Autonomous Data Collection Techniques for Approximating Marine Vehicle Kinematics

by

Jacob Gerlach
B.S. Naval Architecture, United States Naval Academy (2008)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degrees of

Master of Science in Naval Architecture and Marine Engineering
and
Master of Science in Ocean Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

Author .... Signature redacted ....
Department of Mechanical Engineering
May 8, 2015

Certified by .... Signature redacted
Michael R. Benjamin
Research Scientist, Department of Mechanical Engineering
Thesis Supervisor

Certified by .... Signature redacted ....
John. J. Leonard
Samuel C. Collins Professor of Mechanical and Ocean Engineering
Thesis Supervisor

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

Autonomous Data Collection Techniques for Approximating Marine Vehicle Kinematics

by

Jacob Gerlach

Submitted to the Department of Mechanical Engineering
on May 8, 2015, in partial fulfillment of the
requirements for the degrees of
Master of Science in Naval Architecture and Marine Engineering
and
Master of Science in Ocean Engineering

Abstract

Understanding vehicle kinematics is essential in allowing autonomous guidance algorithms to accurately assess short range encounters. Low cost, reconfigurable autonomous vehicles motivate using in-field online techniques rather than tow tank testing or Computational Fluid Dynamics (CFD). While the parameters of many physical dynamic models can be obtained using System Identification (SI) techniques, these models require knowledge of the vehicle actuators, which may not be the case in a "backseat driver" architecture using payload autonomy. Even when an identified physical model is available, using it to simulate trajectories requires insight into the design of the relevant controller, which may be proprietary or otherwise unknown to the back seat. This thesis develops a data collection procedure to obtain empirical kinematic trajectories for unmanned surface vehicles (USVs). A linear black box model of the USV yaw system is also developed, using only data available in the backseat. A prediction table for the M200 USV is developed with both techniques.

Thesis Supervisor: Michael R. Benjamin
Title: Research Scientist, Department of Mechanical Engineering

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

Acknowledgments

First, thanks to my parents for fostering a love for education and an insatiable curiosity about the world. All of my success can probably be traced back to a mother that read to me.

I am indebted to the Navy for offering me the fantastic opportunity to go to MIT as my job. I look forward to contributing to the Navy with everything I've learned here.

Thanks also to Alon Yaari. His efforts to develop and improve the MOOS interface to the M200's, maintenance of the vehicles themselves, and insights into their inner workings were essential.

Thanks to John Leonard for his insight and valuable perspective on the greater world of robotics. I regret not getting his input earlier and more often.

This thesis would have been impossible without Kyle Woerner. I only became aware of MOOS-IvP through his promotion of course 2.680. Although highly challenging, it was extraordinarily rewarding and I can't overstate how much I learned. His accompanying doctoral work was essential to this work. He was always quick to address mutual concerns, listen to ideas, and provide valuable advice.

A heartfelt thanks to Mike Benjamin. It's hard for me to imagine how much better my education over the years would have been if every teacher invested as much time and personal interest in their students.

I can't offer enough thanks to my wife Monica and sons Noah and Elijah. My family was always patient with long hours, late nights, and coding on the couch. Without their support, I could never have finished this work.

Finally, eternal thanks to God for all of the rich blessings in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As Autonomous Marine Vehicle (AMV) technology proliferates, there is an ever increasing demand for improving their performance. Improvements to endurance, sensing, and intelligence are actively researched and continuously implemented.

One of the core subsystems of any autonomous vehicle is Navigation, Guidance, and Control (NGC). The navigation component can be as simple as obtaining data from GPS for a simple USV or as complex as fusing information from a compass, inertial navigation system, and data from fathometers and acoustic positioning systems to maintain an accurate estimate of position for the duration of a UUV mission.

Vehicle guidance consists of choosing the vehicle's course and speed. Often, the vehicle's path for an entire mission or portion of a mission is determined at once, a process called path planning in the literature. Guidance functions may also include reactive components, either determining the desired heading instantaneously or modifying a pre-planned path when new information is obtained.

With the vehicle's trajectory determined by the guidance component, the vehicle control system must determine how to command the vehicle's actuators in order to achieve the desired course and speed. Most AMV's are underactuated and nonholonomic and research into various control methodologies remains highly active.

A huge variety of NGC approaches are available to AMV designers and operators, and the optimal solution varies based on the desired budget, capabilities, and flexibility. One gap in the existing options is the capability to provide accurate vehicle guidance for short range encounters on low cost vehicles using an architecture known as payload autonomy. The goal of this thesis is to develop data collection techniques needed to enable guidance algorithms operating in payload autonomy systems to improve short range decision making by considering approximations of vehicle kinematics.

1.2 Overview

Chapter 2 reviews existing guidance and control methodologies and addresses why a new approach is pursued for payload autonomy. Chapter 3 describes a system to provide short term predictions of vehicle kinematics using either explicitly measured trajectories or a black box system model. The primary product of this thesis is software developed for MOOS-IvP, a modular, behavior based autonomy system. MOOS-IvP is described in Chapter 4 as are the algorithms and data collection mission developed in this work to collect kinematic data. Chapter 5 describes the validation of the data collection mission in simulation and on Kingfisher M200 USV's. Finally, Chapter 6 discusses some limitations of the current implementation and potential approaches to improve on these limitations.

Chapter 2

Background

2.1 Guidance

2.1.1 Path Planning

Path planning has its roots in early land based robots. Given a starting position, goal position, and some knowledge about obstacles, the robot must select a path to travel. Choosing a path is an optimization problem. The most obvious criterion to optimize is distance: what is the path that gets to the goal, avoiding the obstacles, while traveling the least distance? [12]

The basic problem formulation translates well to marine autonomy. The guidance component of many AMV missions can be framed as choosing or following a path while avoiding obstacles. Obstacle avoidance for other moving vessels imposes additional constraints on path planners, especially when there are protocols the AMV should respect. Woerner [18] discusses a variety of reasons to pursue AMV guidance approaches that incorporate the collision avoidance regulations (COLREGS) used by human steered ships [9].

2.1.2 Path Smoothing

Depending on the path planning method used, the resulting optimum path may be piecewise linear. For underactuated vehicles, a piecewise linear path cannot be accurately followed. Various approaches exist in the literature to generate smooth paths. The simplest approach is to generate a smooth path that passes through the waypoints determined by a simple path planning algorithm. This approach is generally implemented by smoothing after a path has been selected rather than being part of the optimization process [8].

One simple smoothing method is the cubic spline. It is straightforward to fit a cubic spline through path planner waypoints. This guarantees that the curvature of the path is continuous, which is an improvement. Cubic splines suffer from the fact that the entire path is affected by each waypoint. This may mean that the smoothed path may be significantly different from the path originally determined by the planner, potentially sacrificing optimality or even being unsafe. It also means that the entire path must be recalculated if a local modification is performed (say, for obstacle avoidance). Finally, although the cubic spline path has a continuous second derivative, it is not necessarily dynamically feasible. [8]

One alternative to cubic splines is B splines. B-splines have an advantage over cubic splines because each control point only has local influence. Moving a control point, for example to conduct local path modifications, only modifies a portion of the path. Despite this advantage, B splines still do not explicitly account for the maneuvering characteristics of the vehicle. However, one guidance algorithm can dynamically change the order of a B-spline used to smooth the path between planned waypoints [13]. Higher order splines were found to be tracked more easily by the vehicle.

Rather than modifying a path to make it smooth, many path planning algorithms explicitly consider a dynamic constraint as part of the planning algorithm. For example, some path planning approaches include curved segments with limited radius of curvature, accounting for the vehicle's turning capability [15]. This simplifies the vehicle's turning kinematics to a combination of straight line motion and constant radius of curvature turns. While turning radius is an important constraint, it can only represent the kinematics of a vehicle that has achieved a steady state turn.

While smoothing techniques and curvature constraints can generally produce dynamically feasible paths, their application in the literature is limited to cases where standoff ranges to obstacles are sufficient that the transition from straight travel to steady state turning is not significant.

2.1.3 Short Range Avoidance

In general, good seamanship (in practice) and the Collision Regulations[1] [9] (by statute) dictate that avoidance maneuvers be taken early and should result in comfortable passing distances rather than more "efficient" maneuvers which might narrowly avoid collision. Nevertheless, there are cases where understanding the limits of the vehicle's maneuvering capabilities is necessary to take appropriate action. While the notion of "short" range depends in general on vehicle size, it depends more specifically on the vehicle's maneuvering characteristics. There is no precise definition for short range, but several cases are presented where range to obstacles is on the order of the vehicle's turning radius. These cases certainly qualify as short range.

