y_{USBL} = \begin{cases} x + \nu_{USBL} & \text{if } \pi = 1 \\ \text{NaN} & \text{if } \pi = 0 \end{cases}$$

$$y_{DVL} = \dot{x} + \nu_{DVL}$$

The objective is to state the scalar Kalman Filter parameters A, C, W, V for the outer tracking USBL loop for each onboard sensor scenario. Since we assume to first approximation that good tracking will lead to good control, for sensor allocation algorithms we only care about the tracking error uncertainty P as predicted by the Kalman Filter (not the actual state estimate $\hat{x}$, which will be used by the onboard controller). Thus, for the purposes of the tracking algorithm model, we do not consider the control $B(t)u$. For the purposes of state estimation and control, the model can have control or not—but the value of $B(t)$ does not matter to the sensor tracking algorithms because it doesn't affect the tracking error uncertainty propagation.

USBL only, No DVL or IMU

For the case where the vehicle has no DVL or IMU, it relies completely on the USBL for position updates (compass/attitude/depth sensors are used for onboard control). The vehicle dynamics are driven entirely by process noise (and control), and the vehicle behaves following an open-loop drift model, $A = 0$ in continuous time. The USBL observation is a noisy measurement of the vehicle position, so $C = 1$. The vehicle drifts according to environmental and hydrodynamic process noise, which we will denote W_{env}. The noise on the measurement is V_{USBL}.

USBL and noisy DVL

In this scenario the vehicle can navigate without help of the USBL by dead-reckoning based on integration of noisy velocity measurements.[7] To properly fuse onboard DVL measurements with intermittent USBL measurements, a state estimator must use second-order dynamics. A nominal Kalman Filter formulation could use x and $\dot{x}$ as the state variables, with vehicle dynamics modeled as a double integrator. This approach allows the noise from the DVL to be properly added onto the velocity measurement. Combined DVL and acoustic navigation has been studied extensively for use with underwater vehicles; for the full 3D treatment, see the approaches taken in [23, 85]. Here, we attempt to capture the fundamental aspects of navigation using an onboard DVL augmented by intermittent USBL hits in a very simple first order model suitable for use in sensor tracking algorithms.

An ideal (noiseless) DVL would be able to correct for process noise drift, resulting in zero process noise. From the view of the outer USBL loop, the vehicle with a noisy DVL is affected by process noise which is related to V_{DVL} and is smaller than W_{env}. Thus, the abstracted kinematic model does not include process noise due to the environment (closed-loop control using the DVL can correct for this).

$$\dot{x} = u_{DVL} \tag{3.10}$$

The onboard controller acts on the DVL measurement and can be arbitrarily represented in the frequency domain as $C(S)$: $u_{DVL} = C(S)y_{DVL}$. For development of the abstract outer loop vehicle model we must assume a form of this controller; the simplest approach is PI control: $C(s) = -\frac{K}{s}$. The control input becomes $u_{DVL} = -Kx - \frac{K}{s}\nu_{DVL}$. In the time domain, vehicle dynamics are given by

$$\dot{x} = -Kx - \int K\nu_{DVL}dt \tag{3.11}$$

[7] In reality, heading is required for dead-reckoning. A compass provides noisy heading measurements, which contribute to the drift error in complex ways depending on the trajectories taken as well as vehicle dynamics. Recent advances in true north-seeking gyrocompasses have helped with this problem [64]. For purposes of simple model development, we treat the heading control as part of the idealized onboard controller.

The integral of a Gaussian random variable is a random walk model—the expected excursion grows with time. This is behavior is captured by a second order KF formulation; transformation of 3.11 to the Laplace domain verifies that a second order system model is required to capture the dynamics properly.

$$\frac{X}{\nu_{DVL}} = \frac{-K}{s(s+K)} \tag{3.12}$$

For a scalar kinematic model, we need to approximate $\int(K\nu_{DVL})dt$ as a Gaussian random variable. This cannot be done exactly; the approximation will be parametrized by some time period from when the DVL dead-reckoning was last initialized (e.g. last USBL hit). There are a few approaches that can be taken, with varying levels of accuracy and difficulty. The simplest solution would be to choose some estimate of a characteristic time period for USBL updates. This could be done in conjunction with an analysis of the measurement allocation algorithm; however the scheduling policy from the algorithm will depend on the process noise, so this analysis would likely need to be performed iteratively. Two more complex approaches could be more accurate, but require more advanced mathematical approaches for sensor allocation algorithms which are beyond the scope of this thesis.[8]

3.3.2 Simple Heuristic Approaches

The infinite-horizon tracking cost integral (3.9) suggests that the sensor allocation algorithms must deal with a tradeoff: focus on the present, or try to plan for the future? This is an example of what is known as the *exploration versus exploitation tradeoff*, which is a common theme in information acquisition problems that arise in both sensor tracking and machine learning applications.

Heuristic approaches to the multiple vehicle tracking problem are best explained

[8]The first approach would be to use a higher order Kalman Filter model to more accurately model the DVL noise and fusion with the USBL—this would require the multidimensional extension to the scheduling approach given in [71], which has certain limitations and is computationally more intensive than the approach we take. The second approach would be to stay with the scalar system model, but modify the basic optimization problem given in [108] and repeated in Eqn. 4.6 such that non-autonomous (time-varying) dynamics may be included.

through the exploration versus exploitation tradeoff, illustrated in Fig. 3-3 The problem is to balance acquisition of information from which to make decisions (exploration), with decisions that aim to best use currently known information for the most gain in reward (exploitation). We will consider two commonly-used heuristics: a round-robin scheme that performs maximum exploration, and a greedy algorithm which performs maximum exploitation.

Round-robin schemes are commonly used in-practice for measurement and communication between multiple agents. Since measurements are obtained for all vehicles at equal frequencies, the round-robin method explores the state space as much as possible. Round-robin methods are well-suited to scenarios when little or no information is known *a priori*, such as initialization, or when considerable dynamic uncertainty exists. However, unless the systems to be measured are identical and operate in identical conditions, with identical priorities for measurements, a round-robin scheme is not optimal for sensor allocation.

Greedy heuristics are a popular method for handling large, difficult problems, due to very tractable computation. The greedy algorithm makes the locally-optimum choice at each decision stage—in the case of multiple vehicle tracking, the algorithm allocates a measurement to the vehicle with the highest instantaneous weighted tracking uncertainty: $\max_i(T_i P_i(t))$. This enables use of the vehicle, sensor and noise models employed by the Kalman Filter, and which gives the potential for improvement over the naive round-robin scheme. However, greedy algorithms are short-sighted and may produce suboptimal or even worst-case solutions. Decision-making based on only the instantaneous state ignores the *non-myopic* prediction power that is possible when models are known. For vehicle tracking, the use of the Kalman Filter necessitates use of a model already—it makes sense to utilize this information in the sensor allocation algorithm.

3.3.3 The Curse of Dimensionality

While heuristic methods are simple, computationally tractable, and commonly used today, it is evident that better approaches are possible. As discussed in Sec. 1.2.2, op-

Figure 3-3: The exploration versus exploitation tradeoff. Round-robin performs maximum exploration, while greedy performs maximum exploitation. The index approach (developed in Chapter 4) balances the two.

timal scheduling policies can theoretically be found through brute-force enumeration, or through dynamic programming. These methods avoid the degenerate performance which can occur when using myopic heuristics such as round-robin and greedy algorithms. However, brute-force enumeration becomes computationally intractable in all but the smallest problem cases. Powell describes three curses of dimensionality commonly encountered in sequential decision-making problems [83]:

1. The state space: if the state variable has I dimensions, and can take on L possible values, there could be up to L^I different states.

2. The outcome space: if output of the system has J dimensions, with M possible outcomes, there could be up to M^J different outcomes.

3. The action space: if the decision vector π has K dimensions, and can take N outcomes, there might be N^K different possible actions.

Of course, for continuous problems, any of these dimensions could be *infinite*, requiring discretization or analytical methods (which can either complicate or simplify the problem, depending on the situation).

Dynamic programming can effectively solve sequential decision-making problems for certain special structures; one successful and relevant result is optimal control theory, which can effectively solve problems with continuous state, outcome and action spaces. However, for multiple-vehicle tracking, the curse of dimensionality still holds due to the combination of continuous time dynamics along with combinatorial decision-making choices: N vehicles with state estimate dynamics coupled through the measurement constraint, with N possible choices to measure at each time step.

3.4 Summary

In this chapter, we have described a general architecture for multiple-vehicle deployments relying on, or augmented by, a centralized global navigation system. We have chosen to abstract the low-level vehicle dynamics and control into simple kinematic models, which describe vehicle dynamics adequately for the purpose of tracking algorithms for allocation of geo-referenced position updates from the ship-based sensor. We described two ocean vehicle mission scenarios and developed simple vehicle models which are suitable for use with the tracking algorithms to be considered. These models will be used in the theoretical development in Chapter 4 as well as the computational experiments of Chapter 5. The use of Kalman Filters allows for decoupling between vehicle tracking and vehicle control, and provides a natural framework for implementing tracking algorithms. We have formulated the Kalman Filter multiple-vehicle tracking problem and explained simple heuristic approaches. The curse of dimensionality was introduced as a major challenge for non-myopic sensor allocation methods, which motivates the discussion of bandit-based sensor management algorithms to come in the Chapter 4.

Chapter 4

Bandit Approaches to Sensor Management

We discuss the theoretical basics of a problem structure well-suited to constrained sensor management, known as the Multi-Armed Bandit (MAB) problem. The general formulation of the MAB problem is outlined, a simple one-dimensional 'single-armed' bandit example is given to give intuition, and a canonical example of the MAB is discussed briefly. We outline the solution method for the Gittins Index policy for the MAB problem, as well as introduce the extension to the MAB known as Restless Multi-Armed Bandits, which fits the dynamic nature of the vehicle tracking problem. We give a specific Restless Bandit example, which is suitable for use with the Kalman Filter tracking problem outlined in Chapter 3. Finally, we present the vehicle tracking solution given in [71] using Restless Bandit Kalman Filters (RBKF) for optimal sensor scheduling, which is the basis of the computational experiments given in Chapter 5.

4.1 Multi-Armed Bandits

The Multi-Armed Bandit problem is named after a slot machine analogy, where each slot machine is termed a 'single-armed bandit,' and the problem is to choose a slot machine from a number of choices to play at a given time in order to maximize long-term winnings. The problem falls into the general framework of stochastic scheduling [78]

and considers situations where the goal is obtain a large cumulative reward as a result of sequential decisions between a number of choices. Making a choice results in a stochastic reward as the output, which is modeled as a probability distribution. Each time a choice is made (this is referred to as *playing the bandit*, which results in one of the bandits becoming *active*), a reward is observed, and these observations form the basis for the *knowledge state*, which is the decision-maker's estimate of the reward distribution of each bandit. The decision-maker learns about the effect of the choices, and uses this model as the basis for making future decisions. However, the problem is how to best balance improving the model (*exploring* decisions in order to observe the outcomes and improve the distribution estimate), versus gaining rewards (making choices that the current model estimate predicts will give good outcomes — *exploiting* current knowledge). This problem is fundamental to many situations that arise in real life, such as the gambling example, finance (choosing stocks to research), experiment design (in clinical trials, which treatment to give to which patient to maximize fairness of treatment to all participants in the trial), and information acquisition in machine learning problems. For the vehicle tracking problem (discussed in more detail in subsequent sections), 'playing the bandit' can be interpreted as taking a measurement of a specific vehicle. The 'reward' in the bandit framework is the reduction in covariance (uncertainty) due to the measurement of the vehicle.

The MAB problem falls under the general framework of Partially Observable Markov Decision Processes (POMDPs). In general, POMDPs are intractable to solve optimally in all but the smallest dimension problem instances. However, the specific structure of the MAB problem allows for a tractable solution method, which is a *priority index policy*. By solving for a priority index that represents the intrinsic value of playing each bandit, a hierarchical ranking can be made which makes the decision very easy – play the bandit with the highest index. The attractive feature of this priority index policy is its computational tractability. The index is computed independently for each bandit, reducing a large dimension problem into a number of easily computed low dimension problems, thus addressing the curse of dimensionality for the MAB class of problems.

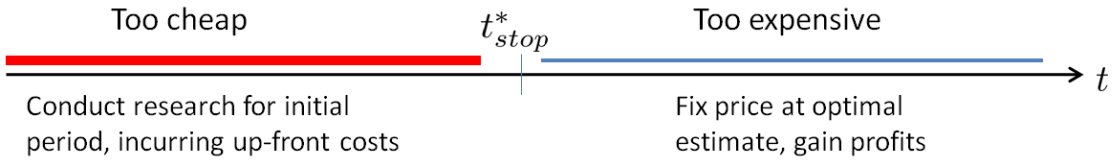


Figure 4-1: The Single Armed Bandit (SAB), or optimal stopping time problem. Terminating market research too early results in suboptimal long-term pricing (company was too cheap), while continuing market research for too long is a waste of money—a case of diminishing returns in terms of improvement offered by market research. The optimal stopping time maximizes the infinite-horizon discounted reward.

Single-Armed Bandit Example To gain intuition about the tradeoffs involved in the MAB problem, we'll consider an example of a single-armed bandit problem: the optimal stopping time for market research when determining a price for a product. As a highly idealized example, imagine a company is trying to decide the best price for its product to maximize revenue. It can conduct market research, but at some cost. While conducting this research the company is still selling the product at its best estimate of the optimal price. The goal is to maximize revenue over time. In this simplified scenario, an optimal policy exists: perform all the market research in some initial exploration period, then set the price, as shown in Fig. 4-1. The logic for this optimal policy is as follows: if market research is performed after the initial period, the product will be selling at a suboptimal cost in the in-between period, while still incurring the same total cost of market research. The fundamental question is how long to make the initial exploration period before switching to exploitation: the optimal stopping time t^*_{stop}, as shown in Fig. 4-1. If t_{stop} is too short, then the company is 'too cheap' — money spent on more market research would result in a better price that would result in more profits over time. If t_{stop} is too long, then money is being wasted on market research that is excessive — the extra research will do little to improve the optimal price. This illustrates the diminishing returns property inherent to the bandit structure.

Multi-Armed Bandit Example Now we will make a further generalization of the above stopping problem, wherein the decision is which one of a number of measure-

ments to make, and when. The simplest way to understand this type of a problem, known as a Multi-Armed Bandit problem, is through a gambling analogy, illustrated in Fig. 4-2. Consider a set of K slot machines at a casino (each known as a 'single armed bandit'), each with a fixed but unknown distribution of winnings. The problem for a gambler is to decide which slot machine to play at a given time in order to maximize the long-term cumulative winnings.[1] For convergence arguments in the development of the theory, accumulated winnings are discounted over time by a constant factor $\beta \in [0, 1]$, often > 0.90. The information at the gambler's disposal is the results of playing the bandit—by choosing to play a given slot machine and observing the winnings, the distribution of the slot machine's winnings can be inferred over numerous plays. For normally distributed rewards, the knowledge state at time n, S^n, of the gambler includes, for each bandit i, estimates of the mean reward and standard deviation of the rewards, and the number of times it has been played: $S^n = (\overline{\theta}_i^n, \overline{\sigma}_i^{2,n}, N_i^n)$. We see that there is a difficult decision to be made here: the gambler can choose to play slot machines that appear to have a favorable distribution based on the information known (*exploitation*), or he can choose to play a different slot machine from which little information has been gathered in order to learn whether this slot machine may be a better candidate (*exploration*).

4.1.1 Multi-Armed Bandit Theory: Gittins Index

The MAB problem appears to be prohibitively large: N agents, each with a number of discrete or continuous states, and N possible choices of which bandit to play. However, it has been shown by Gittins and Jones in their 1974 paper [54] that this problem can be greatly reduced in dimension by using an index policy. One index can be computed for each bandit, using information only about that bandit. The optimal solution is to compute this index, now known as the Gittins index, for each

[1] We note that this problem formulation is a limited real-world analogy for illustrative purposes—the distributions must be favored towards the player. This is not the case in casino gambling, where all of the slot machine distributions are favored towards the house. In that case, the optimal choice for long-term expected rewards is to not play at all! However, the 'adversarial multi-armed bandit problem' does have applications, and is considered in [14]

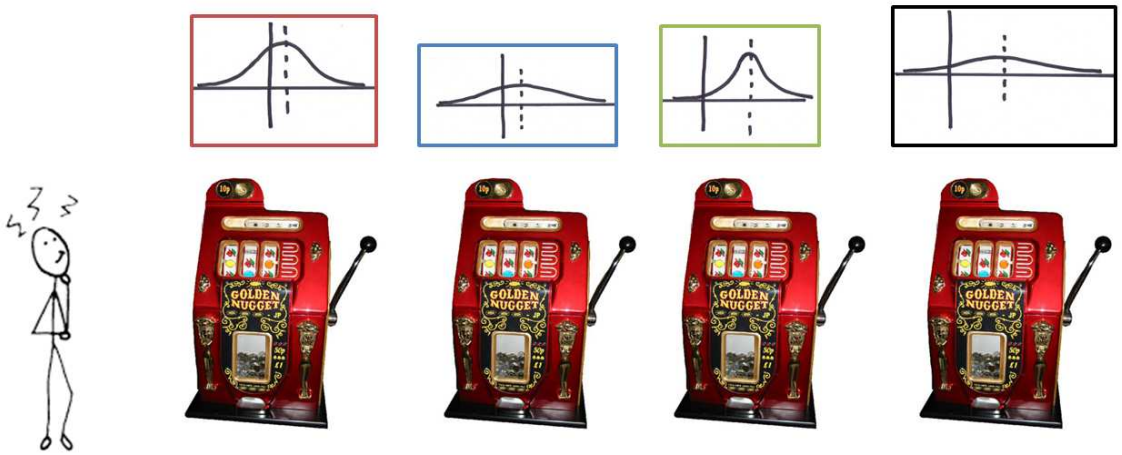


Figure 4-2: Cartoon illustrating the Multi-Armed Bandit problem with slot machines ('single-armed bandits'). A decision-maker must sequentially choose one machine to play out of multiple options. Each slot machine has a different distribution of winnings, which is unknown by the decision maker. The decision-maker can estimate the distributions by observing results from playing a given machine, and can use those estimations to inform future choices.

bandit at each time step, and then play the bandit with the highest index. Thus, the N-dimensional problem can be turned into a series of N one-dimensional problems, greatly improving computational tractability. In order to be eligible for a Gittins index solution, a stochastic decision-making or control problem must exhibit the following properties [100]:

1. Only one project is played (*active*) at each time step (decision epoch)

2. Idle (*inactive*) projects are frozen - the knowledge state remains the same unless the bandit is played

3. Idle/frozen projects contribute no reward

Computation of the Gittins index for each bandit considers comparison of retirement with a fixed reward with the expected future rewards, based on current knowledge of the state. The Gittins index is the value of this fixed reward that makes the controller indifferent to choosing to stop with fixed reward or to continue by playing

the bandit. The Gittins index for a single armed bandit can also be thought of as an optimal stopping time problem with two arms - the bandit and another arm which has fixed rewards. The optimal time to switch from the bandit to the arm with the fixed rewards is the optimal stopping time, and the fixed reward that makes the current time the optimal stopping time is another interpretation of the Gittins index. The computation of the Gittins index is difficult and involves solving an optimality recursion, with the value at step n described implicitly as

$$V_n = \max\left[\frac{\rho}{1-\beta}\ ,\ \mathrm{E}\left\{\bar{\theta}_n(\bar{x}_n)|\bar{x}_n\right\} + \beta\mathrm{E}\left\{V_{n+1}|\bar{x}_n\right\}\right] \tag{4.1}$$

where β is a discount, $\bar{x}_n$ is the estimate of the current state, $\theta_n(\bar{x}_n)$ is the immediate reward at step n and ρ is a hypothetical fixed reward. The Gittins Index, v, is the value of ρ that makes the two terms in the max argument equal, satisfying

$$\frac{v}{1-\beta} = \mathrm{E}\left\{\bar{\theta}_n(\bar{x}_n)|\bar{x}_n\right\} + \beta\mathrm{E}\left\{V_{n+1}|\bar{x}_n\right\} \tag{4.2}$$

Solution of (4.1) or equivalently (4.2) for the Gittins Index is possible using value iteration [62] or other methods [93, 100].

4.1.2 Standard Normal Gittins Index

For the case of normally distributed rewards, the computation is simplified. In a similar manner as the standard normal random variable allows computation of quantities related to any Gaussian distribution, we can compute a 'standard normal Gittins index' [83]. This index only depends on the number of observations/measurements/plays that the bandit in question has received at time step n, and whether the variance of the bandit is known or unknown. Thus, the Gittins index v can easily be computed as:

$$v(\overline{\theta}_i^n, \overline{\sigma}_i^{n,2}, N_i^n) = \overline{\theta}_i^n + \overline{\sigma}_i^n\Gamma(N_i^n) \tag{4.3}$$

where $\overline{\theta}_i^n$ and $\overline{\sigma}_i^{n,2}$are the estimates of the mean and variance of project i and time n, N_i^n is the number of times project i has been sampled at time n, and $\Gamma(n) =$

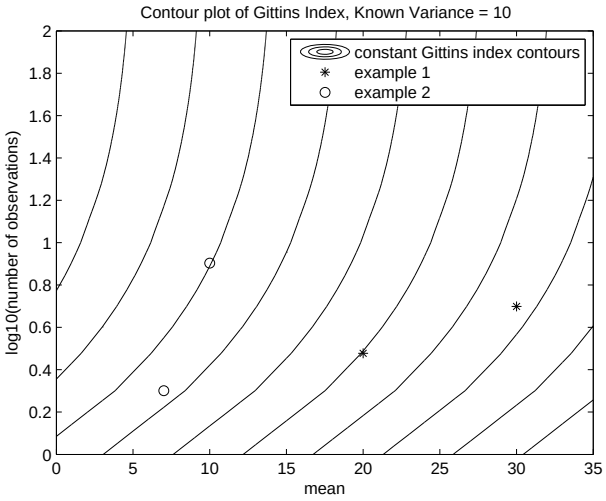


Figure 4-3: Contour plot of the known variance Gittins Indices as a function of mean and number of observations. Larger values of the Gittins index are towards the bottom right of the plot. The two simple examples given in the text for two-bandit projects are shown on the plot. The higher priority for sites with low number of measurements at a given mean illustrates the exploration versus exploitation tradeoff.

$v(0, 1, n)$—the standard normal Gittins index for n observations with zero mean and unit variance. Page 338 of [83] includes a table of the standard normal Gittins indices as a function of the number of observations for discount factors of 0.95 and 0.99, and the known and unknown variance cases. A contour plot of the Gittins Indices for a known variance of 10 is shown in Fig. 4-3. We can see that sites with a high mean and a low number of observations have the highest value; that is, they are the sites have the highest priority for being sampled. Due to the exploration versus exploitation tradeoff balanced by the Gittins Index, it can be more advantageous in the long-term to sample sites that have a slightly lower mean, but have been visited a low number of times. For example, if we have a two-bandit project at time $n = 8$ with mean estimates $\bar{\theta}_1^8 = 20$ and $\bar{\theta}_2^8 = 30$, a constant variance of $\sigma^2 = 10$, and have taken 3 measurements of project 1 ($N_1^8 = 3$) and 5 measurements of project 2 ($N_2^8 = 5$), our indices will be $v_1^8 = 20 + \sqrt{10}\Gamma(3) = 20 + 0.8061\sqrt{10} = 22.55$, and $v_2^8 = 30 + \sqrt{10}\Gamma(5) = 30 + 0.5747\sqrt{10} = 31.82$. Thus, project 2 has a higher Gittins Index and we would choose to play that project next. After taking the next measurement of project 2, we would update our estimate of project 2's mean (project

1's mean will stay the same), plug that into the Gittins Index formula with $N_2^9 = 6$ and $N_1^9 = 3$, and again see which project has a higher index. A different two-bandit project with $n = 11$, mean estimates $\bar{\theta}_1^{11} = 10$ and $\bar{\theta}_2^{11} = 7$, a constant variance of 10, and 8 measurements of project 1 and 3 measurements of project 2 yields Gittins indices of $v_1^{11} = 14.14$ and $v_2^{11} = 15.06$. We would choose to measure project 2, which may be counterintuitive since it has a lower mean. However, as the contour plot shows, the long-term expected reward is better if the exploratory choice is made at this stage.

4.2 Restless Bandits

One of the more restrictive assumptions of the MAB problem is the frozen state assumption. In reality, many systems have dynamics that evolve regardless of whether a decision is made or not. For systems with quantifiable dynamics for both the active and inactive phases, there is an extension of the MAB problem known as the Restless Bandit (RB) problem, proposed by Whittle in 1988 [108].

The formulation considers projects $i = 1 \dots n$, with state variables x_i, and two distinct Markov transition operators for active, and passive phases: P_{i1} and P_{i2}. The immediate rewards realized in the active and passive phases are g_{i1} and g_{i2}. The projects are observed by $m < n$ sensors.

Define the long-term reward from project i as r_i. The problem is

$$\begin{aligned} \underset{\pi}{\text{maximize}} \quad & \mathrm{E}(\sum_i r_i) \\ \text{subject to} \quad & \sum_i \pi_i = n - m \end{aligned}$$

where π_i is an indicator variable for the policy: $\pi_i = 1$ if i is active, $\pi_i = 0$ if i is passive. This formulation uses the constraint $m(t) = m$. Whittle's solution method is to first relax the activity constraint to an average activity constraint:

$$\mathrm{E}[m(t)] = m \tag{4.4}$$

so that the constraint can be adjoined to the Lagrangian. This relaxation technique is commonly used throughout constrained optimization. Using the average activity constraint, the problem becomes

$$\underset{\pi}{\text{maximize}}\ \mathrm{E}(\sum_i r_i + v\sum_i \pi_i) \tag{4.5}$$

which is an unconstrained problem (since the adjoined relaxed constraint is included in the cost function). As with standard dynamic programming problems [20], the value of being in a given state must be fixed to some reference, so a function f_i is defined which represents the differential reward caused by transient effects of starting in state x_i rather than an equilibrium state. Define γ_i as the average reward over time for project i operated without constraint. This value is obtained via

$$\gamma_i + f_i = \max[g_{i1}(x_i) + (P_{i1}f_i)(x_i), v + g_{i2}(x_i) + (P_{i2}f_i)(x_i)] \tag{4.6}$$

where $f_i = f_i(x_i, v)$. The dual function yields the maximum average reward $R(m)$ under the relaxed constraint:

$$R(m) = \inf_v[\sum_i \gamma_i(v) - v(n-m)] \tag{4.7}$$

which is concave. As with the Gittins index solution, the index v is obtained by setting $v_i(x_i)$ so that the controller is indifferent to being active or not:

$$g_{i1}(x_i) + (P_{i1}f_i)(x_i) = v_i + g_{i2}(x_i) + (P_{i2}f_i)(x_i) \tag{4.8}$$

The interpretation of v_i is similar to that of the Gittins index: v_i is a subsidy for passivity (or measurement tax, depending on the convention chosen). The Whittle formulation suggests an alternative interpretation that connects with constrained optimization: v_i corresponds to the Lagrange multiplier associated with the constraint on average activity.

Here, Whittle introduces the important concept of *indexability* for Restless Bandit

problems. Simply put, the index v must induce *consistent* orderings. In other words, a project that is rested with index v will also be rested with index $v' > v$. Indexability is related to submodularity, briefly mentioned in Sec. 1.2.2, which is the notion of diminishing returns, similar to convexity for set functions. Indexability requires monotonic increases in the set of passive actions as the measurement tax (index) increases.

Formally, call $D_i(v)$ the set of values of x_i for which project i is rested. A project is indexable if $D_i(v)$ increases in size monotonically from $\emptyset$ to $\mathbf{X}_i$ as v increases from $-\infty$ to $+\infty$, and $\mathbf{X}_i$ is the full state space of x_i. An important result is that projects are *always* indexable if there are no dynamics in the passive mode, i.e. $P_{2i} = I$ (the standard MAB/Gittins case). Projects are *not* always indexable otherwise—this is why indexability is not encountered in a study of Gittins literature.

Whittle suggests a suboptimal but natural index scheduling policy: choose *exactly* m projects with the highest v_i to activate. This enforces the rigid constraint. The relationship between average rewards is as follows

$$R_{ind}(m) \leq R_{opt}(m) \leq R(m) \tag{4.9}$$

where $R_{ind}(m)$ is the average reward under the index *policy* used, $R_{opt}(m)$ is the optimal average reward bound for the exact $m(t) = m$ constraint, and $R_{ind}(m)$ is the optimal average reward for the relaxed problem (under which the indices are derived) with the constraint $\mathrm{E}[m(t)] = m$. When inactive projects are static, the Whittle index reduces to the Gittins index as expected, and the resulting policy is optimal.

For vehicle tracking, such as the VGR system, this framework is much more accurate than the MAB, as the vehicle continues to move with its open-loop dynamics whether a measurement is taken or not. The projects or systems to be scheduled are the vehicles, and activation of a project corresponds to taking a measurement of that vehicle.

4.2.1 One Dimensional Deterministic Whittle Index

Whittle gives a concrete example of the derivation of indices for the one dimensional deterministic system case. As described in Sec. 3.3, despite the fact that the vehicle tracking problem is stochastic, the use of the Kalman Filter turns the determination of the scheduling policy into a deterministic optimization problem. Here, we outline Whittle's formulation and solution of the general one dimensional deterministic problem. Consider first order continuous time systems, described by

$$\dot{x} = a_k(x) \tag{4.10}$$

where x is a vector of system states, and a_k is a set of two vectors describing the dynamics of the systems, with $k = 1, 2$ describing the active and passive phases, respectively. For each system, we aim to solve

$$\gamma + f = \max[g_1(x) + (P_1 f)(x), v + g_2(x) + (P_2 f)(x)] \tag{4.11}$$

which is (4.6) from Chapter 3, repeated here with the system subscripts i removed for clarity. For this example, the Markov transition operator P is a time derivative. Thus, $(P_k f)(x) = (\frac{d}{dt}(f))$. Since $\frac{d}{dt}(x) = a_k$, we obtain $(P_k f)(x) = \frac{\partial f}{\partial x} a_k$, and (4.11) becomes

$$\gamma = \max\left[g_1 + \frac{\partial f}{\partial x} a_1, v + g_2 + \frac{\partial f}{\partial x} a_2\right] \tag{4.12}$$

From this equation we can deduce expressions for $\frac{\partial f}{\partial x}$ by setting γ equal to the RHS when $k = 1$ or $k = 2$.

$$\frac{\partial f}{\partial x} = \begin{cases} \frac{\gamma - g_1}{a_1} & \text{if active, } k = 1 \\ \frac{\gamma - g_2 - v}{a_2} & \text{if passive, } k = 2 \end{cases} \tag{4.13}$$

Whittle notes that this quantity, $\frac{\partial f}{\partial x}$, and its derivative with respect to x, $\frac{\partial^2 f}{\partial x^2}$, must be continuous on some arbitrary decision boundary (threshold value of x). This gives a system of two equations, from which we can eliminate γ and obtain a relation for

$v(x)$ as a function of $a_k(x)$ and $g_k(x)$:

$$\frac{\gamma - g_1}{a_1} = \frac{\gamma - g_2 - v}{a_2} \Rightarrow \gamma = \frac{g_1 a_2 - a_1 g_2 - a_1 v}{a_2 - a_1} \tag{4.14}$$

$$\frac{a_1 \frac{\partial}{\partial x}(\gamma - g_1) - \frac{\partial a_1}{\partial x}(\gamma - g_1)}{a_1^2} = \frac{a_2 \frac{\partial}{\partial x}(\gamma - g_2 - v) - \frac{\partial a_2}{\partial x}(\gamma - g_2 - v)}{a_2^2} \tag{4.15}$$

Some algebra gives the solution for the Whittle index for one-dimensional deterministic projects, assuming the indexability requirement is met.

$$v(x) = g_1 - g_2 + \frac{(a_2 - a_1)(a_2 g_1' - a_1 g_2')}{a_2 a_1' - a_1 a_2'} \tag{4.16}$$

The quantities on the right hand side of the equation are evaluated at x, and primes denote differentiation with respect to x.

4.2.2 Restless Bandits with Kalman Filters

The MAB example in Sec. 4.1.1 using the standard normal Gittins index requires knowledge of the mean and variance of the observed rewards. This can be done using simple equations for the recursive updates of the mean and variance. For the tracking of stochastic dynamical systems, this invites a clear connection to the Kalman Filter, which is an optimal state estimator for linear time-invariant systems under Gaussian noise assumptions, and is well-suited for real-time recursive implementation. For vehicle tracking, the information state is the tracking error covariance, P. Following the conditional Ricatti equation, (3.8), the error covariance of the vehicles being tracked evolves with two distinct dynamics: one when active (measurement taken), and one when passive.[2] This fits the description of Restless Bandit projects in Sec. 4.2. The Whittle index v defines an intrinsic value for measurement of a given system, which takes into account immediate and future gains. This computation is performed

[2]We note that the conditional Ricatti equation with $\pi_i = 0$ is technically no longer a Ricatti equation—it becomes a Lyapunov equation

independently for each vehicle, and then the controller simply selects the vehicle with the highest index (or in the case of multiple sensors, the vehicles with the M highest indices) for the next measurement(s). We can plug in the corresponding dynamics and rewards from the scalar system Kalman Filter into Whittle's one dimensional project index result. The active and passive dynamics (a_1 and a_2, respectively), as well as the active and passive rewards (g_1 and g_2, respectively) are given by

$$\begin{aligned} a_1 &= 2AP + W - \frac{C^2}{V}P^2 \\ a_2 &= 2AP + W \\ g_1 &= -TP - \kappa \\ g_2 &= -TP \end{aligned}$$

where A describes continuous system dynamics, C is the measurement model, W is the process noise covariance, V is the sensor noise covariance, T is a priority weight on the error covariance, and κ is the measurement cost. Plugging these values into (4.16) we obtain the Whittle index v as a function of the covariance P:

$$v(P) = -\kappa + \left(\frac{C^2}{V}\right)\frac{TP^3}{2\left(AP + W\right)} \tag{4.17}$$

Looking at the Whittle formula, we can see that the denominator can equal zero for certain values of P when $A < 0$ (a stable system). Intuitively, this brings up an important point. When $A \geq 0$, the covariance grows without bound when no measurements are received. However, when $A < 0$, the covariance reaches a steady-state value even in the absence of measurements. This suggests that special consideration must be given to the derivation of indices based on the conditional Ricatti equation.

4.3 Scheduling Kalman Filters

Following on the theory of Whittle, multiple vehicle tracking using Kalman Filters is formally studied by Le Ny, Feron and Dahleh in [71], by posing sensor scheduling

for multiple targets as an optimal control problem. For scalar systems (such as the vehicle outer loop tracking kinematic model given in the previous chapter), they give an analytic solution for an index policy which is a specific form of the Whittle Index. To differentiate the Scheduling Kalman Filter index from the generic Whittle index, we will refer to the index as derived by Le Ny et al. as λ. Le Ny et al. use the same basic approach as Whittle, however give more thorough treatment for all cases of system dynamics and covariance regions. In Appendix A we give a detailed outline of their solution method, as well as show some extended explanations of certain key concepts. Here, we describe the main adjustments made to Whittle's solution and present the closed-form analytic solution.

Le Ny et al. first observe that the covariance evolves in fundamentally different ways depending on whether the system is stable and the value of the covariance relative to steady-state values of the Ricatti equation, which has two roots, x_1 and x_2

$$x_{1,2} = \frac{A \pm \sqrt{A^2 + C^2 W/V}}{C^2/V} \tag{4.18}$$

We assume that $W \neq 0$ (this can be enforced mathematically if necessary by adding a small amount to W; physically this is justified by the fact that process noise is inherent in real-world systems), so x_1 is strictly negative and x_2 is strictly positive. Thus we can take x_2 as the steady-state covariance when the vehicle is always measured. Additionally, if we consider the passive (no measurement) case, we set $\pi = 0$ and (3.8) becomes the Lyapunov equation $2AP + W = 0$. For stable systems ($A < 0$) this equation has a strictly positive solution, $x_e = -\frac{W}{2A}$. This represents the steady-state covariance when no measurements are taken. Note that marginally stable or unstable systems ($A \geq 0$) have no steady state covariance. The active and passive steady state covariance values for a stable system are thus

$$\pi = 1\text{: } P_{ss}^{active} = x_2$$

$$\pi = 0\text{: } P_{ss}^{passive} = x_e$$

Define three different covariance regions which will be used in the solution

$$\text{Region 1: } 0 < P < x_2$$

$$\text{Region 2: } x_2 < P < x_e$$

$$\text{Region 3: } P > x_e$$

For a marginally stable system (The scalar kinematic vehicle drift model $A = 0$, corresponding to a random walk, potentially with control), note that there is no steady-state covariance in the passive mode—we consider $x_e \to \infty$ as $A \to 0_-$, so the covariance remains in region 1 or 2.

The solution method for the nontrivial cases ($T \neq 0$ and $C \neq 0$) first assumes an optimal form for the policy, which takes advantage of the special structure of Restless Bandit problems. Following the discussion of indexability, and the concept behind the single-armed bandit example given in Sec. 4.1, the form of the optimal policy is a *threshold* policy. For some threshold covariance value P_{th}, the policy observes the system when $P \geq P_{th}$ and does not observe for $P < P_{th}$. The approach is to determine the value of the average cost $\gamma(\lambda)$ and the threshold $P_{th}(\lambda)$. In a sense, we solve for the index λ in the opposite way from the way we use it in the policy—we assume a fixed threshold covariance and find the value of λ that satisfies the optimality equation. Since the system is indexable if and only if $P_{th}(\lambda)$ is an increasing function of λ, we can invert this relation to give the index $\lambda(P)$; note that this index is now a function of the actual covariance P of the vehicle at that instant, which is given by the Kalman Filter. Based on the covariance regions described above (in relation to the steady-state values, which are functions of the system model), we must consider three cases for the location of this hypothetical threshold covariance $P_{th}(\lambda)$. We can solve for the index λ in each region separately, and combine these solutions to define λ as a piecewise linear function of P.

For the edge cases (regions 1 and 3), the solution method is natural. In these cases, the threshold is either in an active region (region 1), or passive region (region 2), since the threshold covariance is below the active steady-state (region 1), or above

the passive steady-state (region 2). Thus, after a potential transient period, in these regions the covariance will converge in finite time to the neighborhood of the steady-state covariance of the given region, allowing for direct solution of the index λ. These situations are not considered by the basic Whittle solution, and thus allow proper formulation of the index for stable systems, as well as transient scenarios.

In region 2, the hypothetical threshold covariance P_{th} is in between the steady-state covariance values x_2 and x_e. Thus, there is no explicit relation to provide the value of the average cost. Here, Le Ny et al. use the same formulation as given by Whittle, with the justification that plugging in the index formula indeed satisfies the governing optimality equation.

4.3.1 Scalar Systems: Closed-Form Solution

Here we present the closed-form analytic solution from [71], given in (4.20), and shown graphically for two example systems (one stable and one marginally stable) in Fig. 4-5.

- Case $C_i = 0$ or $T_i = 0$:

$$\lambda_i(P_i) = -\kappa_i, \qquad \forall P_i \in \mathbb{R}_+ \tag{4.19}$$

- Case $C_i \neq 0$ and $T_i \neq 0$:

$$\lambda_i(P_i) = \begin{cases} -\kappa_i + \frac{T_i P_i^2}{P_i - x_{1,i}} & \text{if } P_i \leq x_{2,i} \\ -\kappa_i + \frac{C_i^2 T_i P_i^3}{2V_i(A_i P_i + W_i)} & \text{if } x_{2,i} < Pi < x_{e,i} \\ -\kappa_i + \frac{T_i C_i^2 P_i^2}{2|A_i|V_i} & \text{if } x_{e,i} < P_i \end{cases} \tag{4.20}$$

  where x_1, x_2 and x_e are given by

$$x_{1,2} = \frac{A \pm \sqrt{A^2 + C^2 W/V}}{C^2/V}$$

$$x_e = \begin{cases} -\frac{W}{2A} & \text{if } A < 0 \\ \infty & \text{if } A \geq 0 \end{cases}$$

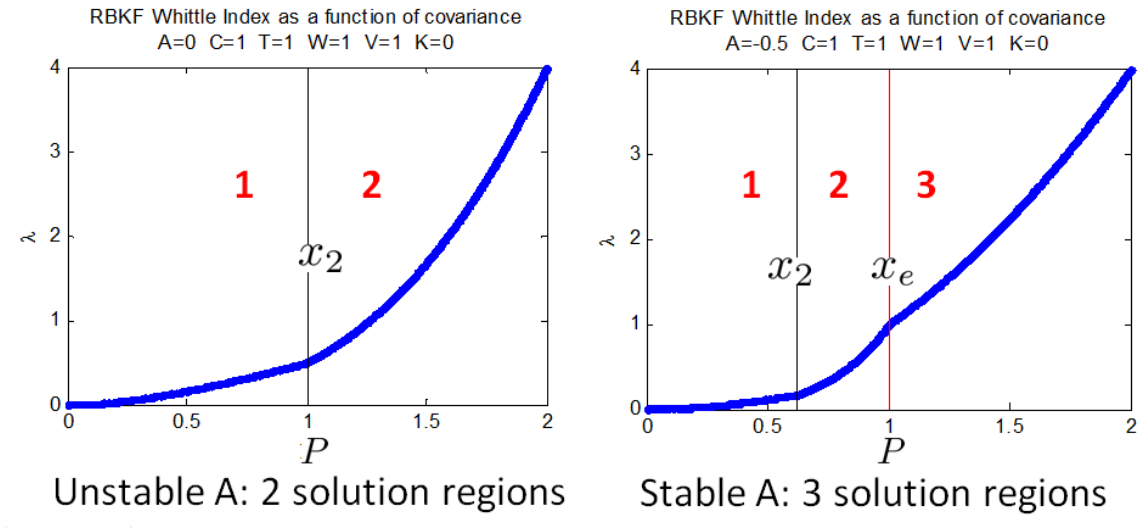


Figure 4-4: Plot of the Whittle index $\lambda(P)$ for two example systems. The index is a piecewise-linear increasing polynomial in P, which verifies indexability. make better matlab versions of this?

4.3.2 Implementation of Index Policy

The closed-form index solution allows for efficient real-time implementation of the scheduling policy. As described in Chapter 3, a Kalman Filter is run onboard the decision-maker to estimate the states and tracking uncertainties of the entire fleet of vehicles. The tracking error covariance P_i for each vehicle can simply be plugged into the closed-form index equations along with the model parameters A_i, C_i, W_i, V_i, T_i and κ_i for that vehicle. The vehicle (or M vehicles) with the largest index λ is chosen for a measurement at the next time step. A flowchart illustrating the real-time process for multiple-vehicle tracking using the index policy is shown in Fig. 4-4. We will refer to this policy as the Restless Bandit Kalman Filter (RBKF) index algorithm.