Formation control for multiple cooperating AMVs is an active area of research. Depending on the application, it may be desirable to maintain very short ranges between members of the swarm.

In some complex interactions, it may be necessary to delay taking avoidance action because of a second obstacle. Figure 2-1 depicts one such possibility. In this and other cases, must know their maneuvering capabilities to determine the correct maneuver.

Guidance algorithms that seek to employ COLREGS protocols for vessel interactions must handle situations where course and speed are constrained as the "stand on" vessel[2]. If the other vessel, required to "give way"[3], does not take appropriate action, the stand on vessel may have the option (Rule 17.a.ii) or obligation (Rule 17.b) to take avoidance action at short range. As in Figure 2-1, the correct maneuver in any given situation may depend on the maneuverability of the vessel in question.

Finally, autonomous sensing and classification of obstacles still has significant

[1] See Rule 8.

[2] See Rule 17.

[3] See Rule 16.

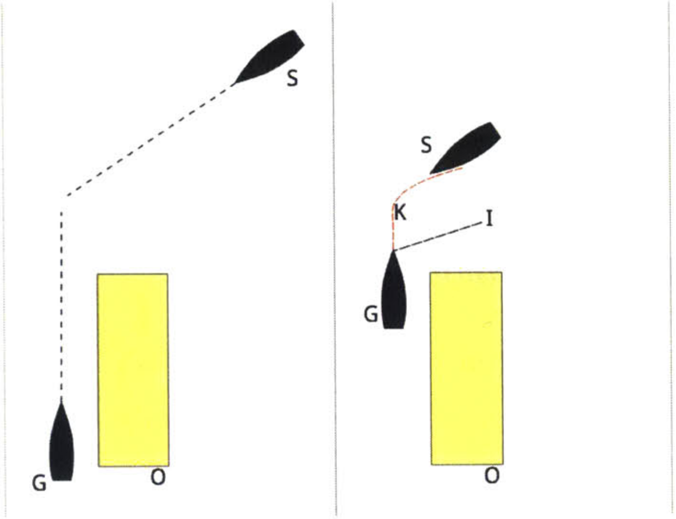


Figure 2-1: **Motivation for Short Range Avoidance:** Remaining to the left of obstacle O means that vessel G is unable to maneuver to avoid vessel S until close range. G's maneuvering capabilities may make it impossible to keep clear of S while turning (path K), in which case she should slow and allow S to pass ahead rather than maintaining speed and trying to maneuver. Assuming an instantaneous turn (path I) will incorrectly make a collision free turn appear feasible

challenges. Advances in radar, LIDAR, and stereo vision are enhancing sensing capability, and methods to fuse data from multiple sensors are improving. Nevertheless, delays in sensing obstacles may require an AMV to take avoidance action at short range.

While research into techniques such as model predictive control is ongoing and practical implementations are used on some aircraft, most algorithms employed in currently fielded AMVs do not consider the kinematics associated with the initiation of a turn. As underactuated vehicles, any AMV with an initial forward speed will have some "advance[4]" in the original direction of travel while turning to a new course. In one autonomy system, obstacle and contact avoidance behaviors consider a straight line from the vehicle's position to the obstacle [4]. Candidate courses are evaluated based on the CPA resulting from this straight line, effectively assuming that the vehicle can make instantaneous turns. A similar approach uses behavior voting for candidate trajectories - arcs laid out such that each cell in a grid is covered[11]. In both of these cases, the evaluation criteria do not consider the advance required to

[4] The term "advance" refers to translation in the original direction of motion during a turn. Advance and its counterpart transfer are discussed in more detail in Section 3.1.

achieve the new course.

These approaches and others currently in use benefit from the fact that obstacles and collision avoidance standoff ranges are typically much larger than the advance distance associated with a particular course change. Furthermore, the NGC system updates decisions and control action many times while approaching such an obstacle. A small error in estimation of the vehicle's trajectory can typically be compensated for with future guidance and control decisions.

Consider, however, an obstacle detected at close range, as shown in Figure 2-2. Without knowledge of the expected vehicle kinematic trajectory, the first course would be selected and a collision could occur. Buffer zones around obstacles help compensate for this, but this is limiting in a crowded obstacle field such as a minefield. Buffers would also be inadequate to enable accurate navigation in a narrow channel.

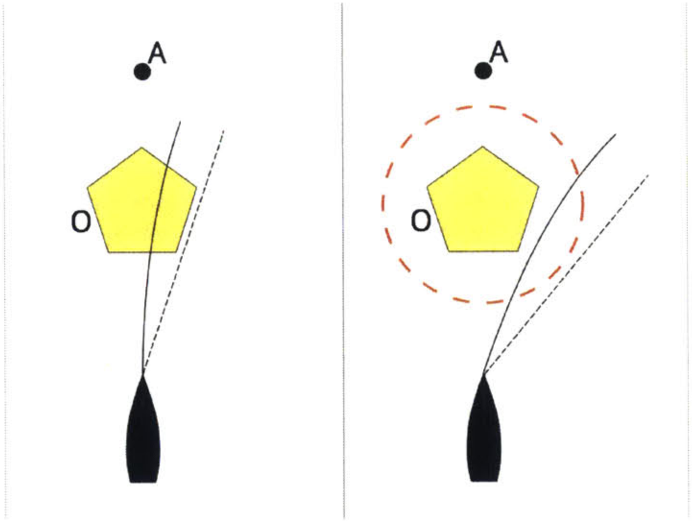


Figure 2-2: **Inadequacy of assuming Instantaneous Turns:** A vehicle driving to waypoint A must avoid obstacle O. At close range, collision may occur due to the difference between an instantaneous turn and actual vehicle trajectory. If an offset is added to the obstacle, a larger turn is executed and collision is avoided.

Alternatively, consider the scenario shown in Figure 2-3. Assuming instantaneous turn capability results in an inaccurate expectation of reversing course to pass to the opposite side of an obstacle. This type of difficulty was encountered in [11], which describes an issue with decision making fluctuating between alternate sides of an obstascle. Similar issues have occurred in obstacle avoidance algorithms in the MOOS-

IvP autonomy software used in this thesis. In both cases, a sort of decision making hysteresis was introduced to "commit" to passing on one side of an obstacle. An algorithm that considers the expected vehicle kinematics would inherently recognize the impossibility of immediately changing course to cross to the other side and would make the correct decision without artificial coping mechanisms.

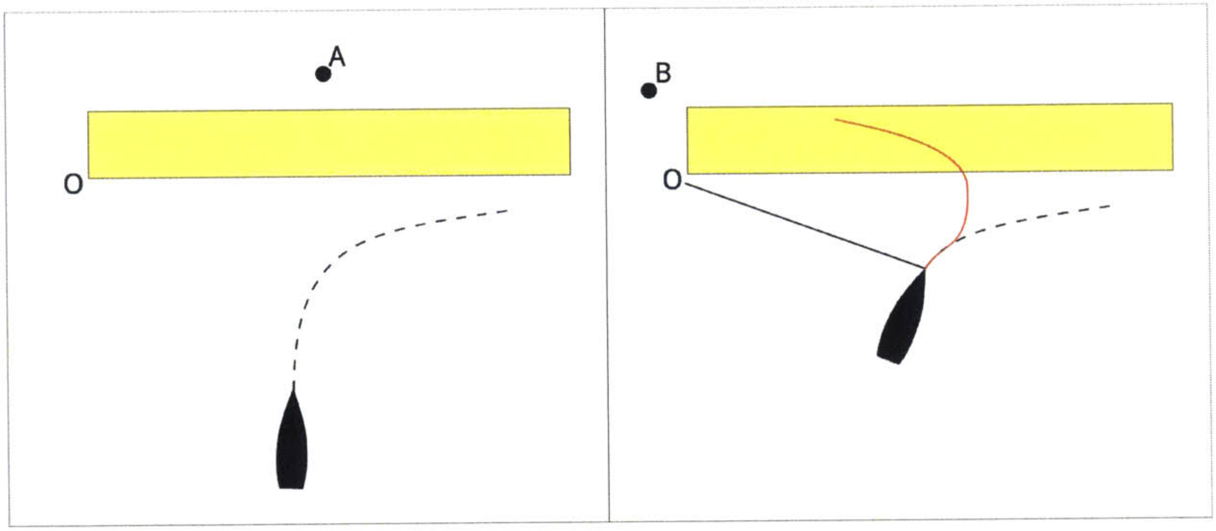


Figure 2-3: **Motivation for Short Range Avoidance:** Initially, the vehicle has decided to pass to the right of obstacle O to reach waypoint A. Later, some other factor has updated the waypoint to B and passing to the left of the obstacle is preferable. Assuming either instantaneous course change or instantaneously achieving a maximum turn rate makes this option appear possible. In reality, the vehicle cannot reverse course without colliding with the obstacle.