We note that in practice, the covariance predominately remains in region 2—most of the time the index is given by Whittle's original solution. Region 1 is a transient region, and is thus rarely encountered in steady-state operation. Region 3 is rarely to be visited since a stable system is unlikely to have a covariance greater than the steady-state covariance when no measurements are taken—this would need to be the result of a large initial covariance, or changing of model parameters. However, this

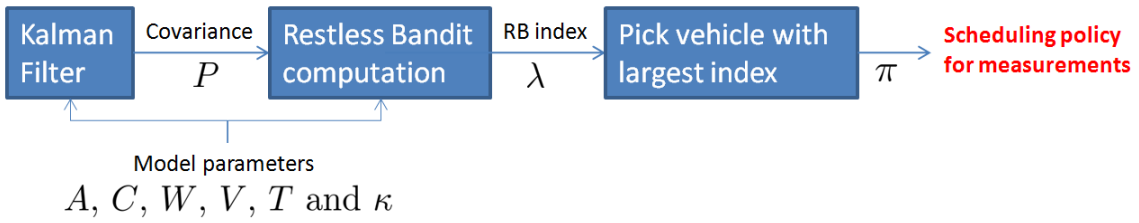


Figure 4-5: Flowchart illustrating implementation of the Scheduling Kalman Filters index algorithm for multiple vehicle tracking.

does illustrate a benefit of the closed-form analytical solution: if model parameters change (for example, due to changes in operation governed by the mission), the new parameters can simply be plugged into the index equations. While this is suboptimal in general, the resulting policy will be optimal going forward under the assumption of infinite-horizon LTI tracking using the new parameters. This approach to varying model parameters is implemented in Sec. 5.2.3.

4.4 Summary

We have given a theoretical tutorial on the Multi-Armed Bandit (MAB) problem, as well as the Restless Bandit problem, an extension to systems with both active and passive dynamics. The approach of Whittle [108] for deriving Restless Bandit priority index policies has been described and applied to multiple vehicle tracking using Kalman filters. We have discussed the more complete treatment of Whittle's formulation for Kalman filter sensor scheduling by Le Ny et al. in [71], and presented the closed-form analytical solution given for scalar LTI systems. This Restless Bandit Kalman Filter (RBKF) algorithm will be investigated in Chapter 5 and compared to commonly-used heuristics for representative multiple-vehicle tracking problems in the ocean. Notably, the RBKF algorithm is computationally tractable and adds only a small increase in computational expense compared to the heuristic methods.

Chapter 5

Computational Experiments

We now use the sensor scheduling algorithms discussed in Chapter 4 in computational experiments. While the MAB and Restless Bandit problems have received considerable theoretical attention in literature, very few experimental results exist, even in simulation. We investigate the performance of the Restless Bandit Kalman Filters (RBKF) scheduling algorithm from Le Ny et al. [71] in simulated mission scenarios of heterogeneous fleets of LTI vehicles as well as the subsea equipment delivery example with depth-varying parameters. For the LTI case, we consider a generic scenario of varying process and measurement noise parameters, as well as two scenarios that model fleets with mixtures of vehicles with and without dead-reckoning capabilities (DVL and compass). In these cases, the index algorithm consistently outperforms the heuristic methods, and does well even in cases where the greedy heuristic shows degenerate performance compared to the round-robin (RR) baseline. For the subsea equipment delivery case, we show how the RBKF index equations can be used in a suboptimal quasi-static manner to handle depth-varying parameters. In all of these examples, performance is affected by mission length, illustrating the influence of the horizon length on the exploration versus exploitation tradeoff.

5.1 Heterogeneous Vehicles, Linear Time-Invariant Parameters

Le Ny et al. give one small computational result comparing the RBKF and greedy heuristic algorithms for a two vehicle system; we have investigated the performance of the RBKF index algorithm with larger fleet sizes and different combinations of varying parameters throughout the fleet. The cases considered in this section all include heterogeneous fleets of vehicles with LTI parameters—this fits the exact assumptions and framework used in the derivation of the RBKF index policy, and we use the index solution exactly as given in [71].

A couple implementation details about the simulations are worth noting. We simulate in discrete-time, using a time step of one second. This matches the 1 Hz update rate of the USBL, and since we are using scalar kinematic models, vehicle dynamics will be accurately represented at the simulation time steps. This brings up an important practical issue when implementing the RBKF index policy. The index solution is formulated in continuous-time; however sensor observations and the policy π are inherently discrete. We use the discrete-time Kalman Filter to update the error covariance $P_i(t)$ using discretized system models, and every time step we evaluate the RBKF index using the continuous time model parameters. An examination of the evolution of the RBKF indices occasionally reveals some large spikes; these are artifacts of the discretization. However, since measurements and decisions physically occur at discrete intervals, this behavior is both expected and accurate (similar to the effects of a zero order hold when using discrete-time controllers). Additionally, we must scale the discrete time process noise covariance to match the continuous time spectral density W which is used by the RBKF index equations.

We now give results from three example scenarios comparing the performance of the RBKF index and greedy heuristic versus the RR baseline. Performance is evaluated based on the average cost (weighted covariance) per vehicle, averaged over

the entire mission:

$$\gamma = \frac{1}{T_f} \sum_{1}^{T_f} \left(\frac{1}{N} \sum_{i}^{N} T_i P_i(t) \right) \tag{5.1}$$

where T_f is the mission time, in integer seconds. This cost is a modification of the cost function (3.6) in the original problem formulation; it is modified for use with finite length missions and normalized by the number of vehicles—this allows for more intuitive comparisons between different fleet sizes. We note here that by convention these costs are expressed in units of variance, $[m^2]$, as opposed to RMS values. For the results given in this thesis, we set all measurement costs to zero, because we assume the ship has unlimited power available (and the small pingers onboard the vehicles have negligible effect on vehicle battery life) and the USBL will be working to maximum capacity at all times. In the heterogeneous vehicle scenarios, we weight tracking of each vehicle equally, $T_i = 1, \forall i$. For each scenario we give plots of the average cost of each algorithm as a function of fleet size, evaluated for fleets of size $N = [2, 10, 30, 50, 70, 100, 150, 200, 300]$. Additionally, we show the % improvement in cost of the RBKF index and greedy heuristic algorithms over the RR baseline.

Mission length is an important parameter in these simulations. For evaluating the performance of the RBKF index in scenarios for which it is intended, long missions are required (to attempt to match the infinite-horizon assumption). The mission length that qualifies as 'infinite-horizon' can be considered a mission length for which longer missions have negligible change on the average cost per vehicle—the transients have a sufficiently small effect on the result. The effect of transients on mission length is heavily dependent on fleet size, as the length of the transient period grows in proportion to fleet size. For simulation purposes, we have empirically determined that a mission length of 10,000 seconds (10,000 total measurements from the USBL) gives good insight into the infinite-horizon performance of the algorithms (the upcoming results will show that for the largest fleets the transients still have an effect, however basic intuition can be gained, and for the purposes of this study the computational time required to run longer simulations was not justified). For the mixed DVL fleet examples, we also give results for a much shorter mission time (1,000 seconds) in order

to show the effects of breaking the infinite-horizon assumption. In real operations, mission times vary, so it is important to understand the performance of scheduling algorithms used (suboptimally) in finite-horizon situations.

5.1.1 Case 1: Vehicles with Varying Sensor and Process Noise

In order to compare the algorithm performance when vehicles in the fleet have large differences in parameters, we first consider a hypothetical example where process noise and measurement noise increase across the fleet. For vehicles $i = 1 \dots N$, the process noise is set as $W = \text{logspace}(-2, 1, N)$ and the measurement noise is set as $V = \text{logspace}(-1, 2, N)$. For example, vehicle 1 in each fleet has $W = 0.01$ and $V = 0.1$, while vehicle N in each fleet has $W = 10$ and $V = 100$. The mission time is $T_f = 10,000$ sec.

Results are shown in Fig. 5-1. The upper plot shows the average cost integral (5.1) (tracking performance) plotted for the three algorithms as a function of fleet size. The bottom plot shows the % improvement over RR for greedy and index algorithms. From the top plot, we see that the average cost per vehicle in general increases as fleet size grows, due to sharing a single sensor among a larger number of vehicles. As expected, the average cost per vehicle when using the RR algorithm increases roughly linearly with fleet size. From the bottom plot, we see that the greedy algorithm is worse than RR for low fleet sizes, and slightly better than RR for large fleet sizes. The index algorithm consistently improves over the RR baseline by roughly 40%, largely independent of fleet size. While we note that large fleet sizes are investigated in order to understand the workings of the algorithm for (near) asymptotically-large deployments, measurable improvement is seen for small, physically-realizable fleet sizes as well. This example demonstrates that in scenarios with greatly varying noise parameters throughout the fleet, the RBKF index algorithm can give large performance benefits, and the greedy algorithm does not necessarily improve over RR.

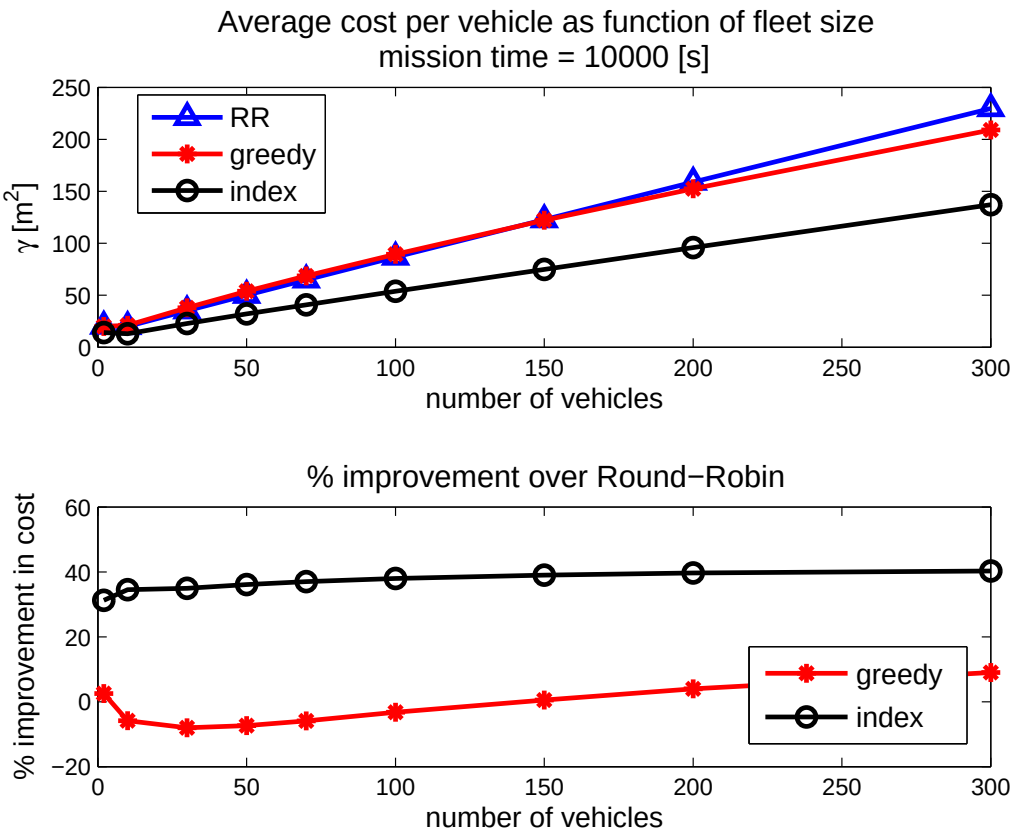


Figure 5-1: Results from heterogeneous vehicle LTI experiments. Case 1: Process noise W and measurement noise V increase logarithmically across the fleet, mission time $T_f = 10,000$ seconds. The upper plot shows the average cost (5.1) plotted for the three algorithms as a function of fleet size. The bottom plot shows the % improvement over RR for greedy and index algorithms. The index algorithm shows measureable improvement at all fleet sizes.

5.1.2 Case 2: Fleet of Vehicles With and Without Dead-Reckoning, Constant Measurement Noise

We consider missions where some vehicles have a DVL and compass and are in range of bottom-lock, while other vehicles do not perform any dead-reckoning. As a simple example, we will consider half the fleet with DVL and half the fleet without. Following on the discussion of simple models in Sec. 3.3.1, we represent the vehicles with DVL through much lower process noise (as discussed in Sec. 3.3.1, the time-dependent random walk nature of dead-reckoning drift is not accurately modeled here). We

model the vehicles which are dead-reckoning with process noise varying from $W = 0.001$ to $W = 0.1$ in order to roughly approximate vehicles which have been dead-reckoning for various amounts of time, or are operating at different depths or mission scenarios which may affect the dead-reckoning drift rate. The vehicles not performing dead-reckoning have a process noise of $W = 2$. In this first scenario, we assume use of a good XY position sensor, with measurement noise $V = 1$. This could be a high quality USBL tracking vehicles in close range (for example, 0.3 degree error at 200 meter range).

We give performance results of average cost per vehicle as well as % improvement over RR as a function of fleet size for three mission lengths. A short mission of $T_f = 1,000$ sec is shown in Fig. 5-2, a moderate length mission of $T_f = 3600$ sec is shown in Fig. 5-3, and a long mission of $T_f = 10,000$ sec is shown in Fig. 5-4. The performance of the algorithms for different fleet sizes and mission lengths illustrates the exploration versus exploitation tradeoff and the differences between greedy and index methods. In general, index and greedy improve over RR, and index improves the most (better than greedy). However, performance depends on the ratio between fleet size and mission length. In Fig. 5-2, we see that the RBKF index achieves improvements of roughly 30% to 40% over RR for fleets larger than 2 vehicles. Here, the index algorithm again shows measureable performance improvements for fleets of 10 vehicles, which is a practically-realizable deployment today, or at least in the near future. The greedy heuristic improves over RR for small fleets, but does not perform as well as the RBKF index. However, for large fleets, the performance of the greedy heuristic matches that of the RBKF index algorithm. For short missions, the exploration portion of the exploration versus exploitation tradeoff is not very important—when a relatively small number of decisions are to be made, exploitation often gives the best outcome. In terms of the ratio of number of decisions to be made versus number of choices for those decisions, larger fleet sizes represent the shortest relative horizon for a given mission time. We see that for the shortest horizons, the RBKF index essentially performs the greedy action, choosing to perform exploitation. These methods show great improvement over RR, which is performing maximum exploration. In Fig. 5-

3, the effect of increasing performance for the greedy algorithm at large fleet sizes is still noticeable, but is not as pronounced. In Fig. 5-4, infinite-horizon behavior exists for nearly all fleet sizes, and the result is nearly constant performance relative to RR for the index and greedy algorithms as fleet sizes grow. The RBKF index shows large improvements over both RR and the greedy algorithm. Notably, while the greedy algorithm's pure exploitation strategy results in performance that varies greatly depending on fleet size and mission time, the RBKF index algorithm shows relatively constant performance benefits over the RR baseline, demonstrating the ability to effectively find the optimal balance between exploration and exploitation.

For some intuition about why the performance varies, a closer look at the $N = 10$ and $T_f = 10,000$ seconds case is shown in Fig. 5-5. The left column shows the measurement distribution—the percentage of total measurements given to each of the 10 vehicles by the scheduling policy. The right column shows the corresponding contribution to the total cost of each vehicle, as a result of the scheduling policy. The rows correspond to the RR, greedy heuristic and RBKF index algorithms. The measurement distributions show the large difference between the RR baseline and the two Kalman filter-based approaches: Vehicles $i = 6 \ldots 10$ are given equal numbers of measurements because they have the same parameters ($W = 2$), while vehicles $i = 1 \ldots 5$ are given slightly different numbers of measurements due to different process noise parameters. While the measurement distributions from the greedy heuristic and RBKF index policies do not look drastically different, the subtle differences in policy result in large differences in the cost contributions of the vehicles. The greedy heuristic essentially attempts to equalize the cost contribution of all vehicles, shown by the relatively flat distribution. The RBKF index cost distribution is in between that of greedy and RR, which results in a lower total cost.

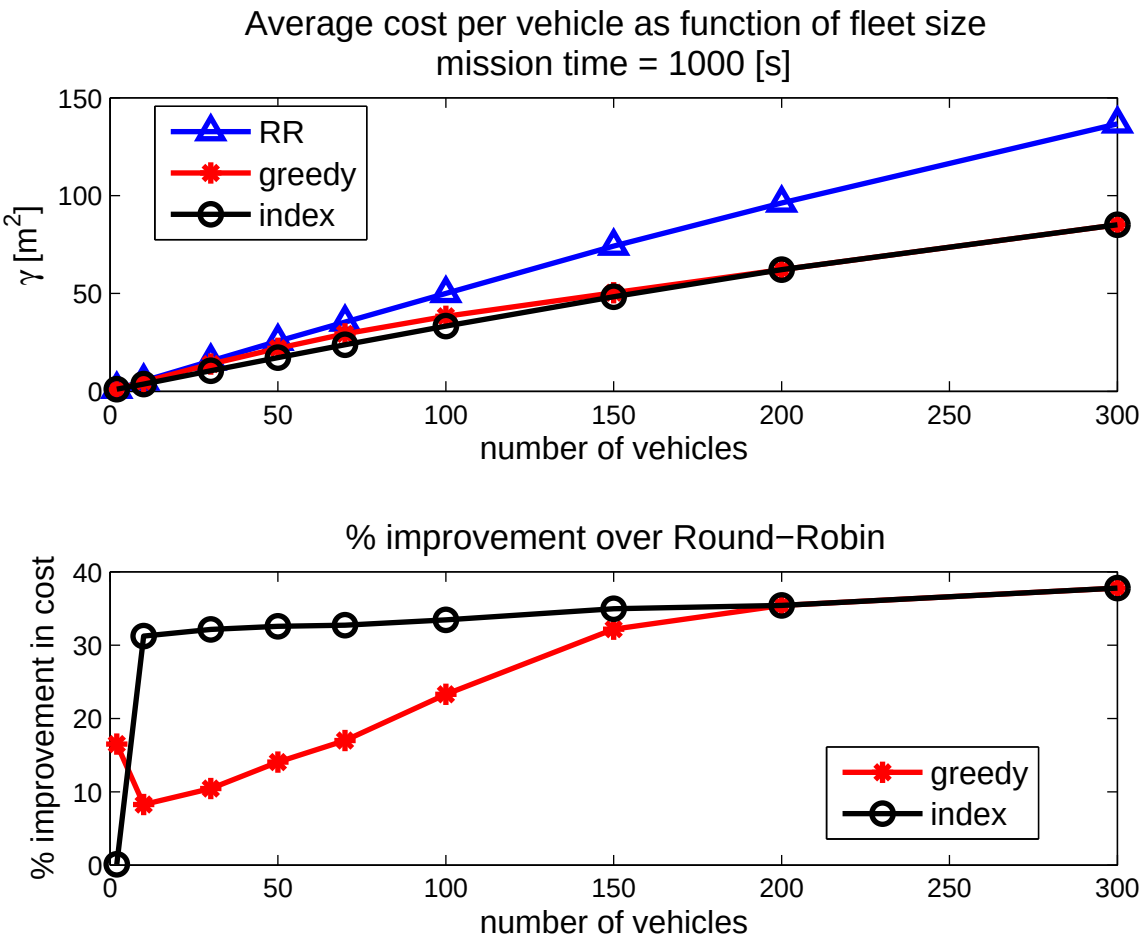


Figure 5-2: Case 2: heterogeneous vehicles with varying process noise and low constant measurement noise, mission time $T_f = 1,000$ seconds. The index algorithm achieves large gains over RR for fleets larger than 2 vehicles. Due to the short mission length, the greedy heuristic approaches the performance of the index algorithm for large fleet sizes, illustrating the value of exploitation for short horizons.

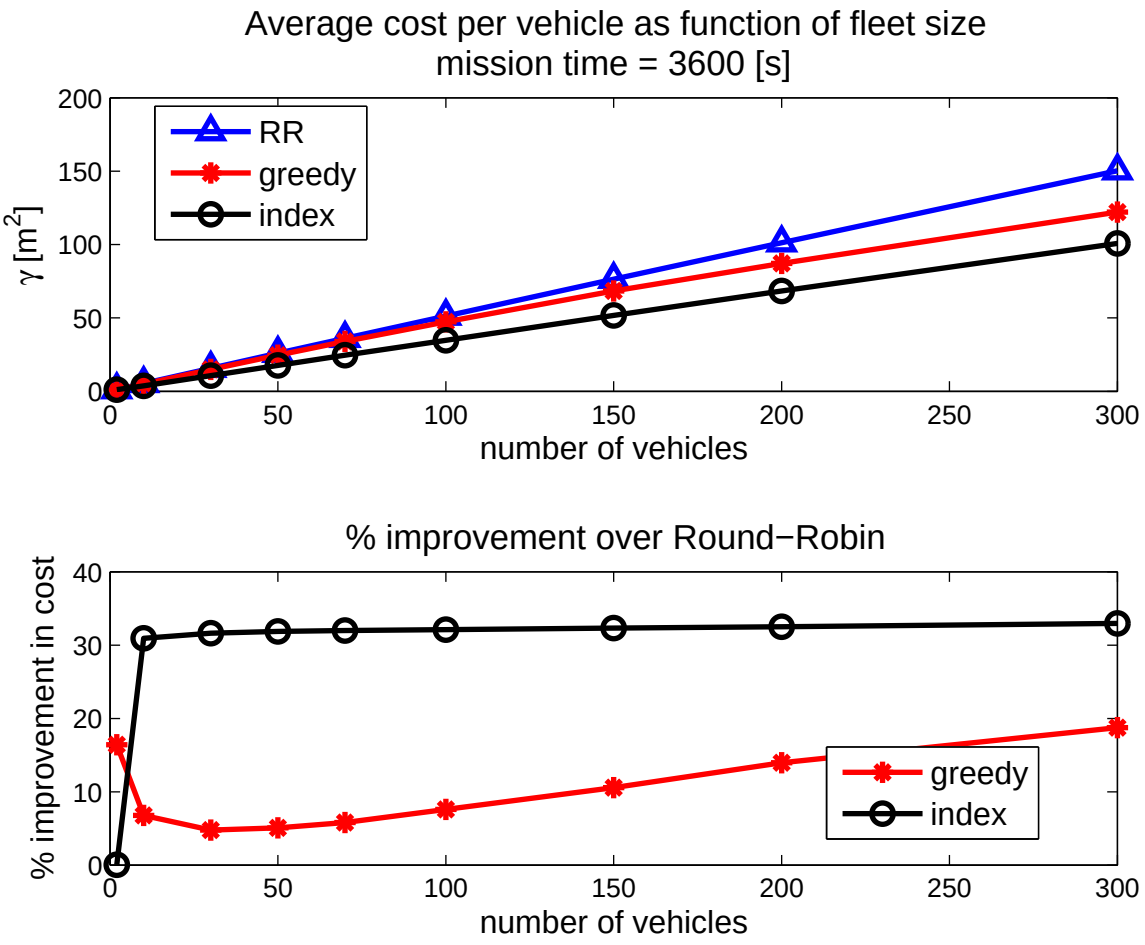


Figure 5-3: Case 2: heterogeneous vehicles with varying process noise and low constant measurement noise, mission time $T_f = 3,600$ seconds. For a moderate mission length the greedy heuristic begins to improve with large fleet sizes, but the index algorithm is significantly better, with nearly constant 30% improvement over RR for fleets larger than 2 vehicles.