Given these types of challenges, it is clear that guidance algorithms cannot be assured of taking the right action in short range engagements without knowledge of the turning kinematics.

2.2 Dynamics and Control

Models of vehicle dynamics have been used extensively in the field of control theory. If the necessary parameters are known, the equations of motion can be used directly to solve for the vehicle's motion based on the applied forces. This leads directly to a prediction of the vehicle's kinematics for a given force input. This type of kinematic prediction would be suitable for enhancing guidance algorithms to better handle short range encounters.

2.2.1 Notation

Notation in this this thesis follows the conventions given in [10], summarized in Table 2.1. Additionally, $\Delta\psi$ is used to denote the discrete heading change associated with a specific intended turn.

Table 2.1: Summary of notation used for kinematics and dynamics in six degrees of freedom.

Direction	Translation	Force	Velocity	Position
x	Surge	X	u	x
y	Sway	Y	v	y
z	Heave	Z	w	z

Axis	Rotation	Moment	Angular Velocity	Angle
x	Roll	K	p	ϕ
y	Pitch	M	q	θ
z	Yaw	N	r	ψ

2.2.2 System Models

The 6 DOF motion of a marine vehicle on the surface (such as a USV) is usually simplified to plane motion for steering models, neglecting the influence of roll, pitch, and heave. With these simplifications, both linear and nonlinear approaches exist to treat the motion of the vehicle with varying degrees of accuracy. After simplifying to 3-DOF the equation of motion is given [10] by

$$\mathbf{M}\dot{u} + \mathbf{D}(u)u = \mathbf{f}(u) \tag{2.1}$$

$u = [u, v, r]^{\mathrm{T}}$ is the planar velocity, $\mathbf{M}$ is the inertia matrix (including added mass), and $\mathbf{D}$ is the hydrodynamic drag matrix. $\mathbf{f}(u)$ is the vector of forces on the vehicle, including actuators and environmental forces. This model might prove inadequate when substantial pitch and roll motions occur (in heavy seas, for example), but is sufficient for many situations where out of plane motions are not significant.

Simplifying assumptions such as neglecting external forces and off-diagonal entries in $\mathbf{M}$ and $\mathbf{D}$ gives a planar model for vehicle dynamics

$$\begin{aligned} m_{11}\dot{u} - m_{22}vr + d_1u &= f \\ m_{22}\dot{v} + m_{11}ur + d_2v &= 0 \\ m_{33}\dot{r} + m_d uv + d_3r &= T \end{aligned} \tag{2.2}$$

While some applications may require models with less assumptions and simplifications (for example, including external forces), it has been noted that the majority of USV heading controllers are based on this model [2].

Determining the added mass and hydrodynamic drag coefficients in $\mathbf{M}$ and $\mathbf{D}$ is a form of system identification (SI). Traditionally, SI for a large ship would be performed experimentally. Computational Fluid Dynamics is also capable of estimating the requisite coefficients. Both of these approaches can be expensive. A complete set of SI experiments requires extensive time in a tow tank [1]. Commercial CFD software, as an alternative, is itself costly and can require specialized computing hardware.

Research and other activities are increasingly relying on larger numbers of small low cost AMVs. In this environment, it may not always be realistic to expect this type of analysis to be done. Furthermore, mission specific payloads have the potential to significantly change the identified properties. This could be as simple as changing the mass and displacement of the vehicle or could be more complex if the payload includes some type of sensor in the water [14]. SI, however performed, would have to be repeated for each configuration of the vehicle.

2.2.3 On Line Identification

An alternative to tow tank experiments and CFD is "on-line" system identification. This allows determining approximate hydrodynamic properties in the course of operating the vehicle in the real world. The flexibility afforded by this technique is an attractive solution when low cost vehicles and mission reconfigurability make it impractical to utilize other SI techniques.

Regardless of whether SI is performed online or offline, any dynamic system model will require some knowledge of the vehicle's actuators: rudder angle, propeller speed, etc. Depending on the type of model used, SI may also require measurements such as sway velocity. This follows from the idea of parameter identifiability [3]. This will become significant in Section 2.3 where sufficient measurements required to ensure system identifiability may not always be available.

2.2.4 Control vs Trajectory Prediction

The typical use for dynamic models is in vehicle control - determining desired actuator motion to achieve a guidance objective. As discussed in Section 2.1.3, considering the turning kinematics in higher level control functions on AMVs is primarily necessary for short term path planning, typically in support of obstacle or collision avoidance. These objectives require predicting the vehicle's trajectory based on future control action. By predicting trajectories, guidance algorithms can be restricted to consider only dynamically feasible paths. Additionally, by more accurately understanding the expected kinematics associated with a given decision, algorithms can more effectively determine if that decision will actually achieve the intended objectives.

Given a dynamic model with the appropriate parameters identified, such a trajectory could be generated by simulating the actuator commands and resultant vehicle motion for a suitable guidance input. Note that this requires knowledge of both the physical properties of the vessel and also the design and parameters of the low level controller generating the actuator commands.

2.3 Payload Autonomy

Payload autonomy is an approach to autonomous vehicles that decouples high and low level control. A "front seat" vehicle computer handles low level control of the vehicle's actuators. The front seat controls actuators to achieve goals such as "desired heading" or "desired speed" that are requested by a separate autonomy "back seat". This has also been referred to as the "backseat driver design philosophy" [5].

One of the benefits to payload autonomy is that an autonomy mission can be agnostic to the hardware where it will be executed. This can provide logistical advantages for organizations that deploy multiple different AMV's. Furthermore, an autonomy mission can be designed and implemented entirely on a payload computer. Provided with a payload interface, that mission should then be usable on any specific AMV. The notion of hardware agnostic payload autonomy imposes several limitations that make it challenging to allow the back seat autonomy system to use a dynamic system model for kinematic predictions.

The first challenge is limitations on the front to back seat interface. Increasing requirements beyond navigation data and simple goals like heading, depth, and speed complicates the interface. This is clearly not insurmountable, but leads to the next problem. Obtaining hardware independence means that a dynamics model in the backseat should work regardless of whether the autonomy mission is running on a differential drive catamaran or a monohull with a fixed propeller and rudder. This means that both the additional interfaces and the dynamics model used depend on the vehicle type.

Finally, even if the front to back seat interface were expanded to include all of the actuators and the dynamics model in the backseat where flexible enough to accommodate all vehicle types, the front seat still has a controller that will influence the vehicle's kinematics. *Predicting kinematics from the back seat would require modeling not just the physical behavior of the vehicle, but also the behavior of the controller.*

Model predictive control does exactly this (albeit without the restriction of payload autonomy). Work in [17] and [16] takes this approach, conducting path planning that explicitly models future control actions and vehicle dynamics to restrict path planning to dynamically feasible paths. While this is possible when the design of the control system is known, many commercial vehicles use proprietary controllers. An approach that approximates kinematics without requiring explicit knowledge of the controller is still desirable.

This chapter has reviewed some existing techniques used to account for realistic vehicle kinematics in guidance algorithms, as well as approaches to rigorously account for the vehicle dynamics in control theory. The next chapter will describe a new approach that does not depend on the controller being used.

Chapter 3

Approximate Vehicle Kinematics

With the goal to provide platform independent short term prediction of turning kinematics, a new approach is necessary. Given the limitations imposed by payload autonomy (Section 2.3), an empirical approach is warranted to provide broad applicability even when specific actuator movements are unknown. This means approximating the vehicle kinematics without the aid of a physical dynamic model (Section 2.2.2). Two alternatives are presented. First, in Section 3.3, an approach utilizing direct measurements of trajectories is described. This is approach is termed Approximate Vehicle Kinematics (AVK). An alternative approach, utilizing a black box model of the ship yaw system is described in section 3.4.