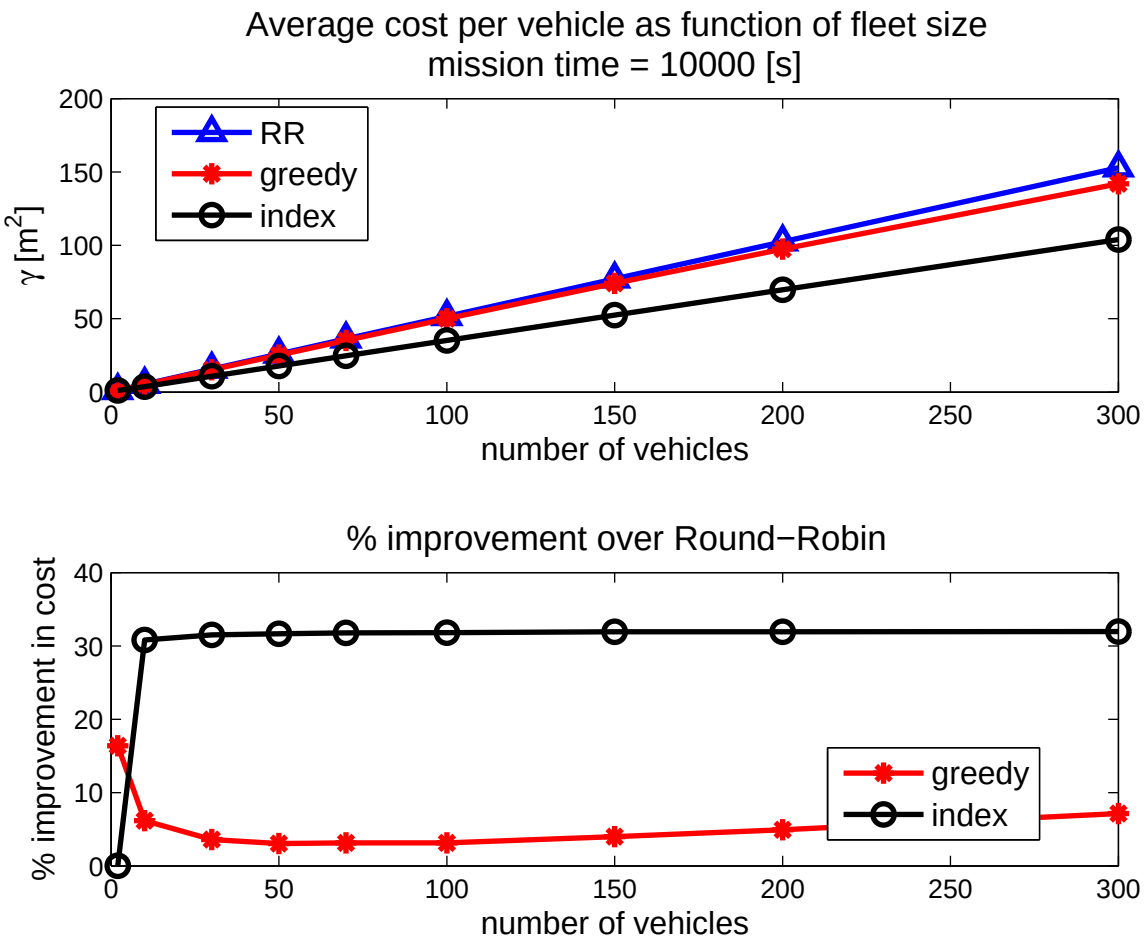


Figure 5-4: Case 2: heterogeneous vehicles with varying process noise and low constant measurement noise, mission time $T_f = 10,000$ seconds. For long missions, the benefit of the index algorithm is notable, as the infinite-horizon assumption is reasonably met and the pure exploitation strategy of the greedy heuristic performs poorly. The index achieves nearly constant 30% improvement over RR, with significant improvements of up to 25% over the greedy heuristic for fleets larger than 2 vehicles.

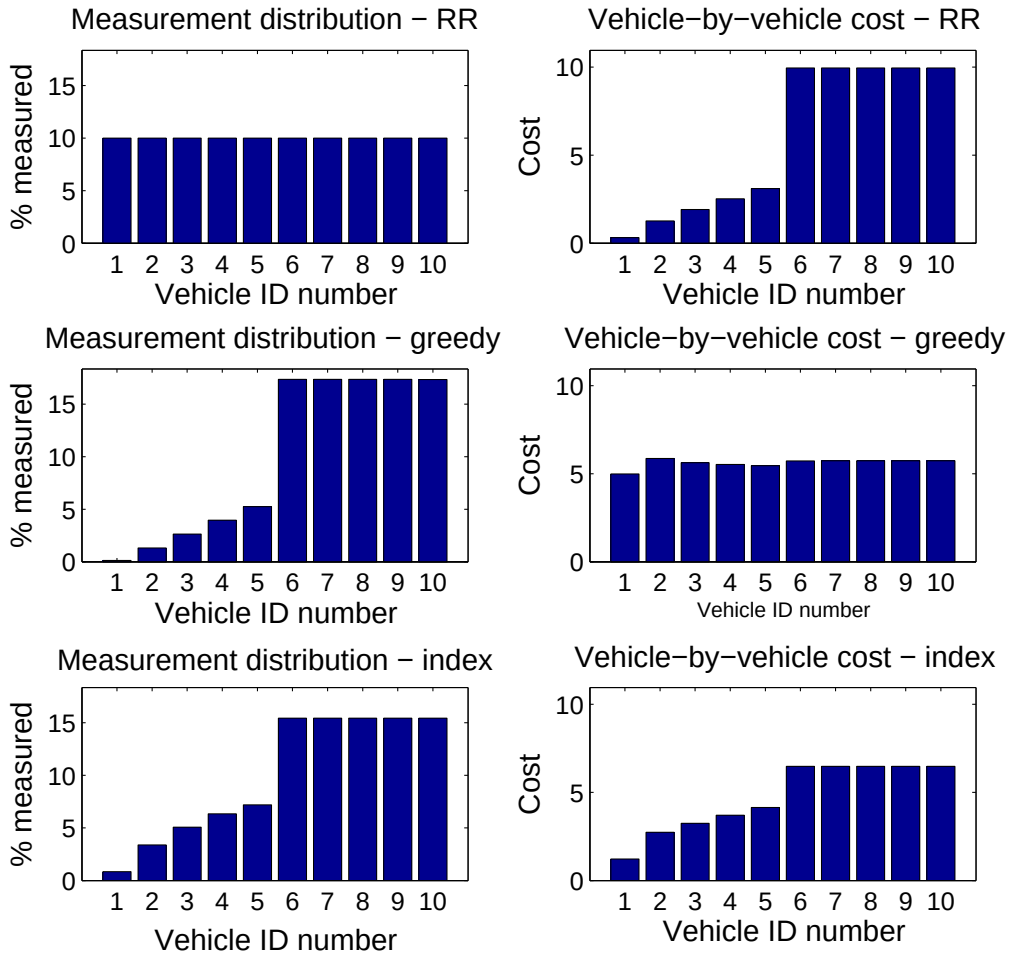


Figure 5-5: Case 2: measurement and cost distributions, $N = 10$ and $T_f = 10,000$ seconds. The left column shows the measurement distribution—the percentage of total measurements given to each of the 10 vehicles by the scheduling policy. The right column shows the corresponding contribution to the total cost of each vehicle, given the scheduling policy. The rows correspond to the RR, greedy heuristic and RBKF index algorithms. Small changes in the measurement distribution for the greedy and index algorithms result in large changes in cost contributions.

5.1.3 Case 3: Fleet of Vehicles With and Without Dead-Reckoning, Varying Measurement Noise

Here we consider a similar scenario with some vehicles with DVL and some without, but include varying measurement noise. In a similar manner to Case 2, we model the vehicles with DVL with process noise varying from $W = 0.01$ to $W = 0.5$, and the vehicles without DVL with process noise of $W = 2$. While accurate analysis of real oceanographic missions can be conducted based on actual mission operation plans, here we simulate a scenario where the vehicles with DVL are near the seafloor and are thus far away from the USBL on the ship. The vehicles without DVL are operating in the mid-water column and are much closer to the USBL on the ship (or the ship position is chosen to locate the USBL closer to vehicles without DVL). The measurement noise for vehicles with DVL ranges from $V = 400$ to $V = 200$ ($V = 400$ is representative of a 0.3 degree error at 4,000 m range), while the measurement noise for vehicles without DVL is set at $V = 50$.

Fig. 5-6 shows results from the short mission, $T_f = 1,000$ seconds, and the Fig. 5-7 shows results from the long mission, $T_f = 10,000$ seconds. For the short mission, the performance of the greedy heuristic is nearly identical to that of the RBKF index, showing that the index is choosing to perform mostly exploitation. The exploitation strategy clearly has large benefits over the RR baseline, with improvements increasing with fleet size up to nearly 40%. The long mission shows very different results. Pure exploitation is no longer a beneficial strategy since the horizon is longer. The greedy algorithm shows degenerate performance, actually performing worse than the RR baseline for all fleet sizes. The RBKF index shows improvements of roughly 10% over RR (and larger improvements over greedy). The overall improvements of the index over RR are smaller than in other cases, because the optimal strategy includes more exploration (which is what RR performs exclusively).

Again, we take a closer look at $N = 10$ case, for the long mission, $T_f = 10,000$ seconds. The measurement and cost distributions for the fleet are given in Fig. 5-8. The extreme exploitation of the greedy heuristic results in a nearly flat cost

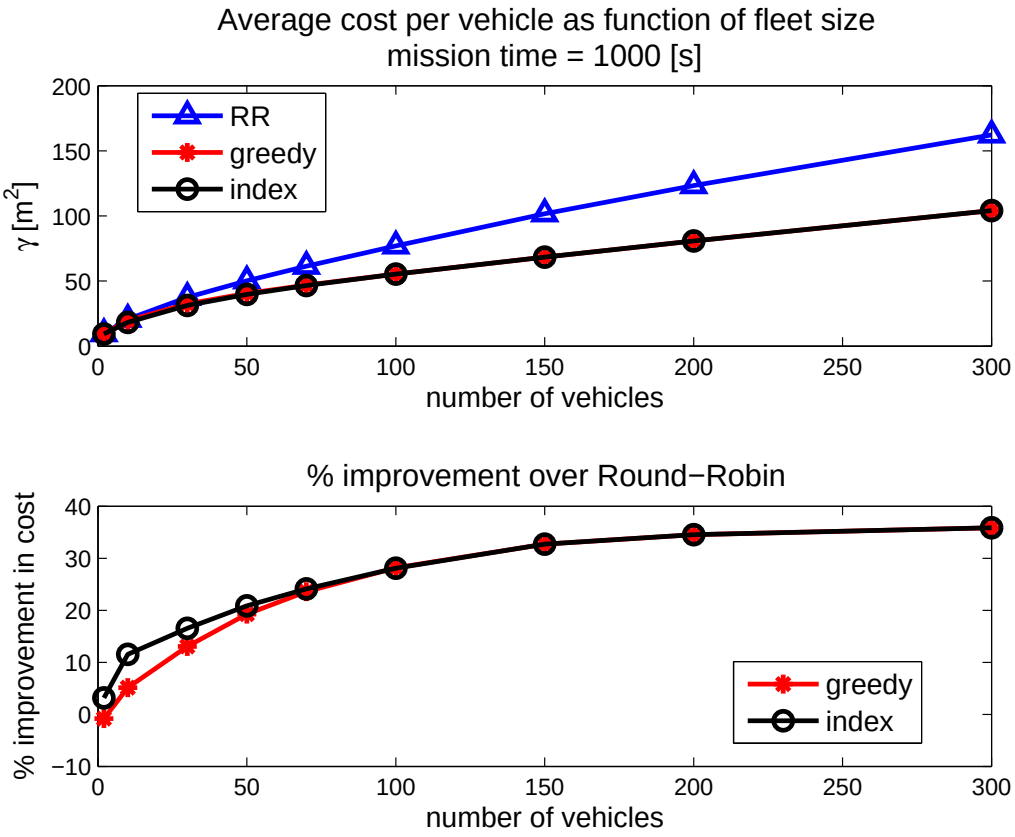


Figure 5-6: Case 3: 1,000 sec mission. Fleet of vehicles with and without DVL, varying measurement noise. Results show the RBKF index performs exploitation, and both the RBKF index and the greedy heuristic have similar, and significant, improvements over RR.

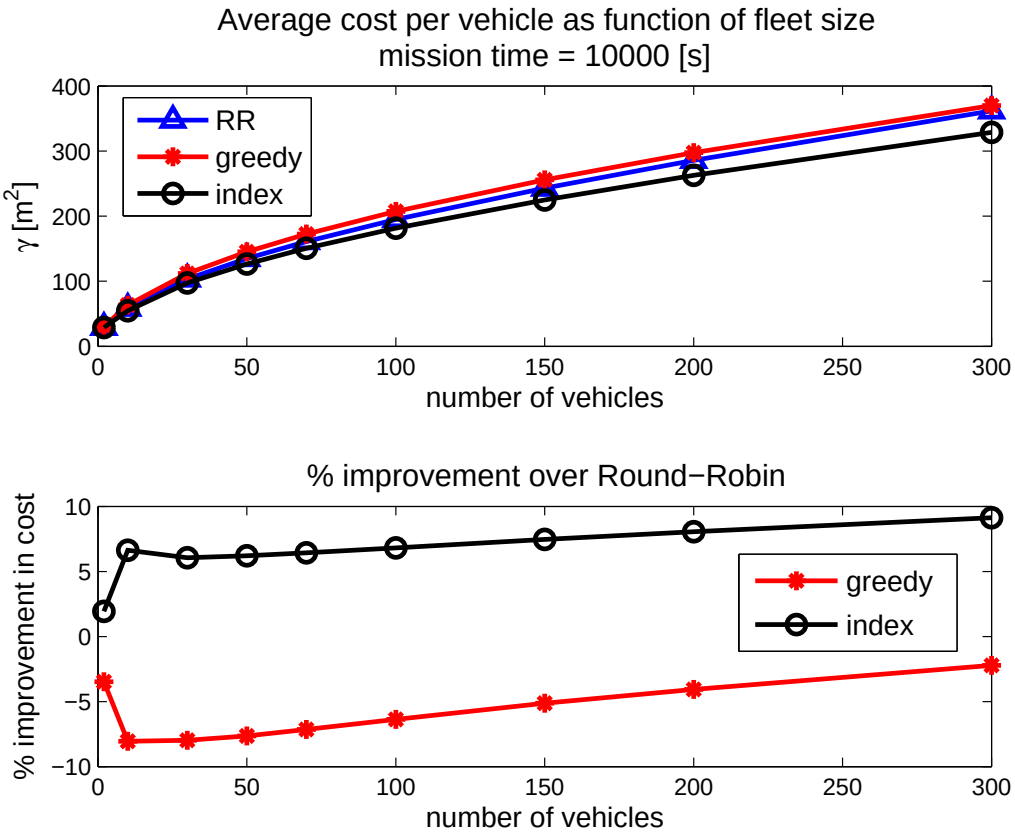


Figure 5-7: Case 3: 10,000 sec mission. Fleet of vehicles with and without DVL, varying measurement noise. Results show degenerate performance of the greedy heuristic, and moderate improvements of the RBKF index over RR.

distribution, while the RBKF index policy results in a cost distribution closer to that of RR in this case.

A snapshot of the actual measurement policy given by the greedy and RBKF index methods is shown in Fig. 5-9. In the top plot, observe that the greedy heuristic waits 1,000 seconds between measurements of vehicle 1, giving most measurements to vehicles $5-10$ due to higher process noise. The RBKF index algorithm measures vehicles $1-5$ more often, and the measurement schedule can be viewed in a much shorter time window.

The decision making strategies employed by the greedy and RBKF index algorithms are illustrated by looking at the covariance evolution of the individual vehicles, shown in Fig. 5-10. We can see that the greedy heuristic tries to keep the covariance of all of the vehicles below a common upper bound. It takes vehicle 1 a very long time to get measured, due to very low process noise and thus slow growth of the error variance. In contrast, the RBKF index is not making choices based solely on the instantaneous variance—it is minimizing the infinite-horizon cost integral. Thus, the RBKF index policy results in vehicles 1-5 operating at different covariances. The RBKF index is attempting to keep the *index values* of the different vehicles roughly constant, as shown in Fig. 5-11. Some transients are visible at the beginning, notably it still takes vehicle 1 a long time before its first measurement, however the algorithm operates in steady-state for much of the 10,000 second mission. The transient is much shorter than that of the greedy heuristic, shown in Fig. 5-10(a), demonstrating the non-myopic scheduling method of the RBKF index. The uneven spikes visible at the top of the index region are artifacts of discretization.

5.2 Finite-Horizon VGR Application with Depth-Varying Parameters

The subsea equipment delivery application using Vertical Glider Robots requires special modifications to the RBKF index algorithm. For one, the mission by definition

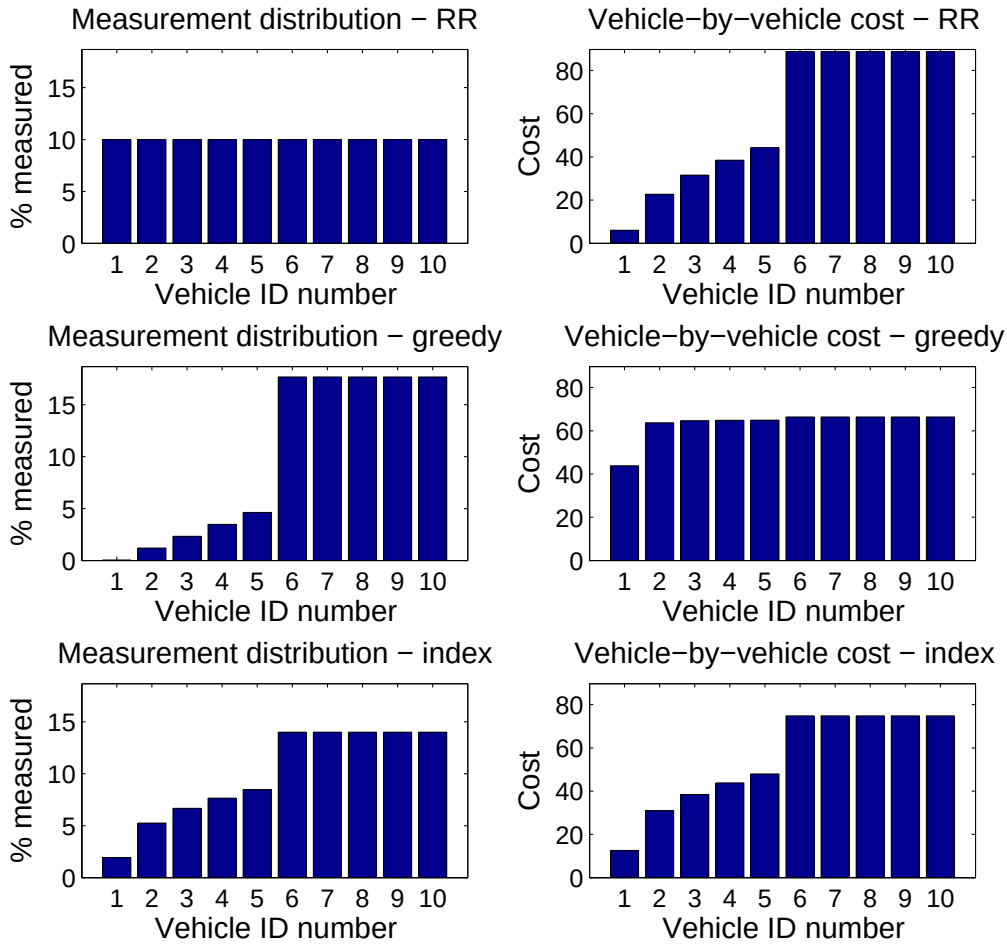
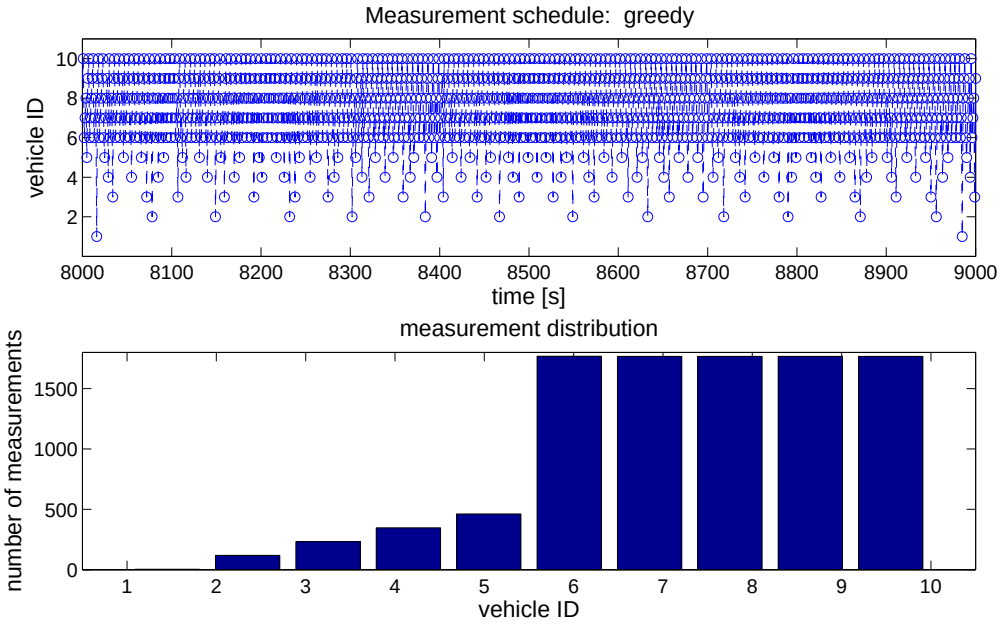
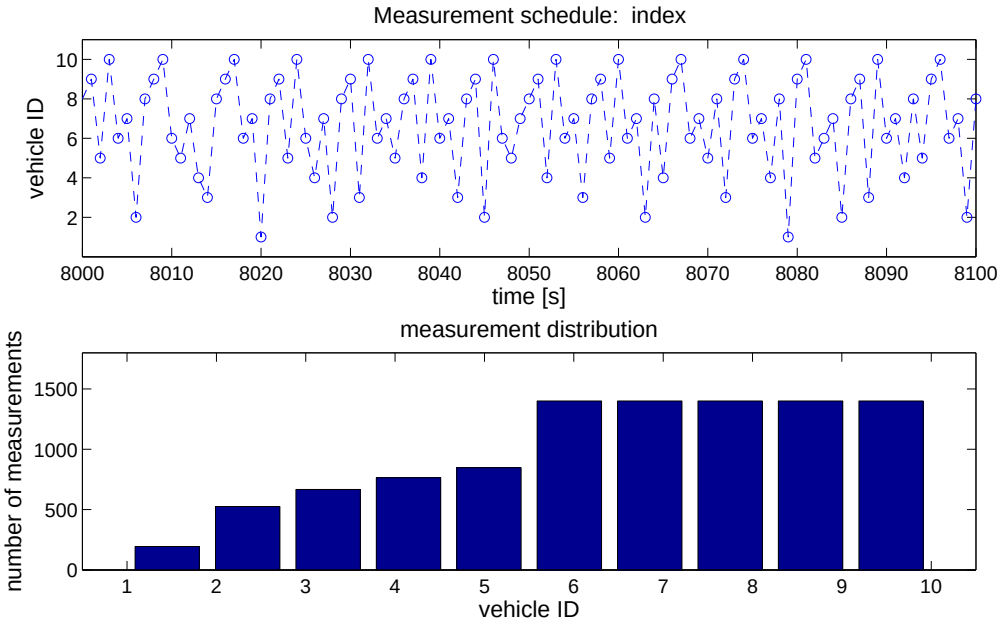