3.1 Advance and Transfer

Approximate kinematics have long held wide acceptance by human mariners. Coastal piloting has many of the same challenges described in Section 2.1.3. In general, overlapping constraints require the ship's crew to execute very precisely timed maneuvers. A prime example is maneuvering a large ship to remain within a channel. The vehicle's kinematics are approximated by two linear distances: advance and transfer (Figure 3-1). These are defined as the distance traveled parallel and perpendicular to the vehicle's original course [7]. These distances are a function of vehicle speed u, initial rudder angle δ_0, and heading change $\Delta\psi$:

$$[A, T] = f(\delta_0, u, \Delta\psi) \tag{3.1}$$

Knowing these distances allows a ship's Navigator to conduct detailed advance planning for a turn, determining the exact point to initiate the turn ("Start of Turn" in Figure 3-1).

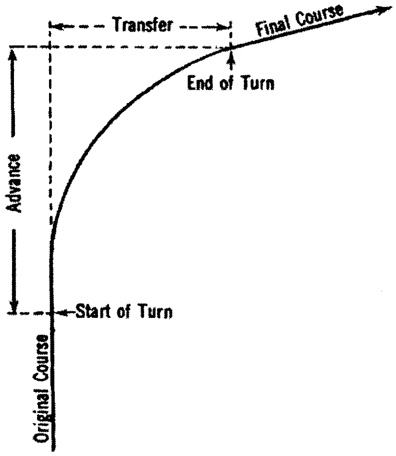


Figure 3-1: Advance and Transfer as considered by typical navigation practice on ships. The helmsman deflects the rudder at the point labeled "Start of Turn". At some point during the turn, the helmsman shifts the rudder, and makes fine adjustments as necessary to steady the ship on the new heading. If the desired "End of Turn" point is known, the Advance and Transfer (known in advance for a given ship based on the intended speed and rudder angle) can be used to calculate the "Start of Turn" point. Image borrowed from [7].

Although advance and transfer are referenced to the rudder angle used to initiate the turn, $\delta(t)$ is not constant. The helmsman manipulates the rudder as appropriate to achieve the new course precisely ("End of Turn" on Figure 3-1). The expected rudder manipulation is implicit in the kinematic approximation. Furthermore, while the end of turn point is calculated, the trajectory between start and end is just an approximation made by drawing a fair line between the points [7].

Just as traditional navigation practice abstracts the actions of the helmsman, the payload autonomy approach abstracts the actions of the front seat controller. Where the goal of this technique in traditional navigation is essentially advance path planning (where exactly to execute a pre-planned maneuver), approximate kinematics are intended here to aid in evaluating the suitability of various maneuvers to achieve reactive collision avoidance goals.

3.2 Kinematics Lookup Table

Just as advance and transfer for a large ship are functions of rudder angle, speed, and heading change (equation 3.1), the kinematic trajectory for a USV turn is a function of similar inputs. Where a ship's Navigator could direct differing amounts of rudder for one heading change, it is assumed here that the payload autonomy interface provides no provision for such a request. Therefore the actuator motions used to produce a certain heading change are assumed to be a single valued function of the guidance input and initial state:

$$[N(t), K(t)] = f(u, \Delta\psi, r) \tag{3.2}$$

where the yaw moment $N(t)$ may be due to a rudder, differential propulsion, or some other method and the thrust $K(t)$ may also change during a turn. Under this assumption, the vehicle's kinematics are then a function of initial state and turn parameters

$$\mathbf{x}(t) = f(\Delta\psi, \mathbf{x}_0) \tag{3.3}$$

where $\mathbf{x} = [x, y, \psi, v, r]$. For the present work, consideration is restricted to initial states $\mathbf{x}_0$ where $v_0 = r_0 = 0$. This is a strong assumption and is discussed further in Section 3.4 and Chapter 6.

In the absence of external influences such as current, the global position $[x, y]$ and heading ψ of the vehicle have no influence on the resulting trajectory. The turn trajectory can then be referenced to the body fixed reference frame $\mathcal{N}$ from the moment the turn is initiated. $\mathcal{N}$ has an origin $[X_N, Y_N] = [x_{O,0}, y_{O,0}]$ with the positive y-axis oriented at ψ_0 where $(\cdot)_0$ represents the value at the start of the turn. The resulting trajectory starts at the origin of $\mathcal{N}$, traveling initially in the positive y direction and is denoted $\mathbf{x}_N$.

As long as the vehicle is symmetric about the longitudinal axis, the kinematics for a left and right turn will be the same. Right turns are chosen to be the reference. For a given combination of $(u_0, \Delta\psi)$, the corresponding left turn $-\Delta\psi$ can be recovered by reflecting the kinematic trajectory $\mathbf{x}_N$ across the y axis of $\mathcal{N}$.

Under these assumptions, $\mathbf{k} = (u_0, \Delta\psi)$ with $0 \leq \Delta\psi \leq 180$ uniquely identifies a turn trajectory. As a practical matter, the space $\mathbf{k}$ can be discretized into appropriate intervals for data collection and vehicle planning. The discretization of $\mathbf{k}$ is referred to as a Cached Kinematics Table (CKT), where each cell in the table $\mathbf{k}_{i,j}$ contains the trajectory $\mathbf{x}(t)$ associated with the turn $(i = u_0, j = \Delta\psi)$. This structure fits very naturally with a data collection mission and is consistent with the existing structure of the guidance decision space. Furthermore, the notion of caching this information may offer practical advantages for computational performance when used in guidance algorithms.

The specific discretization of the AVK table may vary. In a data collection mission, for example, it might be appropriate to choose a 20 degree $\Delta\psi$ spacing to minimize the time associated with the mission. In guidance algorithms, on the other hand, it might be necessary to discretize to 1 degree to match the candidate headings being considered.

3.3 Measuring Kinematic Trajectories

3.3.1 Data Collection

In order to populate the table, each turn k_{ij} must be performed and the resulting trajectory recorded. This is achieved through a data collection mission, described in Section 4.5. Essential to this effort is controlling vehicle initial conditions and controlling the guidance input (i.e. keeping the desired heading constant).

One challenge in executing this mission is the presence of external influences such as wind and current. These have the potential to substantially influence vehicle turning characteristics. Accounting for these factors is complicated further on vehicles where the navigation system is not capable of determining both speed through the water and speed over ground. Further discussion of possible techniques to exclude the influence of external forces is given in Section 6.3.

3.3.2 Fitting, Approximation, and Interpolation

An AVK trajectory collected in this manner is not directly useful. First, the position data recorded may be noisy. While the vehicle's speed generally changes during a turn, it is assumed here that the kinematic trajectory can be adequately approximated by assuming a constant speed during the turn, equal to the average speed required to traverse the recorded trajectory in the observed time. This assumption means that only the geometric shape of the input data needs to be retained, not the discrete time history. As a result, the input data can be simplified by approximating it with a function $y = f(x)$ over an appropriate domain of x. Accompanying work to this thesis examines the suitability of various data fitting techniques. A fourth order polynomial was chosen, fit to the data using least squares regression.

To improve the computational efficiency of guidance algorithms using the AVK trajectory, the trajectory is further simplified by approximating the polynomial as a piecewise linear function.

Section 3.2 discussed possible discretizations of the CKT. If the experimentally collected data is sparse, trajectories in the table can be interpolated to provide data for cells in the fine discretization used for guidance.

3.4 Black Box Modeling

3.4.1 Black Box Yaw System

While AVK trajectories stored in a kinematics lookup table offer a significant improvement in the predictive capability of a back seat guidance algorithm, the utility of the approach is severely limited by the restrictions on initial vehicle state introduced in Section 3.2. A system model, on the other hand, includes the current state explicitly and has fewer restrictions on applicability.

The physical dynamics models discussed in Section 2.2.2 predict vehicle motion on the basis of physical properties of the vehicle and its actuators. In the case of yaw dynamics, a simple example of this approach is the 1st-order Nomoto steering model [10].

$$\dot{r} = -\frac{1}{T}r + \frac{K}{T}\delta \tag{3.4}$$

where δ is the rudder angle and $\dot{r}$ is the derivative of yaw rate (i.e. $\ddot{\psi}$). Some technique must be used to identify the parameters T (a time delay factor) and K (a proportionality constant for the effectiveness of the rudder). Once identified, this model (either in conjunction with a speed model or by assuming some speed profile) could be used to predict a vehicle's trajectory for some known rudder input $\delta(t)$ by simulating the system shown in Figure 3-2 (replacing the actual vehicle dynamics with the model).

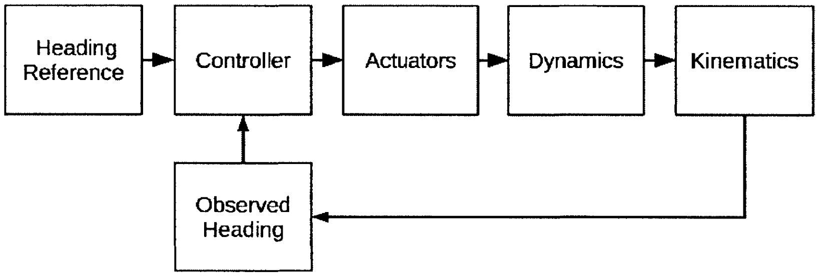


Figure 3-2: The USV yaw motion as a system. The guidance component determines the heading reference. The navigation system provides feedback to the controller. In response to changes in the heading reference and feedback signal, the controller produces actuator motions. The actuators exert forces and moments on the vehicle, resulting in a certain kinematic profile depending on the inertial and hydrodynamic properties of the vehicle (i.e. the dynamics).

To avoid the need to simulate the controller and know actuator positions, a different model is considered. If the boundaries of the model are expanded to include both the control layer and the actuator dynamics, a black box model of the entire system is obtained as shown in Figure 3-3. The inputs are desired and observed headings and the output is vehicle's heading.

A simple linear model of this system is

$$\psi_{t+1} = \mathbf{A}\bar{e} + \mathbf{B}\bar{\psi} \tag{3.5}$$

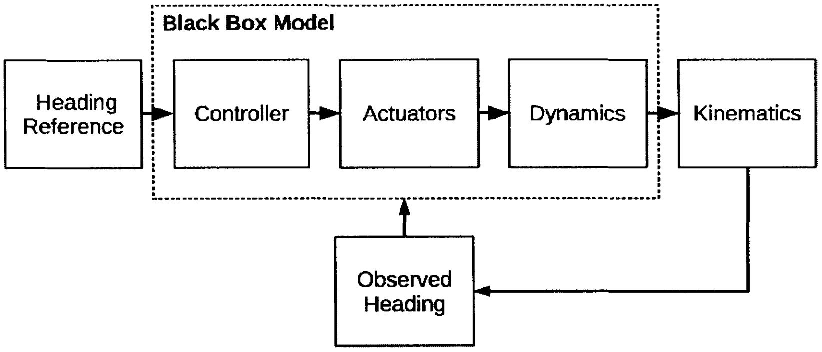


Figure 3-3: Black box model of USV yaw motion. The behavior of the controller and the actuators are modeled together with the vehicle dynamics. The resulting model can make a trajectory prediction without knowledge of the controller design or the actuator positions.

where $\bar{e}$ is a vector of the difference between the desired heading given by the heading reference and the measured heading $(\psi_d - \psi)$ at previous time steps. Similarly, $\bar{\psi}$ is a vector of previous vehicle headings. Given a sample data set from the vehicle, a simple least squares fit of the parameters $\mathbf{A}$ and $\mathbf{B}$ serves to perform SI for the black box model.

It is noted that the choice of input data can have significant impact on the quality of the produced model. For example, if $\bar{e}$ and $\bar{\psi}$ are monotonically increasing, then $\mathbf{A} = \mathbf{0}$ and $\mathbf{B} = [B_1, 0, ..., 0]$ where $B_1 > 1$ and the exact valuedepends on the input data. Such a model is obviously inadequate and can never predict heading decreasing (i.e. left turns). An alternating set of left and right turns, like those shown in Figures 3-5 and 3-6 results in values for each element in $\mathbf{A}$ and $\mathbf{B}$ and can predict both right and left turns. Finally, it is noted in [10] that persistent excitation is preferred for system identification. This is addressed further in Section 3.4.2.

With $\mathbf{A}$ and $\mathbf{B}$ identified, the model can be used to predict turning trajectories by simulating coordinates and heading changes.

- Begin with desired and actual heading for last n steps
- Construct error and heading vectors
- Use current x and y position, heading, and speed
- For each time step:

- Given x, y, heading, and speed
- Estimate next x and y position from heading and speed
- From previous heading data, predict next heading using model
- From next heading, calculate e at next step
- Update error and heading vectors

A modeling approach has several advantages compared to the approach described in Section 3.3. First, it nominally requires less data collection time. Rather than fully discretizing the $\mathbf{k}_{ij}$ domain with data collection points, an appropriate sequence of input data is collected at each speed.

A significant weakness that arises from the CKT structure and the direct measurement AVK approach is that the stored turn trajectories can only be used to directly predict trajectories that match the vehicle's behavior during data collection (i.e. the conditions associated with a certain CKT cell). In the present work, this means the vehicle is on a steady course. *Once the vehicle has started turning, the trajectories stored in the table cannot accurately predict the vehicle's path.* Despite potential options to allow the CKT/AVK approach to more robustly handle varied vehicle initial conditions (Section 6.2), incorporating these changes into the system increases complexity and may also require changes to the data collection mission.

A system model, by nature of considering the system state and recent state history, is able to generate a prediction anywhere along the trajectory of a turn. This avoids the need to classify the vehicle's current state (steady vs. turning) and have special cases to build a trajectory prediction accordingly.

Despite these advantages, obtaining an adequate model is not trivial. The model proposed in Equation 3.5 does not explicitly consider nonlinearities such as actuator saturation and nonlinear hydrodynamic forces. Additionally, although a well identified model may generate a reasonably accurate heading prediction, any error will be propagated at each time step over the course of a trajectory prediction.

3.4.2 Black Box Modeling of M200 Using In-Water Data

In water experiments were performed to evaluate the viability of this sort of black box model. This work was tested at the Battelle Autonomy Lab at the MIT Sailing Pavilion. While some early work was performed on Kingfisher M100's, the majority of the field testing was performed on Kingfisher M200's. The M200 is a small autonomous catamaran (see Figure 3-4). Propulsion is provided by two waterjets. Steering is controlled using differential drive. The frontseat contains interface layers to the thrusters and navigation equipment. Although the front seat can accept individual thruster commands from the backseat, it lacks heading and speed controllers. These functions are instead performed by pMarinePID, a MOOS application implementing a basic PID controller for heading and speed. Key configuration parameters affecting vehicle maneuvering characteristics are listed in Table 3.1.

Table 3.1: Key M200 configuration parameters affecting maneuverability. In tests for this work, the following parameters were used unless otherwise specified.

App	Parameter	Values
iM200	`MAX_THRUST`	100
iM200	`AppTick`	10
pMarinePID	`YAW_PID_KP`	0.5
pMarinePID	`YAW_PID_KD`	0
pMarinePID	`YAW_PID_KI`	0-0.05
pMarinePID	`YAW_PID_INTEGRAL_LIMIT`	10-20
pMarinePID	`SPEED_PID_KP`	1
pMarinePID	`SPEED_PID_KD`	0
pMarinePID	`SPEED_PID_KI`	0
pMarinePID	`MAXRUDDER`	40
pMarinePID	`AppTick`	4

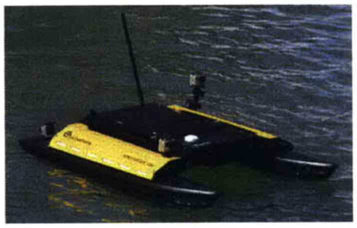


Figure 3-4: Kingfisher M200

Trajectory predictions for the M200 were generated by identifying a linear black box yaw model like that described in 3.5. Two sets of input data were used. The first, shown in Figure 3-5, consisted of several discrete turns. The second, shown in Figure 3-6, was a persistent excitement of the vehicle's heading. In both data sets, the desired vehicle speed was constant at 1.5 m/s.

No filtering was performed on the data. For the discrete turns in particular, nontrivial low frequency noise is apparent in the heading data. This is attributed to environmental effects. Filtering the input used for parameter identification might improve the performance of the model; this was not investigated.

The asynchronous nature of MOOS publications (see Chapter 4) means that data logged from a vehicle must be resampled to be suitable for use in linear least squares identification of the model coefficients. The minimum relevant application frequency was 4 Hz (pMarinePID). Other relevant applications, such as iM200 (providing the navigation data) and pHelmIvP (publishing desired heading) were operated at 10 Hz. To remain within the limits of the actual publications, data was resampled to 4 Hz. To resample, each relevant variable was linearly interpolated between logged values to obtain a value for each synchronous time step.

Ten seconds of history was used, corresponding to 40 samples in $\bar{e}$ and $\bar{\psi}$. Further discussion of model form is given in Section 3.4.3.