Figure 5-8: Case 3: measurement and cost distributions, $N = 10$ and $T_f = 10,000$ sec. The left column shows the measurement distribution—the percentage of total measurements given to each of the 10 vehicles by the scheduling policy. The right column shows the corresponding contribution to the total cost of each vehicle, given the scheduling policy. The rows correspond to the RR, greedy heuristic and RBKF index algorithms.

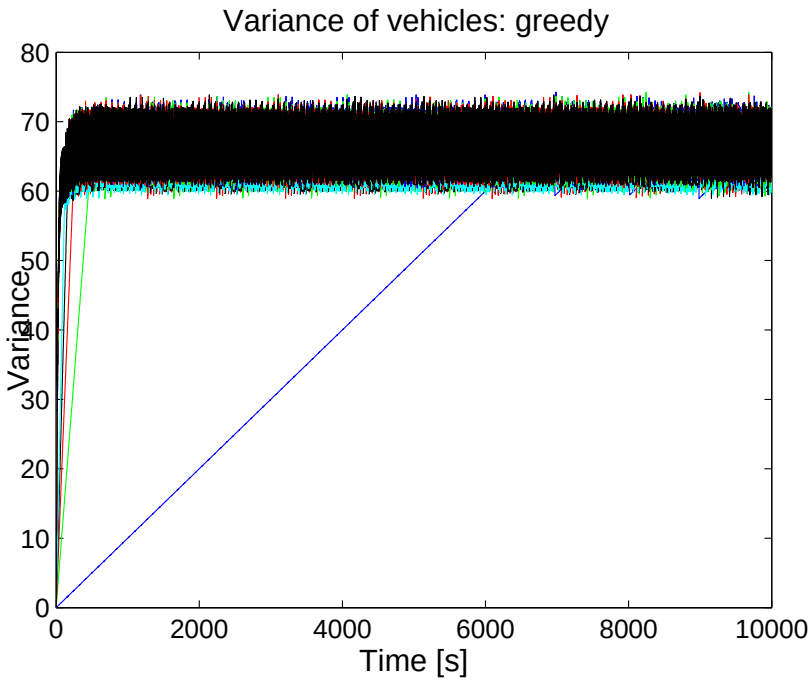


(a) Greedy

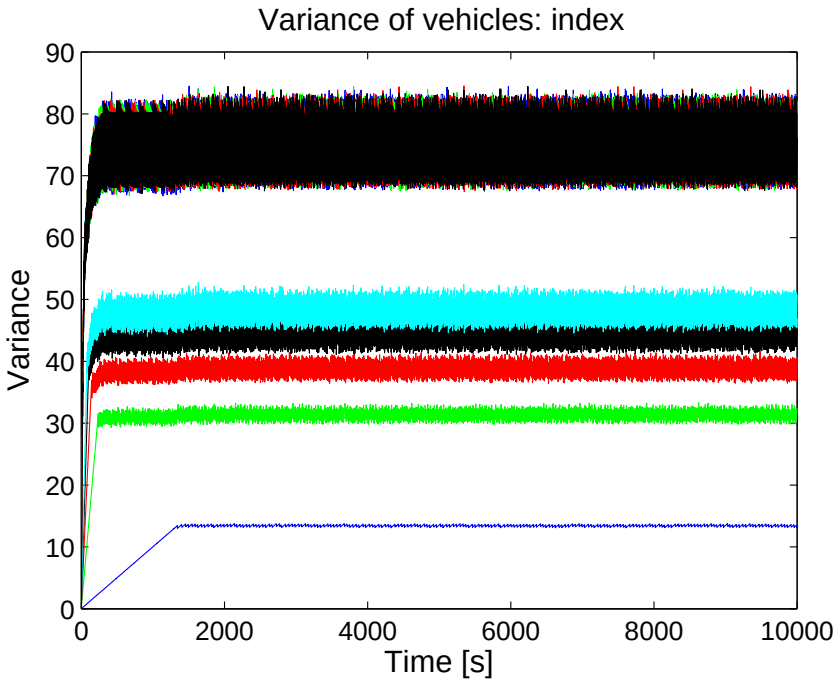


(b) Index

Figure 5-9: Case 3: measurement schedules from the greedy heuristic and the RBKF index algorithm, $N = 10$ and $T_f = 10,000$ seconds. The upper plots show which vehicle is measured at each time step. The time windows on the upper plot are selected to show roughly one measurement cycle, the greedy heuristic takes much more time in between measurements of the least-frequently-measured vehicle, $i = 1$. The bottom plots show the number of times each vehicle is measured in total during the mission.



(a) Greedy



(b) Index

Figure 5-10: Case 3: covariance evolution, $N = 10$ and $T_f = 10,000$ seconds. The greedy algorithm attempts to keep all of the vehicle covariances at a similar level. The RBKF index algorithm allows different vehicles to operate in different covariance neighborhoods, for a lower net tracking cost.

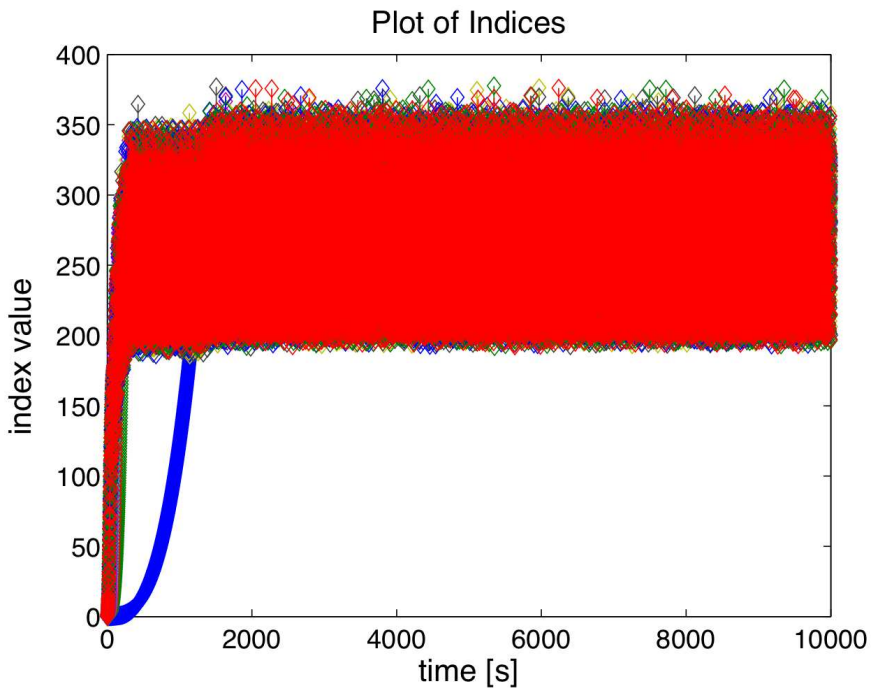


Figure 5-11: Case 3: index evolution. The RBKF index algorithm attempts to keep the index values of all of the vehicles at a similar level. The transient is much shorter than that of the greedy heuristic, shown in Fig. 5-10(a).

has a finite-horizon—once the vehicles reach the bottom, they remain at their landing position. Additionally, vehicles drop at a nominally constant rate, so over the length of an individual vehicle drop, the sensor noise from the USBL increases monotonically due to increasing distance from the ship. We 'bend' the assumptions of the RBKF index algorithm for use in a more accurate (non-LTI) simulation of VGR deployment. This simulation is intended to capture the principal challenges of multiple vehicle deployment of VGRs. The approach has not been to simulate three dimensional geometry, dynamics or control accurately, but rather to include enough detail in an abstract representation to capture the fundamental characteristics of the underlying sensor management problem.

In this section, we briefly restate the VGR system goals, which motivates discussion of modifications to the RBKF index algorithm for this mission. We describe the simulation framework, explain and justify the simple controller used, and give computational results. A suboptimal quasi-static approximation of the RBKF index is

shown to balance mission requirements of landing error and tracking robustness, and is tuned to the operator's desired mix of performance through the intuitive adjustment of only one parameter.

5.2.1 VGR System Goals

As described in Sec. 1.1.2, the fundamental purpose of the VGR system is to place equipment at accurate positions on the seafloor. This includes aspects of two performance criteria: accurate landing positions of each vehicle to satisfy mission goals, and satisfactory tracking during the entire descent for system robustness. Underlying all of these goals is the desire to complete the full mission in as little time as possible, which results in cost savings in terms of ship time per mission.

The VGR system uses active control through USBL navigation to enable each vehicle to properly steer to its target and compensate for unknown disturbances. Thus the landing accuracy metric is a measure of control system performance in the presence of unknown disturbances and sensor noise. As will be shown, with our proposed system architecture, low tracking error uncertainty correlates with landing accuracy.

The second performance metric is less objective and is highly related to practical operations and safety. It is not prudent in practice to allow vehicles to drop 'blindly,' as the underwater environment is notoriously dangerous and it is possible to lose vehicles due to system failures or extreme disturbances. If these situations occur when the vehicle is being tracked, problems can be identified, potential solutions can be implemented in some situations, and in the least, the operators may be able to recover a problematic vehicle because its location is known. In order for operators to trust the system enough for it to be usable in practice, the system must be robust. Thus, we desire a low tracking error uncertainty for all of the vehicles during the entire descent.

As may be evident from these descriptions, the criteria of low ship time, high landing accuracy, and safe tracking during descent are all pulling in opposing directions. However, we will show that the quasi-static application of the RBKF index

can balance these requirements effectively, with improvements over naive schemes.

5.2.2 VGR Simulation Framework

As described in Sec. 3.1.1, vehicles are deployed sequentially from the ship with a certain spacing, which is variable in the simulation. We take a '1.5' dimensional approach and model all vehicles dropping at a constant rate $\dot{z}$ straight down, with one dimensional position errors described by the scalar random walk with control model

$$z = \dot{z}t$$
$$\dot{x} = u + w_{env}$$
$$y = x + \nu_{USBL}(z)$$

Continuous time dynamics are $A = 0$ ($A = 1$ in the discrete-time simulation), with no dead-reckoning. Control u is described in Sec. 5.2.4. Process noise is set such that the expected excursion of a random walk without control (the trajectory taken by a passive lander) roughly matches empirically observed landing errors. We use a mission depth of 4,000 m, and an expected translation distance of 25 m, which results in $W = 0.156$. For the sensor noise, we transform the angular error characteristic into a Cartesian error at a given depth, using the 0.3 degree error specification common for current USBL systems [4]. This simulation approach ignores higher-order effects of varying USBL noise at different angles (for example, for vehicles traveling to the edge of the grid), varying drop speeds due to glide angle and use of control, as well as delays in position updates from the USBL. These characteristics could all be easily modeled and added to the framework, however for the purpose of comparing sensor allocation algorithms we have chosen to keep the simulation as simple as possible.

5.2.3 Modifications of RBKF index algorithm

The biggest difference between the VGR mission and the conditions under which the RBKF index algorithm is derived is the finite horizon landing accuracy metric. Additionally, the USBL measurement noise increases with depth, and is thus time-varying. One way of handling this would be to reformulate the problem as a finite-horizon shortest path problem, however if the time horizon is relatively long a stationary policy such as the RBKF index has the potential to perform well and requires less computation. We take the approach of making a couple heuristic tweaks to the RBKF index algorithm that approximate the finite horizon landing metric and time-varying measurement noise.

To encourage accurate landing position, we build on the assumption that accurate (low uncertainty) tracking will lead to accurate positioning of the vehicle. The achievable glide slope of the VGR (over 45 degrees for the prototype vehicle described in Chapter 2) and the dynamics of the onboard controller allow for large course adjustments relative to the expected drift error in short times, making the distance above the bottom from which errors are non-recoverable small. Thus, our approach is to introduce depth-varying priority weights to encourage higher accuracy tracking of vehicles which are near the bottom. Our implementation uses weights equivalent to depth z raised to some power d: $T_i = z_i^d$, where d is a tunable parameter. The logic is that vehicles in the mid water column still have a lot of time to correct for drift, and will continue to be affected by process noise during the remaining portion of their trip to the bottom. Vehicles near the bottom are closer to landing and control performance will have a large impact on the final landing accuracy. Thus, position measurements are more valuable to vehicles closer to the bottom. However, as results will show, extreme use of this priority weighting method results in less robust policies—vehicles may travel for dangerously long periods of time without receiving updates from the USBL. The mission operator can tune the parameter d in order to set the desired balance between landing accuracy and robust tracking during descent.

For handling the depth-varying weights as well as depth-varying measurement

noise, we take a 'quasi-static' approach and simply plug in the parameters for each vehicle, evaluated at the depth of that vehicle. This approach was suggested through correspondence with Dr. Le Ny, the author of [71], and while suboptimal, shows promise in simulation. Since the time-varying parameters T and V are both monotonically increasing in depth (and therefore in time for the VGR mission), it is reasonable to assume that indexability still holds, although this has not been formally verified. Essentially, the index algorithm is using a zero-order hold on the parameters during a given decision step, and computes the locally optimal solution given those parameters. The degree in which this approach is suboptimal depends on how quickly the parameters change with depth relative to the time-scales of the measurement updates. Possible improvements to more accurately incorporate the depth-varying parameters are discussed in Chapter 6.

5.2.4 Vehicle Control System in Simulation

Analysis of the tracking error uncertainty cost function as in Sec. 5.1 requires only the analytical output from the Kalman Filter. To analyze the landing error metric, we must include a stochastic simulation of the vehicle trajectories as well as a vehicle position control system. Assuming no stability issues mid-drop, the actual landing performance (as will be shown) will depend on how well the controller performs in the conditions encountered near the bottom. Individual vehicle flight controllers can be optimized to perform well in this regime of update rates and noise. The main goal of the VGR simulations is to investigate sensor allocation algorithm performance fairly between different algorithms (not to design optimal control systems), so the method used in these simulations is to use a controller that exhibits no stability issues in conditions that could be encountered in the run, and also performs consistently across various expected operating conditions.

Following on the use of a simple scalar kinematic model for vehicle dynamics, we use a simple proportional controller for position: $u = -K\hat{x}$, resulting in first-order lag behavior of the controlled system. The controller acts on the position estimate from the Kalman Filter, as described in Sec. 3.2.1, which means that there are no

stability issues since the KF removes the zero-order-hold aspect of interacting loops which can cause problems with varying update rates. Some basic z-transform analysis of the filter and controller as well as empirical observations from simulations at different update rates confirm stability of this control method. We have empirically set the discrete time proportional control gain to $K = 0.01$, which is relatively low bandwidth, but reasonable considering the entire drop takes 4,000 seconds. Most importantly, this simple controller achieves closed-loop positioning performance consistently across different delays and update periods, which allows for fair comparison between algorithms.

5.2.5 VGR Simulation Results

We now show simulation results of the 50 vehicle VGR mission in 4,000 m depth. Fig. 5-12 first demonstrates the advantages of adding real-time navigation and control, relative to passive lander deployments. There are three sets of plots which show the performance for three different controller gains: $K = 0$ (passive lander), $K = 0.01$ (gain used in subsequent simulation results), and $K = 0.05$ (a higher gain for comparison purposes). A round-robin measurement scheme is used in all three cases, and the vehicles are dropped 200 seconds apart (during steady-state operations there are 20 vehicles in the water at any given moment). The left plot of each pair shows the trajectories of all 50 vehicles for a single mission; a representative vehicle during the middle of the drop is highlighted in red. The right plot of each pair shows the analytic tracking error standard deviation from the Kalman Filter, with the same representative vehicle highlighted in red. The discrete drops in uncertainty (barely visible in this figure, but more pronounced in Figs. 5-13 and 5-14) correspond to measurements of a vehicle. The effects on landing accuracy by adding navigation and control are clearly evident. The performance of controllers with $K = 0.01$ and $K = 0.05$ is similar, although slightly higher frequency oscillations are visible with the larger gain.

Next, we compare the performance of the navigation and control system when using the RR, greedy and RBKF index algorithms for allocating USBL measurement

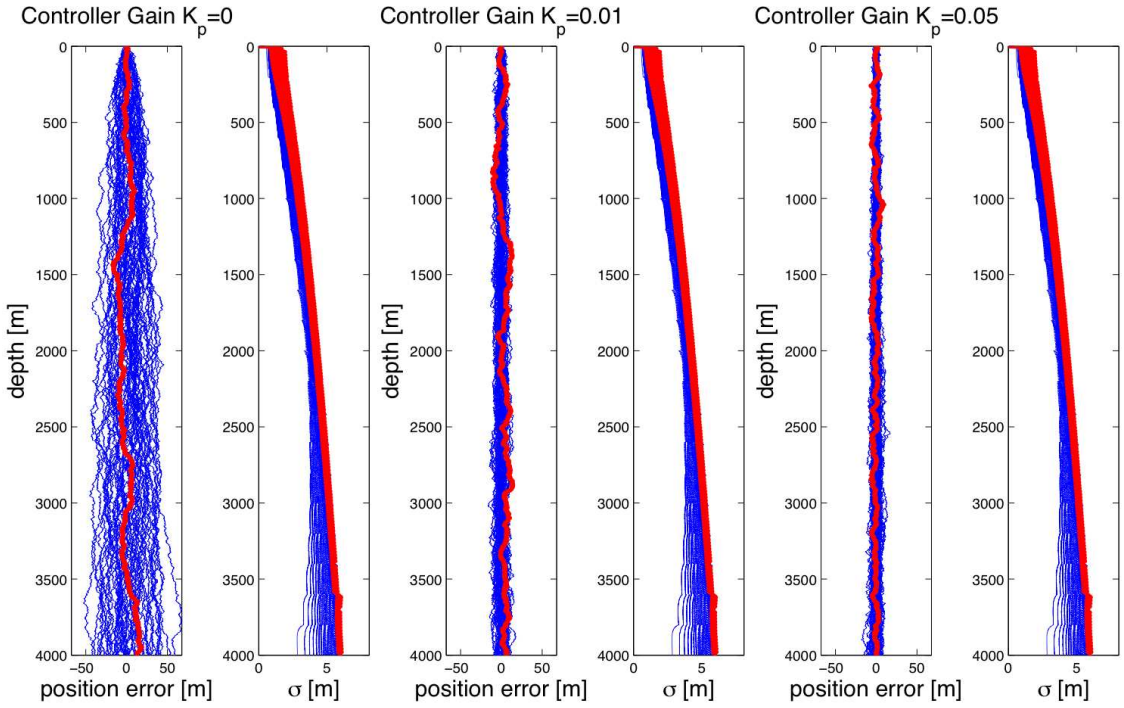


Figure 5-12: Performance with simple RR tracking and various proportional controller gains. Vehicles are deployed sequentially, 200 seconds apart. Blue lines are the trajectory of all 50 vehicles as a function of depth. The left plot of the pairs shows horizontal position (simulated), and the right plot shows the tracking uncertainty as predicted analytically by the KF. Landing accuracy is greatly improved by adding real-time control.

updates. We have experimented with many different values for the parameter d in the depth weighting $T(z)$, and a reasonable balance of tracking during descent and landing accuracy is achieved with $d = 4$, such that $T_i = z_i^4$. Vehicle trajectories and uncertainty evolution are plotted as a function of depth for the three different algorithms in Fig. 5-13. The vehicles are deployed with 200 second spacing, and a gain of $K = 0.01$ is used. Since the USBL measurement noise increases with depth, the tracking error uncertainty increases with depth when the RR scheduling policy is used. This increase in covariance between the vehicles is evident in the vehicle trajectories, as the 'cone' of trajectories grows with depth. On the right, the greedy algorithm exhibits opposite behavior. Vehicles near the bottom are given very high priority for measurements, and since there are a finite of measurements available, vehicles near the surface are given fewer measurements. From the red line on the rightmost plot, we see that a vehicle during steady-state operation travels over 1500 m before receiving its first measurement update. This results in large drift for vehicles when in the upper half of the ocean, and decreasing covariance in trajectories near the bottom. Better landing accuracy than the RR algorithm comes at the expense of a large worst-case tracking uncertainty in the middle of the drop. The index algorithm in the middle is still trying to minimize the infinite-horizon cost integral (with the modifications of depth-varying parameters), and thus the worst-case uncertainty is much lower than with the greedy algorithm. However, the index still allocates more measurements to vehicles near the bottom than RR, resulting in better landing accuracy than RR (but not as good as greedy).