Trajectories simulated by modeling heading changes (as described in Section 3.4.1) are given in Figures 3-7 and 3-8. Both predictions are calculated from the point where the desired heading has changed but the control system has not yet responded (i.e. the first sample where DESIRED_HEADING changes). In both cases the model is stable and eventually converges to the new desired heading. While it under predicts the heading change at each time step, resulting in a wider turn than actually observed, the predicted response is generally similar to the actual trajectory.

The speed at the beginning of the turn was assumed to be constant and was used to propagate the predicted position. If a suitable speed model were also developed and identified, a more accurate prediction could be developed. Both models were also tested using the actual speeds from the resampled logs instead of an assumed constant speed. While this provided minor improvement, the M200 speeds in this data were

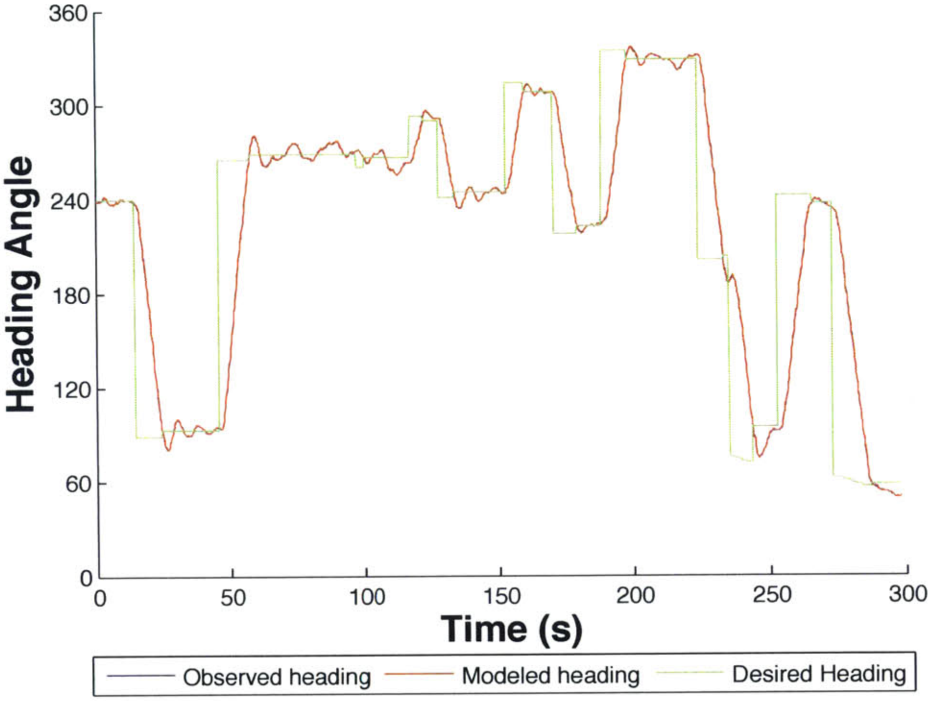


Figure 3-5: **Using M200 In-Water Data to Identify Model Parameters**: Input data for the black box system identification process. Here discrete turns are used, like those used to record AVK trajectories

relatively constant during the turns and the trajectories using actual speed data were not significantly different than those shown here using a constant speed assumption. For other cases where speed changes significantly during a turn, an accurate speed model would prove essential in developing accurate predictions.

While the model was able to predict some of the turns in its input data with some accuracy, it still needs refinement to adequately predict turns over the full range of heading changes. A comparison of results from this black box model and the AVK measured trajectory is given in Section 5.2.

3.4.3 Model Improvements

The black box approach described meets the goal of providing a predictive tool for cases where models of the vehicle's dynamics or low level control cannot easily be obtained. However, further work is necessary to evaluate whether a different mod-

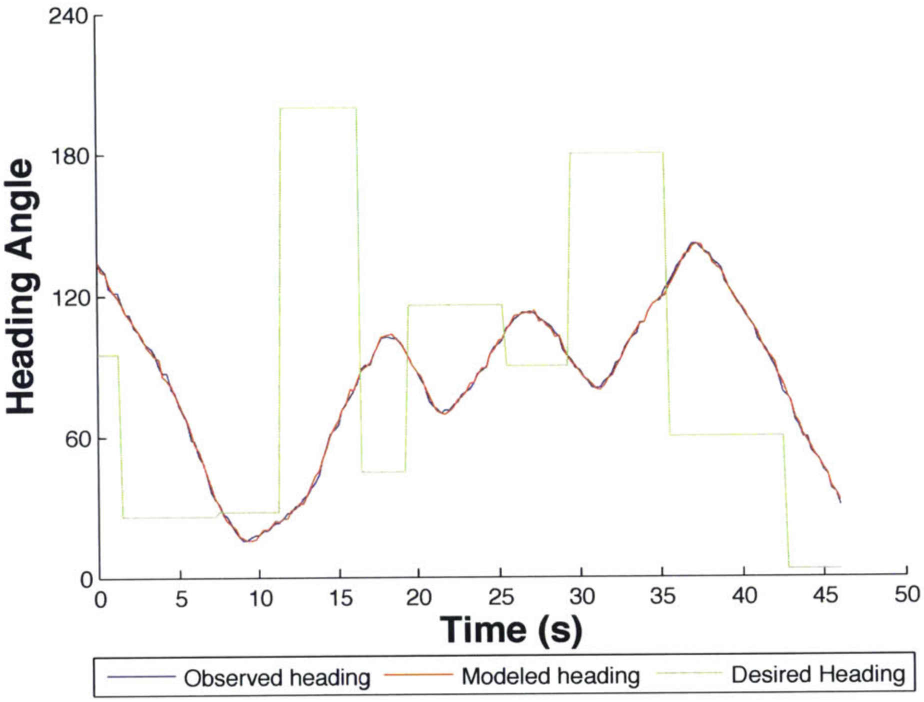


Figure 3-6: **Using M200 In-Water Data to Identify Model Parameters**: Input data for the black box system identification process. In this case, vehicle heading was persistently excited by intentionally varying `DESIRED_HEADING` to prevent the vehicle from steadying on course.

eling approach could more accurately predict trajectories. In particular, techniques other that least squares should be explored for identifying the values of the model parameters. A least squares fit minimizes the prediction error for heading *at the next time increment*. The actual desire, however, is to minimize error *in the predicted trajectory*. Parameter estimation techniques that allow an arbitrary cost function would allow the entire trajectory to be considered in the estimation process.

One parameter which could be systematically varied in a more robust identification process is the number of time steps included in the model. The best choice for this parameter should also be informed by the ultimate usage of the model. Predicting the latter part of a large turn trajectory may require 10-15 seconds of history in the model. On the other hand, this may have undesirable effects on predicting the beginning of a turn if the prior 15 seconds have significant changes. An example of the effect of model history is given in Figure 3-9. A related analysis is to determine how

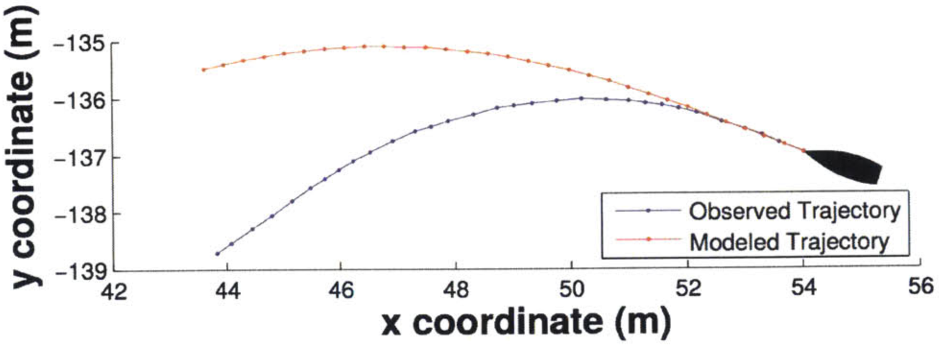


Figure 3-7: **Poor input data produces poor estimates:** Trajectory predictions using black box model. Model coefficients were identified from the discrete turn data set shown in Figure 3-5. The initial speed is assumed to be constant throughout the turn. In this case, the model has too much delay and predicts too little heading change at each time step. The result is a trajectory prediction with a much larger advance and transfer than the actual vehicle trajectory.

much of the trajectory must be accurately simulated to achieve the desired guidance results. It may be the case that modeling only the first (say) half of the turn achieves the necessary guidance algorithm performance. The results of this analysis could be included in the cost function for an alternative parameter estimation technique, giving higher weight, for example, to accurately predicting the beginning of the trajectory. Similarly, smaller turns (up to about 90 degrees) are more relevant to effective obstacle avoidance and could be given higher weight.