The tradeoff between landing error accuracy and robust tracking during the descent is similar to the exploration versus exploitation tradeoff. Round-robin performs maximum exploration, and greedy performs maximum exploitation. The depth priority weighting parameter d can be used by the mission operator to adjust the performance of the index algorithm towards once metric or another. In one extreme, the index algorithm can prioritize tracking during the whole descent by setting $d = 0$, which results in a round-robin scheme. In the other extreme, the weighting can be set to increase very drastically with depth, prioritizing accurate tracking for vehicles

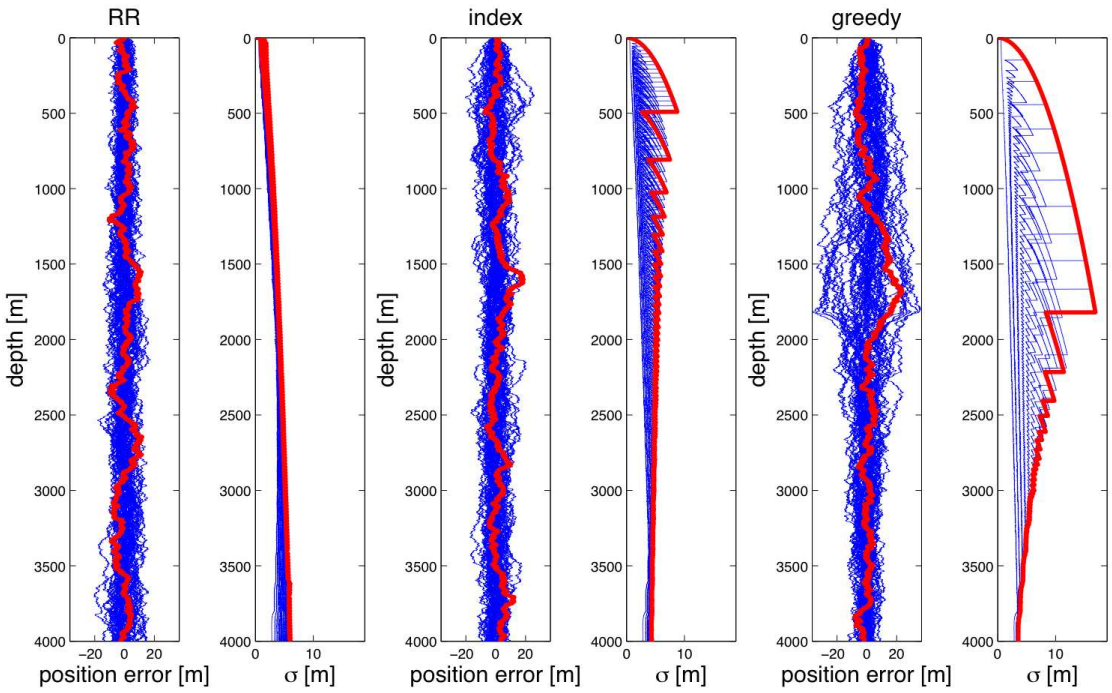


Figure 5-13: Performance of index, RR and greedy algorithms with weighting function $T = z^4$. Vehicles are spaced 200 seconds apart. The blue lines are the trajectory of all 50 vehicles as a function of depth. The left plot of the pairs shows horizontal position (simulated), and the right plot shows the tracking uncertainty as predicted analytically by the KF. Jumps in the uncertainty correspond to measurements of that vehicle.

near the bottom at the expense of tracking during descent. To demonstrate this, we give an extreme example, with $d = 20$, shown in Fig. 5-14. Here, the index begins to approach the greedy policy, however still attempts to balance the two metrics. When using the greedy scheduling policy, a vehicle in steady-state operation travels almost 3/4 of the way to the bottom before receiving a measurement update; with the RBKF index, vehicles receive the first measurement roughly halfway down.

Obviously, if the operator only cares about one metric, the specific use of either the RR (for robust tracking), or the greedy algorithm with an extreme weighting function (for accurate landing under the assumptions of well-behaved vehicles and known environmental conditions) will give the best results. However, the RBKF index algorithm gives a good solution when a balance between the two metrics is desired, and this balance can be tuned using the parameter d.

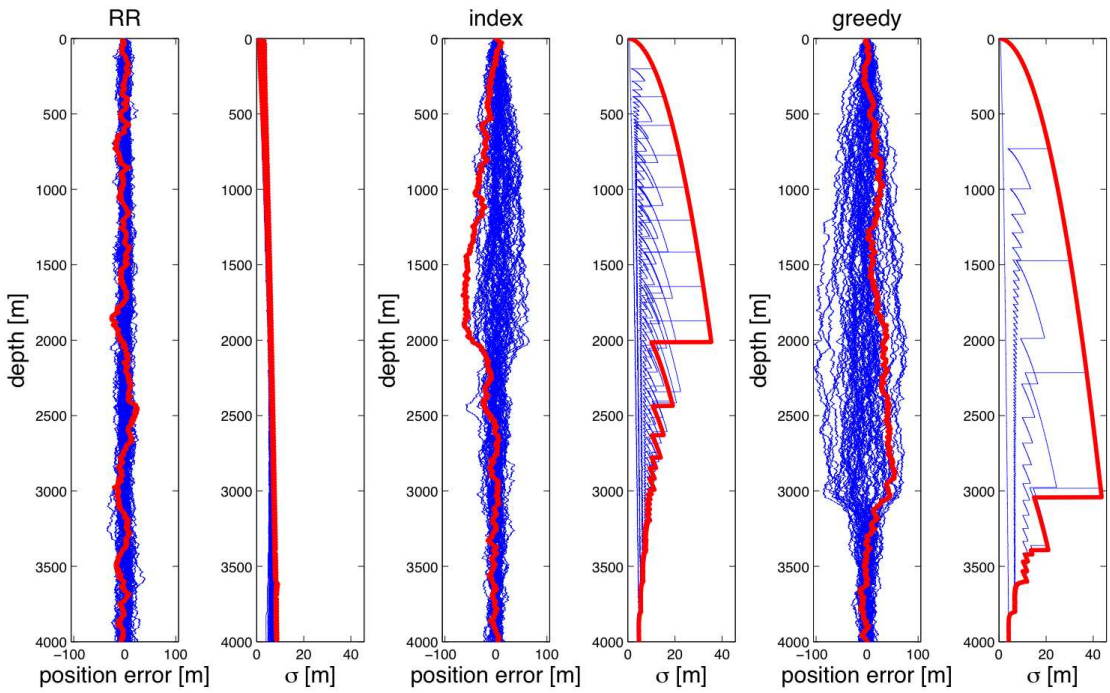


Figure 5-14: Performance of index, RR and greedy algorithms with the extreme weighting function $T = z^{20}$. Vehicles are spaced 200 seconds apart. The blue lines are the trajectory of all 50 vehicles as a function of depth. The left plot of the pairs shows horizontal position (simulated), and the right plot shows the tracking uncertainty as predicted analytically by the KF.

Results comparing the algorithm performances as a function of the spacings between vehicles are given in Fig. 5-15. These comparisons are performed using 50 vehicles, a depth of 4,000 m and a weighting function $T = z^4$. The horizontal axis shows spacings between vehicles in seconds, which scale with the total ship time necessary to complete the entire mission (time to drop all 50 vehicles). The top plot shows the worst-case tracking uncertainty of any vehicle over the entire mission, which is a measure of the robust tracking during descent performance metric. The middle plot shows the analytical tracking uncertainty as predicted by the Kalman Filter at the time of landing (averaged across the whole fleet). The bottom plot shows the RMS landing error of the fleet as computed by the stochastic simulation (averaged over 300 Monte-Carlo trials). The similar shape of the middle and bottom plots supports our assumption of accurate tracking leading to good control performance. The difference in magnitudes between the middle and bottom plots shows that there are differences between the predicted performance and actual performance due to the controller (as expected).

At very low spacings, all algorithms approach a round-robin scheme, since there are no differences in parameters between the vehicles due to the whole fleet dropping simultaneously (all vehicles are at the same depth at any given time). At very long spacings, performance approaches that of a single vehicle drop—a 4,000 second spacing means one vehicle is in the water at a time and thus receives all possible USBL updates. This represents a lower bound on performance of the scheduling algorithms, and is an indication of the control system performance given the noise parameters. In intermediate spacings, we see there are gains to be made by using the greedy and RBKF index algorithms, depending on the desired performance metric. The maximum improvement in landing error performance compared to RR occurs with spacings in the 200-300 second range, where the RBKF index algorithm shows 15% improvement, and the greedy heuristic is gives 25% improvement. The greedy algorithm however exhibits much higher worst-case tracking uncertainty than the index algorithm. This indicates that, depending on the performance metric tradeoff desired, the index algorithm can balance the two metrics well. For the VGR mission,

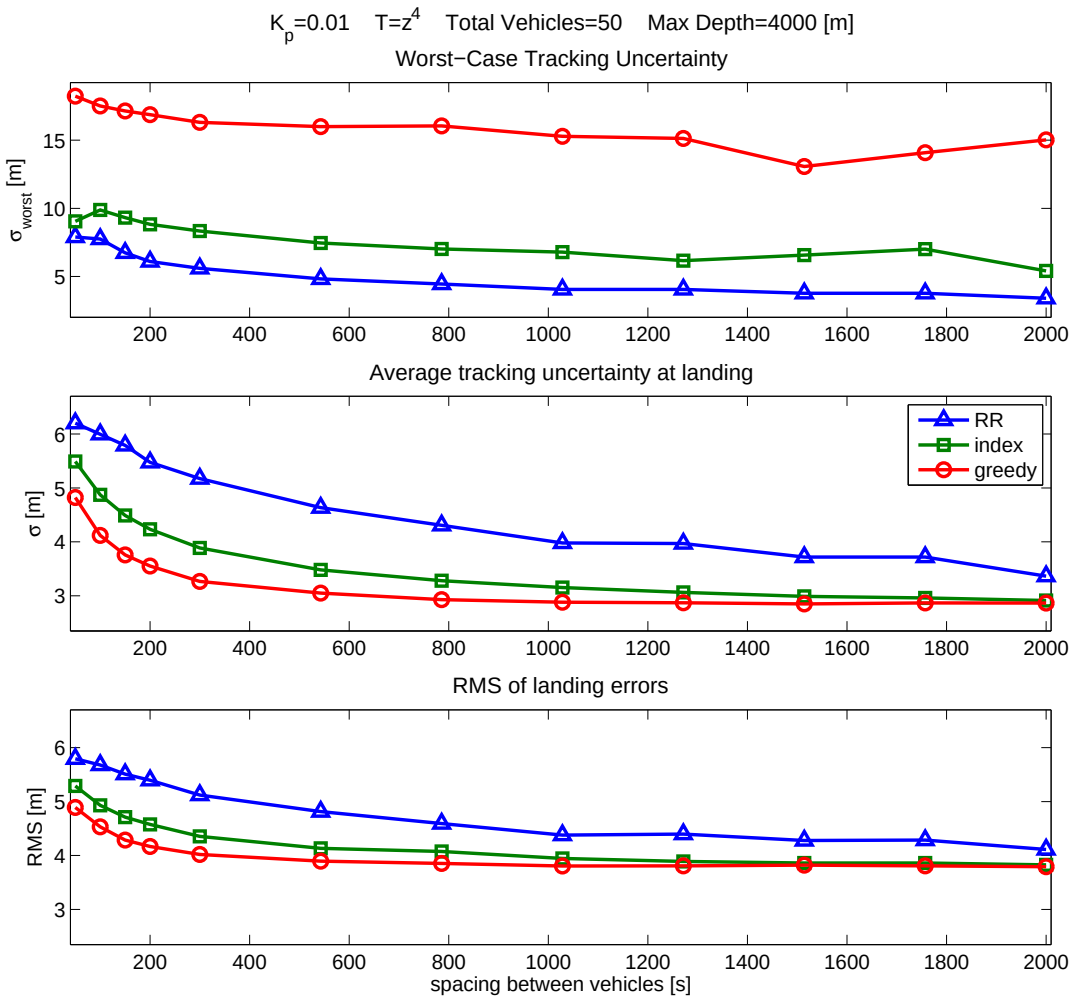


Figure 5-15: Algorithm performance as a function of spacing in between sequential vehicle drops.

future work could use the general techniques described in Sec. 6.2 to accurately incorporate the depth-varying parameters and finite-horizon landing metric. Additionally, design tools could be developed that account for tradeoffs in vehicle spacing, fleet size, expected accuracy, expected worst-case tracking, and ship time to explore the design space and help mission operators make decisions about key parameters.

5.3 Summary

We have shown computational results comparing round-robin, greedy heuristic and Restless Bandit Kalman Filter (RBKF) sensor scheduling algorithms. The first section examined the performance for mission scenarios with heterogeneous LTI vehicles, including two cases of fleets containing mixtures of vehicles with and without DVL-based dead-reckoning capabilities. In these examples the RBKF index algorithm performs well, especially in long missions where balancing exploration and exploitation is important. The greedy heuristic performs well in some short missions where exploitation is the preferred strategy, but shows degenerate performance in other cases. In all LTI cases considered the index algorithm has proved the best choice for fleets larger than 2 vehicles. The second section demonstrated the application of the RBKF index scheduling policy for the VGR subsea equipment delivery mission. A quasi-static approximation allows for handling of depth-varying parameters such as sensor noise as well as a priority weighting heuristic used to address the landing accuracy performance metric. While suboptimal, this method shows benefits in balancing landing accuracy with robust tracking, allowing mission operators to easily tune the scheduling policy to their desired performance and total mission time. Overall, the combination of potential benefits, low likelihood of degenerate performance, and low computational cost makes the RBKF index an attractive solution for multi-vehicle tracking with constrained sensors.

Chapter 6

Conclusions and Future Work

Accurate geo-referenced navigation is important for underwater vehicle operations, and future capabilities of ocean systems will be enhanced by the deployment of large multiple vehicle fleets. Centralized navigation systems such as a USBL sonar onboard a ship are popular, convenient and economical options for providing, or augmenting, position estimates to vehicle control systems. However, these navigation sensors represent a constrained resource due to physical limitations of the sensor and the acoustic channel on which it relies. We have studied methods for allocating navigation updates among multiple vehicles with different dynamics, noise properties, and priorities. In particular, we have investigated the use of non-myopic scheduling policies based on Restless Multi-Armed Bandit theory, including a specific Kalman Filter multi-vehicle tracking algorithm given in [71]. We give a short summary of the work in this thesis, and conclude with future directions and broader uses of Restless Bandit scheduling algorithms in ocean applications.

6.1 Summary

Multiple vehicle deployments offer special challenges for underwater navigation. The sharing of a centralized geo-referenced navigation system among multiple vehicles allows for the design of individual vehicles suitable for economically scalable fleet sizes, due to the low cost of the required onboard navigation sensors. We give an example

of a vehicle design that fits this philosophy: the Vertical Glider Robot concept for subsea equipment delivery. In Chapter 2 we present model-scale proof-of-concept prototype tests of this vehicle, demonstrating accurate localization with minimal onboard sensing.

USBL or similar drift-free navigation systems can be incorporated into vehicle onboard control systems through the use of a Kalman Filter or similar estimator. In Chapter 3 we outline the general approach, which can be used both for vehicles that rely solely on the USBL for position measurements, and for augmenting the navigation of vehicles capable of dead reckoning by compensating for drift. We formulate multiple vehicle Kalman filter tracking as an infinite-horizon average cost problem for the optimal scheduling policy, and describe simple heuristic approaches.

The curse of dimensionality is a major challenge for optimal non-myopic scheduling policies; however problems that fit the Multi-Armed Bandit structure are made computationally tractable through the use of a priority index scheduling policy that balances exploration and exploitation. In Chapter 4 we first give a tutorial introducing Multi-Armed Bandit theory, including an extension known as Restless Bandits which can handle dynamic systems such as underwater vehicles. We give an explanation of the index policy derived by Whittle in [108] for Restless Bandits, and show its applicability to the Kalman Filter tracking problem. We discuss the Restless Bandit Kalman Filters (RBKF) algorithm from Le Ny et al. in [71] which builds on Whittle's approach, and show how it can be easily incorporated into a multiple vehicle tracking system.

While the theoretical elegance of Multi-Armed Bandit theory is by itself useful for developing intuition regarding decision-making and information acquisition problems, we aim to demonstrate the usefulness of these methods for multiple vehicle tracking. In Chapter 5 we present simulation results comparing the performance of the RBKF index algorithm with the round-robin baseline as well as a greedy heuristic. Using simple scalar kinematic vehicle models we investigate algorithm performance for a variety of mission scenarios.

We consider infinite-horizon tracking of heterogeneous fleets of LTI vehicles, in-

cluding two idealized examples of fleets with some DVL-equipped vehicles, and some vehicles incapable of dead-reckoning. The RBKF index performs as well or better than the other two methods, and in certain cases offers improvements of up to 40%. The index method performs well in cases where the greedy algorithm or round-robin algorithm perform well, adjusting the policy to favor exploitation or exploration as appropriate. Additionally, the index method does not show degenerate performance when compared to the round-robin baseline, as is sometimes the case with the greedy heuristic.

We also investigate the performance of scheduling algorithms for simulated subsea equipment delivery missions of vehicles such as the VGR. A suboptimal quasi-static approximation of the RBKF index algorithm is used to handle depth-varying sensor noise and priority weightings. This algorithm is shown to effectively balance the VGR mission requirements of landing accuracy and robust tracking through the use of a mission-tunable heuristic, and we use the exploration versus exploitation tradeoff as well as the effects of mission horizon to explain the strengths and weaknesses of this modified algorithm.

Compared to commonly used heuristics, the combination of potential benefits, low likelihood of degenerate performance, and low computational cost makes the RBKF index an attractive solution for multi-vehicle tracking with constrained sensors. Additionally, the RBKF index is based on sound theory from mathematical optimization, from which further extensions to the method can be derived.

6.2 Future Work

There are a number of potential improvements to this work, that either modify the Restless Bandit theory in order to better capture time-varying aspects of the problem, or extend the use of Restless Bandit-based scheduling to broader applications in the ocean.

One approach for handling time-varying parameters is to fundamentally derive the Restless Bandit index of Whittle [108] using time-varying (non-autonomous) dy-

namics. This has been briefly investigated, but the full solution is future work. The multidimensional system solution from Le Ny et al. [71] gives open-loop periodic scheduling policies determined through solution of a semidefinite program (SDP) using linear matrix inequalities, and thus involves significantly more computation *a priori* compared to the scalar solution. However, if conditions are stationary, the SDP can be run just once, before the mission. Multidimensional systems could be used to incorporate more accurate dynamic vehicle models into the tracking framework, as well as handle noisy velocity measurements (such as from a DVL). The delayed-state filtering approach of [94] could easily be incorporated into the tracking method. Additionally, multi-state models could be used to handle the depth-varying parameters in the VGR case; the parameters would be augmented states that are functions of the depth. This approach could potentially be used for time-varying parameters as well, provided that indexability can be verified.

The multidimensional formulation allows for many potential extensions, however there are also benefits of the scalar closed-form index used in this thesis. The scalar index allows for closed-loop and transient implementation, making it more robust to model errors and changing parameters. For certain parameters such as the depth-varying noise of the VGR case, the time-varying aspects are known and can thus be modeled and planned for. However, it is easy to imagine other scenarios where parameter variations are unknown before the mission. These could be situations where human-in-the-loop operators change the priority of navigation accuracy for different vehicles in real time depending on changing mission priorities, or collaborative/adaptive missions where the goals of the vehicle fleet change based on observed conditions. The closed-loop index allows for adjustments to parameters in a manner similar to the quasi-static method used in the VGR case—however, in the case of *a priori* unknown or reactively-adjusted parameters this method may not be so suboptimal. The RBKF index will still attempt to balance present rewards with predicted future rewards given the parameters used (which would be the best-known parameters to the operator at that moment). The decision of whether to use the closed-loop scalar index versus the full multidimensional case would depend on the

size of the problem and the time-scales involved—whether it is reasonable to re-run the semidefinite optimization to adjust the open-loop scheduling policy whenever parameters change, versus just plugging the new parameters into the scalar closed-form solution.