It would also be beneficial to study the model's sensitivity to the size of the time step. The time step used in the input data is implicit in the coefficients identified for a model like Equation 3.5. Utilizing a different time step requires changing the model structure and repeating the identification process.

3.5 Summary

This chapter has presented two potential approaches to allow a payload autonomous guidance system to estimate the kinematics of a turn. A simple black box model for USV yaw dynamics was developed and tested on an M200 USV. The AVK approach for directly measuring turn kinematics serves as the basis for the data collection procedure outlined in the next chapter.

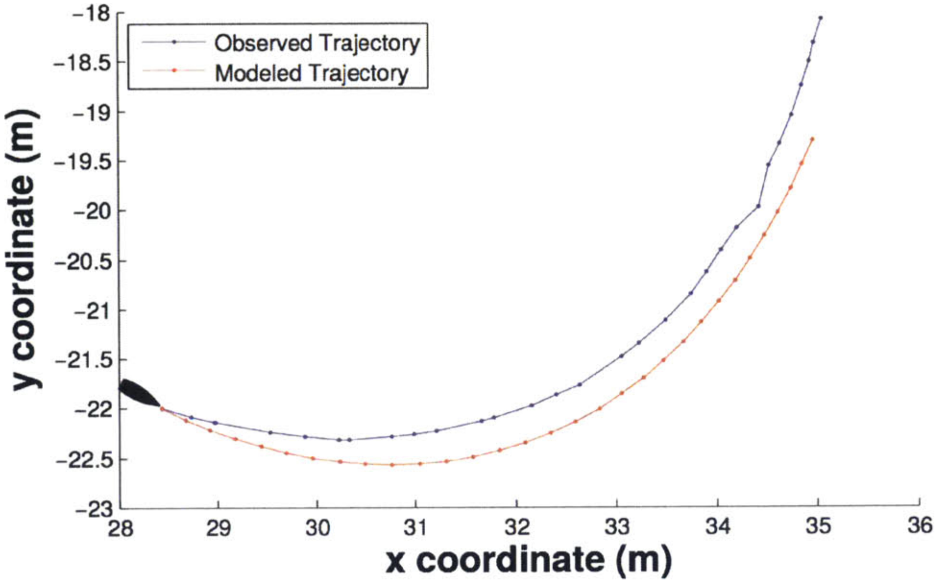


Figure 3-8: **Persistent excitation improves predictive capability:** Trajectory prediction using black box model. Model coefficients were identified from the persistently excited data set shown in Figure 3-6. The initial speed is assumed to be constant throughout the turn. Compared to Figure 3-7, the prediction has much better agreement with the observed trajectory, but has a lesser the same problem under-predicting heading change, resulting in a modest over-estimate of advance.

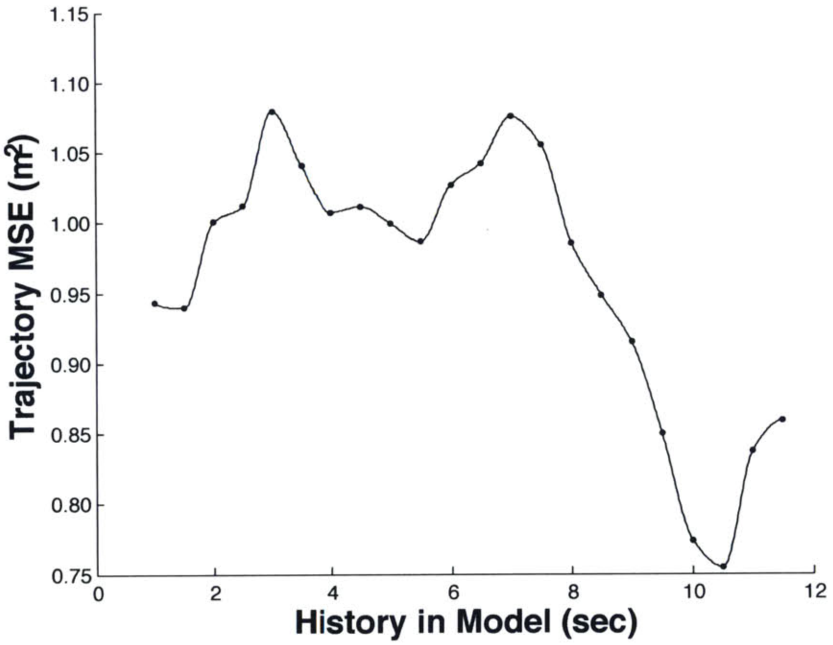


Figure 3-9: **Evaluation of model time scale in prediction accuracy:** The accuracy of the trajectory prediction generated by the model depends directly on the amount of history included. The actual (blue) turn shown in Figure 3-8 was predicted by models with varying amounts of history. The mean squared error between each point on the predicted trajectory and the observed trajectory is shown for each prediction. It should be noted that the minimum MSE is for only one turn and does not necessarily imply an optimum choice across all heading changes.

Chapter 4

MOOS Applications and Behaviors

Data collection for the AVK approach described in Chapter 3 has been implemented in MOOS-IvP here. A brief general description of MOOS-IvP is given. More detail is available in [5].

4.1 MOOS-IvP

The Mission Oriented Operating Suite (MOOS) serves as middleware, allowing various applications to communicate with each other. For a typical autonomous vehicle mission, applications for communication, sensing, data logging, and guidance would communicate and share data through the MOOS database (MOOSDB). In a payload autonomy architecture, an interface application to communicate with the front seat computer would also be present.

MOOS applications share data through a publish and subscribe architecture. A typical exchange of data relevant to this work is as follows: GPS data is processed in the front seat navigation system and shared to a MOOS interface application such as iM200. This application does any additional required processing such as conversion to local grid coordinates and posts a message such as `NAV_X = 10.5` to the MOOSDB. An application interested in navigation data subscribes to the variable `NAV_X` and receives a message from the MOOSDB containing the value of the variable as well as useful metadata such as the publishing application and time of publication. Some discussion of the impacts of this data exchange architecture is given in Section 5.1.

For autonomous vehicles, one of the most important functions for MOOS applications is vehicle guidance. In MOOS-IvP, this role is fulfilled by pHelmIvP ("the helm"), a behavior-based architecture where action selection among competing behaviors is achieved with multi-objective optimization using interval programming [6]. The helm makes guidance decisions by solving an optimization problem framed by one or more behaviors. Typical behaviors include waypoint following, (static) obstacle avoidance, and contact avoidance. Each behavior generates an objective function, assigning a utility value to each possible combination of speed and heading. (Although simple behaviors might only produce an objective function for one of these variables).

Of note for this work, the helm is configured with a specific discretization of the decision making domain. A typical configuration would be discretizing the heading domain to 1° and the speed domain to 0.1 m/s. Recalling Section 3.3.2, this illustrates the need for methods such as interpolation to reduce the number of discrete heading change and initial speed combinations that are empirically tested to populate the table.

4.2 TurnSequence Class

The TurnSequence C++ class is the fundamental data structure used in this work. It contains the meta-information needed to identify a turn, namely the turn speed and heading change. The turn trajectory is recorded using a list of X/Y coordinates.

A TurnSequence has two fundamental purposes. First is to serve as a record of a turn performed in order to ascertain vehicle kinematics. The TurnSequence is passed to functions to fit a polynomial to the data (Section 3.3.2). In order to provide consistent input to the curve fitting functions, the standard orientation described in Section 3.2 has been applied to TurnSequence's: the turn starts at the origin with an initial heading of North and the turn is made to the right. The turn records obtained during a data collection mission will be initiated at arbitrary X/Y coordinates and could be either left or right hand turns. The class includes functions to rotate, translate, and reflect an arbitrary turn to match the orientation convention.

When rotating the TurnSequence to be North facing, the turn is rotated based on the instantaneous heading at the start of the turn. If the vehicle is steady on the initial heading, the normalized turn trajectory will initially be perfectly vertical (due North), whereas if the vehicle had an initial yaw rate, the trajectory will diverge.