Another direction of potential research is to apply Multi-Armed Bandit theory to related problems in underwater navigation and autonomy. The exploration versus exploitation tradeoff shows up in many fundamental decision-making and information acquisition problems, and thus is applicable to a broad mix of scenarios. One extension could be to include RBKF-style decision-making in the design of multiple-access schemes for navigation and communication networks, such as multi-vehicle LBL, inter-vehicle one way travel time navigation, and acoustic communication networks. For persistent missions, USBL-augmented navigation can be combined with the option of surfacing for GPS updates—the optimal balance of surfacing versus USBL updates could be formulated in the bandit framework. Additionally, MAB methods could be used to aid stochastic mapping problems, such as hydrothermal vent prospecting or plume tracking [60], where the exploration versus exploitation tradeoff considers whether to look for new potential environmental triggers, or follow up in directions that seem promising based on current information. Finally, since the exploration versus exploitation tradeoff is in fact an integral component of a large number of stochastic learning and decision-making processes, MAB approaches have the potential to improve many oceanographic missions via navigation methods as well as mission designs for effective data collection.

Appendix A

Restless Bandit Kalman Filter Index Solution

We give a detailed outline of the solution method given by Le Ny et al. in [71], as well as show some extended explanations of certain key concepts. Some of this material is included in Sec. 4.3, however it is repeated here for continuity and completeness.

A.1 Problem Setup

Here, we repeat the formal problem setup given in Sec. 3.3, but for the full multidimensional, multi-sensor problem.

The sensor management task is to provide state estimates for all targets that minimizes the weighted mean-square error on the system states plus additional measurement costs. Generally, the targets to be tracked are N independent Gaussian linear time-invariant (LTI) systems whose dynamics evolve according to

$$\dot{x}_i = A_i x_i + B_i u_i + w_i, \quad x_i(0) = x_{i,0}, \quad i = 1, \ldots, N \tag{A.1}$$

where A_i describes the dynamics of vehicle i, B_i is the control input matrix, and the driving process noise w_i is a stationary white Gaussian noise process with zero mean and a known continuous-time power spectral density matrix W_i, i.e. $\mathrm{Cov}(w_i(t)w_i(t)') =$

$W_i\delta(t-t')$, $\forall t, t'$. $M < N$ sensors are available to track the targets (note that in the VGR USBL case considered in this thesis, $M = 1$). If sensor j observes target i, a noisy measurement is obtained according to

$$y_{ij} = C_{ij}x_i + v_{ij} \tag{A.2}$$

where C_{ij} is the system measurement matrix for target i and sensor j and v_{ij} is a stationary white Gaussian noise process with power spectral density matrix V_{ij}, assumed to be positive-definite. We note that while Le Ny considers the continuous time case, the implementation of sensor scheduling in a real system is inherently a discrete-time process and a finite sample period must be chosen. The continuous-time description of the problem allows for powerful analysis methods, and real-world system dynamics of course evolve in continuous time, so this method allows true continuous-time dynamics to be used in the solution. For the specific analytic solution for LTI scalar systems, any discretization of the system will in fact give the exact states of the continuous-time equivalent system at the sample times.

The goal is a measurement policy, which is denoted by π. Define

$$\pi_{ij} = \begin{cases} 1 & \text{if plant } i \text{ is observed at time } t \text{ by sensor } j \\ 0 & \text{otherwise} \end{cases} \tag{A.3}$$

Each sensor can observe at most one system at each instant:

$$\sum_{i=1}^{N} \pi_{ij}(t) \leq 1, \quad \forall t, \quad j = 1, \ldots, M \tag{A.4}$$

Each system can be observed by at most one sensor at each instant:

$$\sum_{i=1}^{M} \pi_{ij}(t) \leq 1, \quad \forall t, \quad j = 1, \ldots, N \tag{A.5}$$

The problem considered is an infinite-horizon average cost problem to design an observation policy $\pi(t) = \{\pi_{ij}(t)\}$ satisfying the constraints A.4 and A.5, and a state

estimator $\hat{x}_\pi$ of the state of all targets x that *depends only on the past and current observations produced by the observation policy*, such that the average weighted error covariance over all targets, plus measurement costs are minimized. The cost function γ is thus

$$\gamma = \min_{\pi, \hat{x}_\pi} \overline{\lim_{\tau_f \to \infty}} \frac{1}{\tau_f} E\left[\int_0^{\tau_f} \sum_{i=1}^{N} \left((x_i - \hat{x}_{\pi,i})' T_i (x_i - \hat{x}_{\pi,i}) + \sum_{j=1}^{m} \kappa_{ij} \pi_{ij}(t) \right) dt \right] \quad \text{(A.6)}$$

where $\kappa_{ij} \in \mathbb{R}$ is the measurement cost per unit time when target i is observed by vehicle j, the T_i's are positive semidefinite weighting variances (how important a low error covariance is for a given target compared to another), and $\overline{\lim}$ denotes the upper limit, or lim sup. The formal statement uses lim sup because the covariance is inherently periodic (or at least has intermittent jumps downward) due to the switching observations—so lim sup means the upper limit of those cycles (since there is no true steady-state). Since the limit is as $T_f \to \infty$, as T_f gets longer, T_f could fall at different points in the measurement cycle, so the time average will move up and down, requiring the use of the supremum.

An unbiased estimator for the state estimate, $\hat{x}_\pi$ in continuous time is given by the Kalman-Bucy filter, with state estimates $\hat{x}_{\pi,i}$ for all vehicles $i = 1, \ldots N$ updated in parallel following

$$\frac{d}{dt}\hat{x}_{\pi,i}(t) = A_i \hat{x}_{\pi,i}(t) + B_i(t) u_i(t) - P_{\pi,i}(t) \left(\sum_{j=1}^{M} \pi_{ij}(t) C_{ij}^T V_{ij}^{-1} \left(C_{ij} \hat{x}_{\pi,i}(t) - y_{ij}(t) \right) \right) \quad \text{(A.7)}$$

with $\hat{x}_{\pi,i}(0) = \bar{x}_{i,0}$ for $1 \leq i \leq N$. The error covariance matrix $P_{\pi,i}(t)$ for system i satisfies the matrix Ricatti differential equation

$$\frac{d}{dt}P_{\pi,i}(t) = A_i P_{\pi,i}(t) + P_{\pi,i}(t) A_i^T + W_i - P_{\pi,i}(t) \left(\sum_{j=1}^{M} \pi_{ij}(t) C_{ij}^T V_{ij}^{-1} C_{ij} \right) P_{\pi,i}(t) \quad \text{(A.8)}$$

where $P_{\pi,i}(0) = P_{i,0}$. The dependence on the policy is evident in that the terms having to do with a new observation are switched on and off by the policy indicator function $\pi_{ij}(t)$. Thus, we refer to this as the *conditional* Ricatti equation. Note that while the

covariance evolution is dependent on the policy, due to the use of the Kalman Filter, it does not depend on the actual observation values—only if a measurement is taken. This means that the Kalman Filter handles the stochastic aspects of the system, and the problem of finding the optimal policy becomes a *deterministic* optimal control problem, described by the cost function

$$\gamma = \min_{\pi} \overline{\lim_{\tau_f \to \infty}} \frac{1}{\tau_f} \left[\int_0^{\tau_f} \sum_{i=1}^{N} \left(\mathbf{Tr} \left(T_i P_{\pi,i}(t) \right) + \sum_{j=1}^{M} \kappa_{ij}(t) \pi_{ij}(t) \right) dt \right] \tag{A.9}$$

subject to the constraints 3.4 and 3.5, where $E((x_i - \hat{x}_i)' T_i (x_i - \hat{x}_i)) = \mathbf{Tr}(T_i P_i)$ and the dynamics of the error covariance are given by A.8.

A.1.1 Targets with Scalar Dynamics and Identical Sensors

While [71] aims for an open-loop (steady-state) solution to the multidimensional case using semidefinite programming, they also give a closed-form analytic solution to the problem for targets with scalar dynamics and identical sensors. The closed-form analytic policy for scalar systems can be implemented during transient regimes, and (suboptimally) in situations where the parameters are changing dynamically (not time-invariant). We follow [71] and lay out the problem in the context of Lagrangian duality before proceeding with the solution method. First, the two constraints A.4 and A.5 can be combined into the single constraint that the requires the total number of vehicles measured at each instant to be M

$$\sum_{i=1}^{N} \pi_i(t) = M, \ \ \forall t \tag{A.10}$$

This constraint results in a difficult combinatorial optimization problem, so in order to obtain a lower bound on achievable performance, the constraint can be relaxed to enforce it only on average

$$\overline{\lim_{\tau_f \to \infty}} \frac{1}{\tau_f} \int_0^{\tau_f} \sum_{i=1}^{N} \pi_i(t) dt = M \tag{A.11}$$

Using standard nonlinear programming techniques, the Lagrangian function is formed by adjoining the (relaxed) constraint to the cost function using a (scalar) Lagrange multiplier λ:

$$L(\pi, \lambda) = \overline{\lim_{\tau_f \to \infty}} \frac{1}{\tau_f} \int_0^{\tau_f} \sum_{i=1}^{N} \left[\mathbf{Tr}(T_i P_{\pi,i}(t)) + (\kappa_i + \lambda)\, \pi_i(t) \right] dt - \lambda M \tag{A.12}$$

This optimization problem (with the relaxed constraint) can be expressed as

$$\underline{\gamma} = \inf_{\pi} \sup_{\lambda} L(\pi, \lambda) = \sup_{\lambda} \inf_{\pi} L(\pi, \lambda) \tag{A.13}$$

This leads us to compute the dual function $\underline{\gamma}_d(\lambda) := \inf_\pi L(\pi, \lambda)$

$$\underline{\gamma}_d(\lambda) := \inf_{\pi} \overline{\lim_{\tau_f \to \infty}} \frac{1}{\tau_f} \int_0^{\tau_f} \sum_{i=1}^{N} \left[\mathbf{Tr}(T_i P_{\pi,i}(t)) + (\kappa_i + \lambda)\, \pi_i(t) \right] dt - \lambda M \tag{A.14}$$

The dynamics of the systems are decoupled, and the only coupling is through the adjoined constraint, λM. This special problem structure allows for decomposition of the problem into N similar independent subproblems. The contributions of the individual system dynamics to the dual function can be computed independently as

$$\underline{\gamma}^i(\lambda) := \inf_{\pi_i} \overline{\lim_{\tau_f \to \infty}} \frac{1}{\tau_f} \int_0^{\tau_f} \left[\mathbf{Tr}(T_i P_{\pi,i}(t)) + (\kappa_i + \lambda)\, \pi_i(t) \right] dt \tag{A.15}$$

and the dual function is $\underline{\gamma}_d(\lambda) = \sum_{i=1}^{N} \underline{\gamma}^i(\lambda) - \lambda M$. The dual function $\underline{\gamma}_d(\lambda)$ over λ is concave, and maximizing it gives the performance bound $\underline{\gamma} \leq \gamma$.

A.1.2 Connection to Restless Bandits

For vehicle tracking, the projects or systems to be scheduled are obviously the vehicles, and activation of a project corresponds to taking a measurement of that vehicle. Following the conditional Ricatti equation A.8, the error covariance of the vehicles being tracked evolves with two distinct dynamics: one when active (measurement taken), and one when passive (We note that the conditional Ricatti equation with

$\pi_{ij} = 0$ is technically no longer a Ricatti equation—it becomes a Lyapunov equation). This fits the description of Restless Bandit projects in Section 4.2. The key insight in considering the problem A.13 in the framework of Whittle is to consider the Lagrange multiplier λ as a measurement tax that penalizes measurements of the system. By indexability, the passive action (not measuring) should become more attractive as λ increases. The Whittle index λ defines an intrinsic value for measurement of a given system, which takes into account immediate and future gains; this value is obtained by determining the measurement tax (potentially negative) that makes the controller indifferent between measuring and not measuring the system. This computation is done independently for each vehicle, and then the controller simply selects vehicle with the highest index (or in the case of multiple sensors, the vehicles with the M highest indices) for the next measurement(s).

A.2 Solution Method

Due to the decomposition made possible by the Whittle Index, we can now consider the computation of the index in problem A.15 for a single vehicle, dropping the index i for simplicity. For a single vehicle with scalar dynamics, the error variance evolution is described by

$$\dot{P} = 2AP + W - \pi \frac{C^2}{V} P^2 \tag{A.16}$$

with the policy $\pi(t) \in \{0, 1\}$. First, we will examine the behavior of this equation, which will inform our solution method. Consider the case where $\pi = 1$, i.e. the vehicle is always measured. For the nontrivial cases where $T \neq 0$ and $C \neq 0$, (A.16) becomes an algebraic Ricatti equation (ARE) for P, which has two roots, x_1 and x_2

$$x_{1,2} = \frac{A \pm \sqrt{A^2 + C^2 W/V}}{C^2/V}$$

We assume that $W \neq 0$ (this can be enforced mathematically if necessary by adding a small amount to W; physically this is justified by the fact that process noise is inherent in real-world systems), so x_1 is strictly negative and x_2 is strictly positive. Thus

we can take x_2 as the steady-state covariance when the vehicle is always measured. Additionally, if we consider the passive (no measurement) case, we set $\pi = 0$ and (3.8) becomes the Lyapunov equation $2AP + W = 0$. For stable systems ($A < 0$) this equation has a strictly positive solution, $x_e = -\frac{W}{2A}$. This represents the steady-state covariance when no measurements are taken. Note that marginally stable or unstable systems ($A \geq 0$) have no steady state covariance. The active and passive steady state covariance values for a stable system are thus

$$\pi = 1\text{: } P_{ss}^{active} = x_2$$
$$\pi = 0\text{: } P_{ss}^{passive} = x_e$$

Define three different covariance regions which will be used in the solution

$$\text{Region 1: } 0 < P < x_2$$
$$\text{Region 2: } x_2 < P < x_e$$
$$\text{Region 3: } P > x_e$$

For a marginally stable system (The scalar kinematic vehicle drift model $A = 0$, corresponding to a random walk, potentially with control), note that there is no steady-state covariance in the passive mode—we consider $x_e \to \infty$ as $A \to 0_-$, so the covariance remains in region 1 or 2.

For continuous sequential optimization, we start with the Hamilton-Jacobi-Bellman equation (HJB) for dynamic programming. In this case, the HJB is

$$\gamma(\lambda) = \min\left\{TP + (2AP + W)h'(P;\lambda)\ ,\ TP + (\kappa + \lambda) + (2AP + W - \frac{C^2}{V}P^2)h'(P;\lambda)\right\} \tag{A.17}$$

The HJB takes the minimum of the passive and active costs, which are the first and second arguments in the min function respectively. Note that the active cost includes the virtual measurement tax and Lagrange multiplier λ. The relative value function $h(P;\lambda)$ represents the differential cost caused by the transient effect of starting in

state P, rather than an equilibrium state. The derivative of h with respect to P is $h'(P;\lambda)$, which appears in the HJB equation, and can be written informally as

$$h' = \frac{dh}{dP} = \frac{\text{Equilibrium Cost - Actual Cost}(P)}{\dot{P} = 2AP + W - \pi\frac{C^2}{V}P^2} \tag{A.18}$$

The solution method for the nontrivial cases ($T \neq 0$ and $C \neq 0$) first assumes an optimal form for the policy. Following the discussion of indexability, and the concept behind the single-armed bandit example given in 4.1, the form of the optimal policy is a *threshold* policy. For some threshold variance value P_{th}, the policy observes the system when $P \geq P_{th}$ and does not observe for $P < P_{th}$. The approach is to determine the value of the average cost $\gamma(\lambda)$ and the threshold $P_{th}(\lambda)$. In a sense, we solve for the index λ in the opposite way from the way we use it in the policy—we assume a fixed threshold variance and find the value of λ that satisfies the HJB equation. Since the system is indexable if and only if $P_{th}(\lambda)$ is an increasing function of λ, we can invert this relation to give the Whittle index $\lambda(P)$; note that this index is now a *function* of the actual variance P of the vehicle at that instant, which is given by the Kalman Filter. Based on the variance regions described above (in relation to the steady-state values, which are functions of the system model), we must consider three cases for the location of this hypothetical threshold variance $P_{th}(\lambda)$. We can solve for the index λ in each region separately, and combine these solutions to define λ as a piecewise linear function of P.

For the edge cases (regions 1 and 3), the solution method is natural. In these cases, the threshold is either in an active region (region 1), or passive region (region 2), since the threshold variance is below the active steady-state (region 1), or above the passive steady-state(region 2). Thus, after a potential transient period, in these regions the variance will converge in finite time to the neighborhood of the steady-state covariance of the given region. We leverage this fact by explicitly stating the

average cost $\underline{\gamma}(\lambda)$ in these two regions

$$\text{Case } P_{th} \leq x_2 \text{ , active steady-state, region 1:} \qquad \underline{\gamma}(\lambda) = Tx_2 + \kappa + \lambda \tag{A.19}$$

$$\text{Case } P_{th} \geq x_e \text{ , passive steady-state, region 3:} \qquad \underline{\gamma}(\lambda) = Tx_e \tag{A.20}$$

We can equate the average cost expressions above with the HJB equation with $P = P_{ss}$ to determine $h'(P)$. For region 1, this becomes (note that the ARE is presented in factored form)

$$Tx_2 + \kappa + \lambda = TP + \kappa + \lambda - \frac{C^2}{V}(P - x_2)(P - x_1)h'(P) \tag{A.21}$$

so

$$h'(P < x_2) = h_1'(P) = \frac{TV}{C^2(P - x_1)} \tag{A.22}$$

Similarly, for region 3

$$Tx_e = TP + 2A(P - x_e)h'(P) \tag{A.23}$$

so

$$h'(P > x_e) = h_3'(P) = \frac{T}{2|A|} \tag{A.24}$$

Some algebra can relate these expressions for h' to more intuitive expressions derived from the original definition of the relative value function A.18

$$h'(P < x_2) = h_1' = \frac{T(x_2 - P)}{2AP + W - \frac{C^2}{V}P^2} = \frac{TV}{C^2(P - x_1)} \tag{A.25}$$

$$h'(P > x_e) = h_3' \qquad = \frac{T(x_e - P)}{2AP + W} = \frac{T}{2|A|} \tag{A.26}$$

In regions 1 and 3 we can use continuity at the active and passive interface to set the two arguments of the HJB equation equal, allowing solution for $\lambda(P_{th})$. For region 1,

$$TP_{th} + (2AP_{th} + W)\,h'(P_{th}) = TP_{th} + (\kappa + \lambda) + \left(2AP_{th} + W - \frac{C^2}{V}P_{th}^2\right)h'(P_{th}) \tag{A.27}$$

and for region 3,

$$\kappa + \lambda = \frac{C^2}{V} P_{th}^2 h'(P_{th}) = -\frac{C^2 T}{2AV} P_{th}^2 \tag{A.28}$$

Plugging in the appropriate expression for $h'(P)$, these equations can be solved algebraically to give $\lambda(P)$ as desired. Graphical examples of the two sides of the HJB equation are give in Fig. A-1. The measurement tax λ is used to translate the 'active' cost curve up or down in order for the active and passive cost curves to intersect at the desired value of P. This operation is essentially what the Whittle index is doing: determining the amount of measurement tax necessary to make the controller indifferent between measuring and not measuring (the point where the active and passive costs are equal). Note that here, P_{th} is the hypothetical threshold covariance used in the solution method. When the threshold is below the active steady-state variance (left plot), the policy is to always observe (after a potential transition period if the variance started at a value smaller than P_{th}), and the infinite-horizon average cost γ is the same as for the policy that always observes—as shown by the constant blue line. When the threshold is above the passive steady-state variance (right plot), then the policy is to never observe, and the infinite-horizon average cost γ is the same as for the policy that never observes—as shown by the constant red line. In region 2, the hypothetical threshold covariance P_{th} is in between the steady-state covariance values x_2 and x_e. Thus, we cannot determine an explicit relation to provide the value of the average cost. The authors in [71] use the method of Whittle and enforce continuity of the derivative of the relative value function with respect to P, h', and its derivative h'' at the region 1 and region 2 boundary. Following the smooth-fit principle, Whittle proposes a form for the index which is a function of the active and passive costs and the active and passive dynamics. Plugging in these expressions into Whittle's form leads to solution for $\lambda(P)$ in region 2. Formal justification is obtained by verifying that the solution proposed indeed does verify the HJB.

Refer to Sec. 4.3.1 for the closed-form index solution for scalar systems, and [71] for the multidimensional decomposition and algorithm for determining open-loop periodic policies.

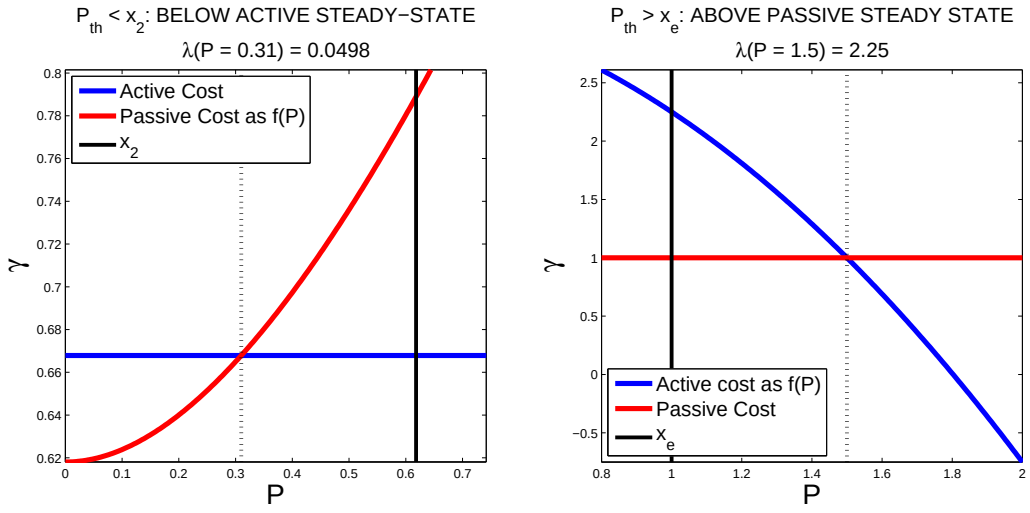


Figure A-1: Graphical illustration of solution for the Whittle index λ in regions 1 and 3 by equating the active and passive costs at desired P_{th}.

Bibliography

[1] DSPComm Acoustic Modems. http://www.dspcomm.com/products.html.

[2] EvoLogics R Series Acoustic Modems. http://www.evologics.de/en/products/acoustics/index.html.

[3] IXSEA Subsea Positioning Systems. http://www.ixsea.com/en/subsea_positioning.

[4] Kongsberg Acoustic Underwater Positioning and Navigation Systems. http://www.km.kongsberg.com.

[5] LinkQuest SoundLink Underwater Acoustic Modems. http://www.link-quest.com/html/intro1.htm.

[6] Sonardyne Positioning and Navigation. http://www.sonardyne.co.uk/Products/PositioningNavigation/index.html.

[7] Teledyne Benthos Acoustic Modem Product Comparison. http://www.benthos.com/acoustic-telesonar-modem-product-comparison.asp.

[8] Teledyne RDI Instruments Phased Array Velocity Sensor (PAVS). http://www.rdinstruments.com/PAVS.aspx.

[9] Hydro International Acoustic Modem Product Survey, 2007. http://www.hydro-international.com/productsurvey/id7-Acoustic_Modems.html.

[10] I.F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, 3(3):257–279, 2005.

[11] A. Alcocer, P. Oliveira, and A. Pascoal. Underwater acoustic positioning systems based on buoys with gps. In *Proceedings of the Eighth European Conference on Underwater Acoustics*, pages 12–15. Citeseer, 2006.

[12] B. Allen, R. Stokey, T. Austin, N. Forrester, R. Goldsborough, M. Purcell, and C. von Alt. Remus: a small, low cost auv; system description, field trials and performance results. In *OCEANS '97. MTS/IEEE Conference Proceedings*, volume 2, pages 994 –1000 vol.2, oct 1997.

[13] C. Ambler. Design of an Underwater Vertical Glider for Subsea Equipment Delivery. Master's thesis, Massachusetts Institute of Technology, 2010.

[14] P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *focs*, page 322. Published by the IEEE Computer Society, 1995.

[15] A. Bahr, J.J. Leonard, and M.F. Fallon. Cooperative localization for autonomous underwater vehicles. *The International Journal of Robotics Research*, 28(6):714, 2009.

[16] R.D. Ballard. *Archaeological oceanography*. Princeton Univ Pr, 2008.

[17] R.D. Ballard, L.E. Stager, D. Master, D. Yoerger, D. Mindell, L.L. Whitcomb, H. Singh, and D. Piechota. Iron age shipwrecks in deep water off Ashkelon, Israel. *American Journal of Archaelology*, 106(2):151–168, 2002.

[18] P. Batista, C. Silvestre, and P. Oliveira. A sensor-based controller for homing of underactuated AUVs. *Robotics, IEEE Transactions on*, 25(3):701–716, 2009.

[19] M.R. Benjamin, J.J. Leonard, H. Schmidt, and P.M. Newman. An overview of moos-ivp and a brief users guide to the ivp helm autonomy software. *Massachusetts Institute of Technology, MIT CSAIL, Tech. Rep. TR-2009-28-07*, 2009.

[20] D.P. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995.

[21] B. S. Bingham, J.M. Walls, and R.M. Eustice. Development of a flexible command and control software architecture for marine robotic applications. *Marine Technology Society Journal*, 45(3):25–36, May/June 2011.

[22] L.E. Bird, A. Sherman, and J. Ryan. Development of an active, large volume, discrete seawater sampler for autonomous underwater vehicles. In *OCEANS 2007*, pages 1 –5, 29 2007-oct. 4 2007.

[23] M. Blain, S. Lemieux, and R. Houde. Implementation of a rov navigation system using acoustic/doppler sensors and kalman filtering. In *OCEANS 2003. Proceedings*, volume 3, pages 1255 –1260 Vol.3, sept. 2003.

[24] J. Blandin, S. Bradley, R. Danioux, P. Gray, and G. Loaic. A network architecture concept for deep ocean lander systems. In *Electronic Engineering in Oceanography, 1997. Technology Transfer from Research to Industry., Seventh International Conference on*, pages 30–33. IET, 1997.

[25] M.F. Bowen, P.J. Bernard, D.E. Gleason, and L.L. Whitcomb. Elevators-autonomous transporters for deepsea benthic sample recovery. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 1, pages 343 –348 vol.1, 2000.

[26] S. Bradley, S. Addison, I.G. Priede, M. Collins, and P.M. Bagley. A deep-ocean fish tracking system using code-activated transponders and hydrophone array. In *Electronic Engineering in Oceanography, 1997. Technology Transfer from Research to Industry., Seventh International Conference on*, pages 34 –38, June 1997.

[27] E. Breuer, OC Peppe, and GB Shimmield. A novel approach for the study of north sea drill cuttings accumulations: The combined use of an rov and benthic lander for in situ measurements. *Underwater Technology: The International Journal of the Society for Underwater*, 25(2):77–82, 2002.

[28] L. Brignone, J. Alves, and J. Opderbecke. GREX sea trials: first experiences in multiple underwater vehicle coordination based on acoustic communication. In *OCEANS 2009-EUROPE*, pages 1–6. IEEE, 2009.

[29] J. Byron and R. Tyce. Designing a Vertical / Horizontal AUV for Deep Ocean Sampling. In *Oceans 2007 MTS/IEEE Conference Proceedings*, pages 1–10, 2007.

[30] D.A. Castanon. Approximate dynamic programming for sensor management. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 2, pages 1202 –1207 vol.2, dec 1997.

[31] A.D. Chave and C.S. Cox. Conductivity beneath the oceans. *Journal of geophysical research*, 87(B7):5327–5338, 1982.

[32] M. Chitre, S. Shahabudeen, L. Freitag, and M. Stojanovic. Recent advances in underwater acoustic communications & networking. In *OCEANS 2008*, volume 2008, pages 1–10. IEEE, 2008.

[33] R.D. Christ and R.L. Wernli. *The ROV manual: a user guide to observation-class remotely operated vehicles*. Elsevier, 2007.

[34] D.L. Codiga, J.A. Rice, and P.A. Baxley. Networked acoustic modems for real-time data delivery from distributed subsurface instruments in the coastal ocean: Initial system development and performance. *American Meteorological Society*, 21:331–346, 2004.

[35] S. Constable. Marine electromagnetic methods–a new tool for offshore exploration. *The Leading Edge*, 25(4):438, 2006.

[36] S. Constable and L.J. Srnka. An introduction to marine controlled-source electromagnetic methods for hydrocarbon exploration. *Geophysics*, 72(2):WA3, 2007.

[37] S.C. Constable, A. Orange, G.M. Hoversten, and H.F. Morrison. Marine magnetotellurics for petroleum exploration, Part I: A sea-floor equipment system. *Geophysics*, 63(3):816–825, 1998.

[38] S. Cowen, S. Briest, and J. Dombrowski. Underwater docking of autonomous undersea vehicles using optical terminal guidance. In *OCEANS'97. MTS/IEEE Conference Proceedings*, volume 2, pages 1143–1147. IEEE, 1997.

[39] J. Crowell. Small auv for hydrographic applications. In *OCEANS 2006*, pages 1 –6, sept. 2006.

[40] J.H. Cui, J. Kong, M. Gerla, and S. Zhou. The challenges of building scalable mobile underwater wireless sensor networks for aquatic applications. *IEEE NETWORK*, 20(3):12, 2006.

[41] J. Curcio, J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt, and M. Grund. Experiments in moving baseline navigation using autonomous surface craft. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 730–735. IEEE, 2005.

[42] V. Djapic and D. Nad. Using collaborative autonomous vehicles in mine countermeasures. In *OCEANS 2010 IEEE - Sydney*, pages 1 –7, may 2010.

[43] E. Dzielski, C. Tangirala, W.W. Moyer, and D.L. Bradley. Navoceano seahorse auv design, testing, and capabilities. In *OCEANS '02 MTS/IEEE*, volume 1, pages 151 – 155 vol.1, oct. 2002.

[44] C.C. Eriksen, T.J. Osse, R.D. Light, T. Wen, T.W. Lehman, P.L. Sabin, J.W. Ballard, and A.M. Chiodi. Seaglider: a long-range autonomous underwater vehicle for oceanographic research. *IEEE Journal of Oceanic Engineering*, 26(4):424–436, Oct 2001.

[45] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based slam. *Robotics, IEEE Transactions on*, 22(6):1100 –1114, dec. 2006.

[46] R.M. Eustice, L.L. Whitcomb, H. Singh, and M. Grund. Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4257–4264. IEEE, 2007.

[47] Ryan M. Eustice, Hanumant Singh, and Louis L. Whitcomb. Synchronous-clock one-way-travel-time acoustic navigation for underwater vehicles. *Journal of Field Robotics, Special Issue on State of the Art in Maritime Autonomous Surface and Underwater Vehicles*, 28(1):121–136, January/February 2011.

[48] N. Farr, A. Chave, L. Freitag, J. Preisig, S. White, D. Yoerger, and P. Titterton. Optical modem technology for seafloor observatories. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 928–934. IEEE, 2005.

[49] P. Favali, L. Beranzoli, et al. Seafloor observatory science: a review. *Annals of Geophysics*, 49(2/3):515–567, 2006.

[50] G.R. Fones, W. Davison, O. Holby, B.B. Jorgensen, and B. Thamdrup. High-resolution metal gradients measured by in situ DGT/DET deployment in Black Sea sediments using an autonomous benthic lander. *Limnology and oceanography*, 46(4):982–988, 2001.

[51] T.I. Fossen. *Guidance and control of ocean vehicles*. John Wiley & Sons Inc, 1994.

[52] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1086–1092. IEEE, 2005.

[53] I.G. Friede, J. Mienert, R. Person, T.C.E. van Weering, O. Pfannkuche, N. O'Neill, A. Tselepides, L. Thomsen, P. Favali, F. Gasparoni, et al. ESONET–European Sea Floor Observatory Network. *Elsevier Oceanography Series*, 69:291–294, 2003.

[54] J.C. Gittins and D.M. Jones. A dynamic allocation index for the sequential design of experiments. *Progress in statistics*, 1(241):66, 1974.

[55] R. Headrick and L. Freitag. Growth of underwater communication technology in the u.s. navy. *Communications Magazine, IEEE*, 47(1):80 –82, january 2009.

[56] J. Heidemann, Wei Ye, J. Wills, A. Syed, and Yuan Li. Research challenges and applications for underwater sensor networking. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 1, pages 228 –235, April 2006.

[57] B.W. Hobson, R.S. McEwen, J. Erickson, T. Hoover, L. McBride, F. Shane, and J.G. Bellingham. The development and ocean testing of an auv docking station for a 21" auv. In *OCEANS 2007*, pages 1 –6, 29 2007-oct. 4 2007.

[58] FS Hover, J. Vaganay, M. Elkins, S. Willcox, V. Polidoro, J. Morash, R. Damus, and S. Desset. A vehicle system for autonomous relative survey of in-water ships. *Marine Technology Society Journal*, 41(2):44–55, 2007.

[59] J. Howland, N. Farr, and H. Singh. Field tests of a new camera/led strobe system. In *OCEANS 2006*, pages 1 –4, sept. 2006.

[60] Jakuba. *Stochastic mapping for chemical plume source localization with application to autonomous hydrothermal vent discovery.* PhD thesis, MIT/WHOI Joint Program, 2007.

[61] J. Kalwa. The GREX-Project: Coordination and control of cooperating heterogeneous unmanned systems in uncertain environments. In *OCEANS 2009-EUROPE*, pages 1–9. IEEE, 2009.

[62] M.N. Katehakis and A.F. Veinott Jr. The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research*, pages 262–268, 1987.

[63] D.B. Kilfoyle and A.B. Baggeroer. The state of the art in underwater acoustic telemetry. *Oceanic Engineering, IEEE Journal of*, 25(1):4 –27, jan 2000.

[64] J Kinsey, R Eustice, and 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.

[65] W. Kohnen. 2007 MTS Overview of Manned Underwater Vehicle Activity. *Marine Technology Society Journal*, 42(1):26–37, 2008.

[66] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proc. of Uncertainty in Artificial Intelligence (UAI)*. Citeseer, 2005.

[67] A. Krause and C. Guestrin. Near-optimal observation selection using submodular functions. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1650. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[68] V. Krishnamurthy and R.J. Evans. Hidden markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking. *Signal Processing, IEEE Transactions on*, 49(12):2893–2908, 2001.

[69] N.H. Kussat, C.D. Chadwell, and R. Zimmerman. Absolute positioning of an autonomous underwater vehicle using GPS and acoustic measurements. *IEEE Journal of Oceanic Engineering*, 30(1):153–164, 2005.

[70] P. Lacovara. High-bandwidth underwater communications. *Marine Technology Society Journal*, 42(1):93–102, 2008.

[71] J. Le Ny, E. Feron, and M. Dahleh. Scheduling Continuous-Time Kalman Filters. *IEEE Transactions on Automatic Control*, 2011.

[72] Pan-Mook Lee, Bong-Hwan Jeon, and Sea-Moon Kim. Visual servoing for underwater docking of an autonomous underwater vehicle with one camera. In *OCEANS 2003. Proceedings*, volume 2, pages 677 – 682 Vol.2, sept. 2003.

[73] N.E. Leonard, D.A. Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni, and R.E. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48 –74, jan. 2007.

[74] S. Licht, V. Polidoro, M. Flores, F.S. Hover, and M.S. Triantafyllou. Design and projected performance of a flapping foil auv. *Oceanic Engineering, IEEE Journal of*, 29(3):786 – 794, july 2004.

[75] G. Marani, S.K. Choi, and J. Yuh. Underwater autonomous manipulation for intervention missions auvs. *Ocean Engineering*, 36(1):15–23, 2009.

[76] C. McGann, F. Py, K. Rajan, H. Thomas, R. Henthorn, and R. McEwen. T-rex: A model-based architecture for auv control. In *3rd Workshop on Planning and Plan Execution for Real-World Systems*, volume 2007. Citeseer, 2007.

[77] P.H. Milne. *Underwater acoustic positioning systems*. Gulf Publishing Co., Houston, TX, 1983.

[78] J. Niño-Mora. Stochastic scheduling. *Encyclopedia of Optimization*, 5:367–372, 2001.

[79] J. Partan, J. Kurose, and B.N. Levine. A survey of practical issues in underwater networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):23–33, 2007.

[80] N.G. Pavlidis, N.M. Adams, D. Nicholson, and D.J. Hand. Prospects for bandit solutions in sensor management. *The Computer Journal*, 53(9):1370, 2010.

[81] J.-P. Peyronnet, R. Person, and F. Rybicki. Posidonia 6000: a new long range highly accurate ultra short base line positioning system. In *OCEANS '98 Conference Proceedings*, volume 3, pages 1721 –1727 vol.3, sep-1 oct 1998.

[82] O. Pfannkuche and P. Linke. GEOMAR landers as long-term deep-sea observatories. *Sea Technology*, 44(9):50–55, 2003.

[83] W.B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. Wiley-Blackwell, 2007.

[84] I.G. Priede, S. Addison, S. Bradley, P.M. Bagley, P. Gray, C. Yau, J.-F. Rolin, J. Blandin, J. Legrand, A. Cremer, U. Witte, O. Pfannkuche, A. Tengberg, S. Hulth, W. Helder, and T. Van Weering. Autonomous deep-ocean lander vehicles; modular approaches to design and operation. In *OCEANS 1998 MTS/IEEE Conference Proceedings*, volume 3, pages 1238–1244, 1998.

[85] P. Rigby, O. Pizarro, and S.B. Williams. Towards Geo-Referenced AUV Navigation Through Fusion of USBL and DVL Measurements. In *OCEANS 2006 MTS/IEEE Conference Proceedings*, pages 1 –6, Sep. 2006.

[86] P. Rives and J.-J. Borrelly. Underwater pipe inspection task using visual servoing techniques. In *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 1, pages 63 –68 vol.1, sep 1997.

[87] C. Roman, T. Gregory, E. Martin, A. Sanguinetti, and J. Drummond. Preliminary model tests for the design of a gliding deep water elevator. In *OCEANS 2007*, pages 1 –5, 292007-oct.4 2007.

[88] P.A. Rona. Deep-diving manned research submersibles. *Marine Technology Society Journal*, 33(4):13–25, 1999.

[89] T. Schneider and H. Schmidt. Unified command and control for heterogeneous marine sensing networks. *Journal of Field Robotics*, 27(6):876–889, 2010.

[90] J. Sherman, RE Davis, WB Owens, and J. Valdes. The autonomous underwater glider 'Spray'. *IEEE Journal of Oceanic Engineering*, 26(4):437–446, Oct 2001.

[91] H. Singh, J. Catipovic, R. Eastwood, L. Freitag, H. Henriksen, F. Hover, D. Yoerger, J.G. Bellingham, and B.A. Moran. An integrated approach to multiple AUV communications, navigation and docking. In *OCEANS 1996 MTS/IEEE Conference Proceedings*, volume 1, pages 59–64 vol.1, Sep 1996.

[92] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *Automatic Control, IEEE Transactions on*, 49(9):1453–1464, 2004.

[93] I.M. Sonin. A generalized Gittins index for a Markov chain and its recursive calculation. *Statistics & Probability Letters*, 78(12):1526–1533, 2008.

[94] M.J. Stanway. Delayed-state sigma point Kalman filters for underwater navigation. In *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, pages 1 –9, sept. 2010.

[95] M.J. Stanway. Water profile navigation with an acoustic doppler current profiler. In *OCEANS 2010 IEEE - Sydney*, pages 1 –5, may 2010.

[96] M. Stojanovic. Recent advances in high-speed underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, 21(2):125–136, 1996.

[97] M. Stojanovic and J. Preisig. Underwater acoustic communication channels: Propagation models and statistical characterization. *Communications Magazine, IEEE*, 47(1):84 –89, january 2009.

[98] R. Stokey, T. Austin, B. Allen, N. Forrester, E. Gifford, R. Goldsborough, G. Packard, M. Purcell, and C. von Alt. Very shallow water mine countermeasures using the remus auv: a practical approach yielding accurate results. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 1, pages 149–156. IEEE, 2001.

[99] R. Stokey, M. Purcell, N. Forrester, T. Austin, R. Goldsborough, B. Allen, and C. von Alt. A docking system for REMUS, an autonomous underwater vehicle. In *OCEANS 1997 MTS/IEEE Conference Proceedings*, volume 2, pages 1132–1136, Oct 1997.

[100] P. Varaiya, J. Walrand, and C. Buyukkoc. Extension of the multi-armed bandit problem. In *Decision and Control, 1983. The 22nd IEEE Conference on*, volume 22, pages 1179–1180. IEEE, 1983.

[101] K. Vickery. Acoustic positioning systems: A practical overview of current systems. In *Proceedings Of The 1998 Workshop on Autonomous Underwater Vehicles, 1998.*, pages 5–17, Aug 1998.

[102] R. Washburn. Application of multi-armed bandits to sensor management. *Foundations and Applications of Sensor Management*, pages 153–175, 2008.

[103] D.C. Webb, P.J. Simonetti, and C.P. Jones. SLOCUM: an underwater glider propelled by environmental energy. *IEEE Journal of Oceanic Engineering*, 26(4):447–452, Oct 2001.

[104] S.E. Webster, R.M. Eustice, C. Murphy, H. Singh, and L.L. Whitcomb. Toward a platform-independent acoustic communications and navigation system for underwater vehicles. In *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pages 1–7. IEEE, 2010.

[105] R. Wernli. Recent us navy underwater vehicle projects. *Proceedings of the 24th Marine Facilities Panel, December*, 2001.

[106] L. L. Whitcomb, D. R. Yoerger, and H. Singh. Combined Doppler/LBL based navigation of underwater vehicles. In *Proceedings of the 11th International Symposium on Unmanned Untethered Submersible Technology*, 1999.

[107] L.L. Whitcomb and D.R. Yoerger. A new distributed real-time control system for the jason underwater robot. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 1, pages 368–374. IEEE, 1993.

[108] P Whittle. Restless Bandits: Activity Allocation in a Changing World. *Journal of Applied Probability*, 25:287–298, 1988.

[109] AJ Williams, Ruud Groenewegen Christoph Waldmann, and Anders Tengberg. Lander technology. In *Proceedings Inmartech, Texel The Netherlands*, 2000.

[110] J.L. Williams. *Information theoretic sensor management*. PhD thesis, Massachusetts Institute of Technology, 2007.

[111] J.L. Williams, J.W. Fisher III, and A.S. Willsky. Sensor management for multiple target tracking with heterogeneous sensor models. In *Proc. of SPIE Vol*, volume 6235, pages 62350F–1. Citeseer, 2006.

[112] D. Yoerger, M. Jakuba, A. Bradley, and B. Bingham. Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *Robotics Research*, pages 416–429, 2007.

[113] D.R. Yoerger, A.M. Bradley, M.V. Jakuba, C.R. German, T.M. Shank, and M.A. Tivey. Autonomous and remotely operated vehicle technology for hydrothermal vent discovery, exploration, and sampling. *Oceanography*, 20:152–161, March 2007.

[114] DR Yoerger, AM Bradley, SC Martin, and LL Whitcomb. The sentry autonomous underwater vehicle: field trial results and future capabilities. In *AGU Fall Meeting Abstracts*, volume 1, page 1674, 2006.