The TurnSequence class is also used to express the piecewise linear approximation of the polynomial fit to the source data (discussed in Section 3.3.2). The meta-data is the same as the original turn record, but the list of points is simplified. This type of TurnSequence is generated either by processing raw data or using the interpolation process described in Section 3.3.2. To evaluate vehicle trajectories with respect to obstacles, the rotation, translation, and reflection functions are used to match the TurnSequence to the vehicle's pose.

4.3 pDynamicsMonitor

pDynamicsMonitor is a MOOS App designed to coordinate the data collection mission run to collect AVK trajectories. It interacts with the ConditionSet, ConstantHeading, and ConstantSpeed behaviors. These behaviors are described in Section 4.4 and the overall coordination in the mission is described in Section 4.5. pDynamicsMonitor is configured with a range of heading changes and speeds directs each combination to be conducted. As turn data is collected, the resulting TurnSequence is logged for post processing.

4.4 Behaviors

MOOS-IvP utilizes a behavior based architecture. This allows each goal (such as waypoint following or collision avoidance) to be expressed individually. While this architecture generally performs multi-objective optimization from multiple competing behaviors, the data collection process was utilized several discrete helm modes, with no competing behaviors active simultaneously. This arose from the need to measure the response of the vehicle (controller and physical) with precisely known input from the backseat. Several existing behaviors were either utilized or modified, and two new behaviors were written.

4.4.1 ConditionSet

Section 6.2 discusses the importance of defining a specific vehicle state associated with the start of a particular turn. The ConditionSet behavior exists to establish the required state. For the simplest case, this means establishing a desired speed and a steady course. Steady course is determined by calculating yaw rate from `NAV_HEADING` postings.

$$\dot{\psi}(t_k) = \frac{\psi_{k-1} - \psi_k}{t_k - t_{k-1}} \tag{4.1}$$

Differentiating discrete heading readings to determine yaw rate was found to generate a highly noisy yaw rate, even in simulation. To smooth the data, a moving average was used to reduce the effect of noisy heading data.

$$\dot{\psi}_{avg} = \frac{1}{n} \sum_{k=0}^{n-1} \dot{\psi}_{n-k} \tag{4.2}$$

Where n is the window size of the filter. It was found that with navigation data published both at 4 Hz (in simulation) or 10 Hz (on the m200), a window size $n = 5$ gave good results.

ConditionSet is configured with a desired speed and acceptable speed tolerance and a maximum yaw rate. On becoming active, ConditionSet notes the current course. Until the configuration thresholds are achieved, ConditionSet generates an objective function to maintain course and achieve the desired speed.

4.4.2 Operating Region and Container

The OpRegion behavior is distributed with MOOS-IvP. It is configured with a polygon that the vehicle is to remain within. If the vehicle exits the polygon, it is shut down.

In testing, this keeps the vehicle close enough to be recovered by remote if a computer issue or bug occurs.

The operating region behavior generates no objective function (it is “inactive”) while the vehicle remains inside. In order to remain inside the operating region, other behaviors in conjunction with the mission structure are required. The OpReg Container behavior was developed for this work to keep the vehicle inside of the OpRegion. It is configured with the same polygon used by the OpRegion behavior as well as a buffer distance. Normally, this behavior is inactive. When the vehicle enters the buffer region, an objective function is produced with a desired heading towards the center of the OpRegion polygon. Once the vehicle has exited the buffer region, the container behavior becomes inactive again. No useful maneuvers can be performed while the container behavior is active, so it requests a very narrow heading range to quickly open range from the operating region boundary. It is not intended to compromise on courses that parallel the operating region boundaries.

4.4.3 Constant Heading and Speed

The ConstantHeading behavior does exactly what the name suggests. For this work, the notion of a completion threshold was important. When the heading error decreases below a configured value, the behavior is “complete.” If the vehicle was turning to achieve the desired constant heading, completion indicates that the turn is finished.

Depending on the magnitude of the threshold used, an absolute comparison can be problematic in real world conditions. It was observed that the test vehicle could “pass” the target heading without an intermediate heading reading that met the threshold. This resulted in a non-representative recorded turn trajectory. This problem was caused at least in part by the relative frequencies of various MOOS applications and stopped occurring when pHelmIvP’s AppTick was raised to 10.

The constant speed behavior is similar to Constant Heading behavior except there is no notion of completion. It requests a constant speed as long as it is active.

4.5 AVK Collection Mission

The idea of a table structure to store AVK trajectories was introduced in Section 3.2, and Section 3.3 addressed directly measuring turns as the means to fill this table. This section describes a mission where turns with certain heading changes and speeds are performed sequentially to observe and record their kinematic trajectories. A mission to collect AVK trajectories involves coordination between pDynamicsMonitor and the behaviors described in Section 4.4. The primary attribute of the mission is the `turn_array` parameter in pDynamicsMonitor, which describes the speed and heading combinations to be collected. A typical mission is shown in Figure 4-1.

The mission begins the vehicle dockside, just outside of the operating region. On deployment, the vehicle is initially governed by the OpRegion Container behavior. The Container behavior is configured with active and inactive flags such that the MOOS variable `CONTAINING` is true or false. Once the vehicle is inside of the buffer region configured in the container behavior, `CONTAINING` is set to false and the ConditionSet behavior becomes active.

Once speed and yaw rate are within the ConditionSet thresholds, the behavior posts `CONDITIONS_SET = true`. This message prompts pDynamicsMonitor to post updates to the Constant Speed and Constant Heading behaviors. To ensure that the helm does not shift modes and initiate the turn prior to the update postings, an additional posting from pDynamicsMonitor, `TURNING = true`, is used to control the helm mode. Finally, pDynamicsMonitor posts the update variable `NEXT_CONDITONS` to configure the ConditionSet behavior with the next target speed.

On receipt of `TURNING = true`, the helm mode shifts to `TURNING`. This activates the constant speed and constant heading behaviors. No other behaviors are active, therefore `DESIRED_SPEED` and `DESIRED_HEADING` are constant. The result is a pure turn, governed by the vehicle's hydrodynamics, the front seat, and any backseat layers such as a PID controller that calculate rudder or thrust commands. The turn is continued until the Constant Heading behavior reaches a complete state.

As the vehicle approaches the new desired heading, the vehicle trajectory is dominated by the heading PID controller settings rather than by hydrodynamics. This is generally not relevant for obstacle avoidance decisions: the guidance decision will be updated many times prior to that portion of the turn, and the trajectory is straight enough to be reasonably approximated by a ray at the goal heading. To avoid unnecessary delays to observe this gradual approach, the Constant Heading behavior is configured with a completion threshold. The turn is considered "complete" when the vehicle's instantaneous heading is within a configurable threshold of the goal.

At the initiation of the turn, pDynamicsMonitor records initial heading, speed, time, and the intended magnitude of the turn. During the turn, pDynamicsMonitor records position data. On receipt of TURN_FINISH, an endflag posted when the constant heading behavior completes, pDynamicsMonitor records the final heading and marks the turn for processing.

During processing, the data recorded by pDynamicsMonitor is stored in a TurnSequence. Error checking is performed to detect any anomalies such as an instantaneous "turn" (initial and final headings equal). Depending on when the turn start and stop postings are received relative to navigation mail, it is possible that the number of x and y coordinates received are mismatched. Any difference is adjudicated by ignoring the last coordinate received. It is assumed that any error introduced (for example if the first record was mismatched instead of the last) is small. While the heading change goal is the primary identifier for the turn, the observed heading change (which is smaller than the goal due to the Constant Heading completion threshold) is also stored in case it is needed for analysis.

After the TurnSequence is fully populated, a string serialization of the raw data is published under TURN_RECORD_RAW. The turn is then normalized and the normalized string serialization is published as TURN_RECORD.

In addition to to posting TURN_FINISH, the constant heading behavior posts an additional endflag to switch the helm mode. The ConditionSet behavior once again becomes active and starts attaining the speed and yaw rate that were published under NEXT_CONDITIONS at the start of the turn. The process is repeated for each turn specified by the pDynamicsMonitor configuration parameter turn_array.

Periodically while performing these turns, the vehicle will enter the buffer specified for the OpRegion Container behavior. The helm mode shifts to `CONTAINING` and the vehicle drives back towards the center of the operating region. The Container behavior is configured with an "activeflag," `CONTAINING = true`. If pDynamicsMonitor receives this posting during a turn, the turn data is discarded. The aborted turn is reported in the pDynamicsMonitor AppCast. The aborted turn's initial conditions are re-posted to `NEXT_CONDITIONS` to reset the ConditionSet behavior.

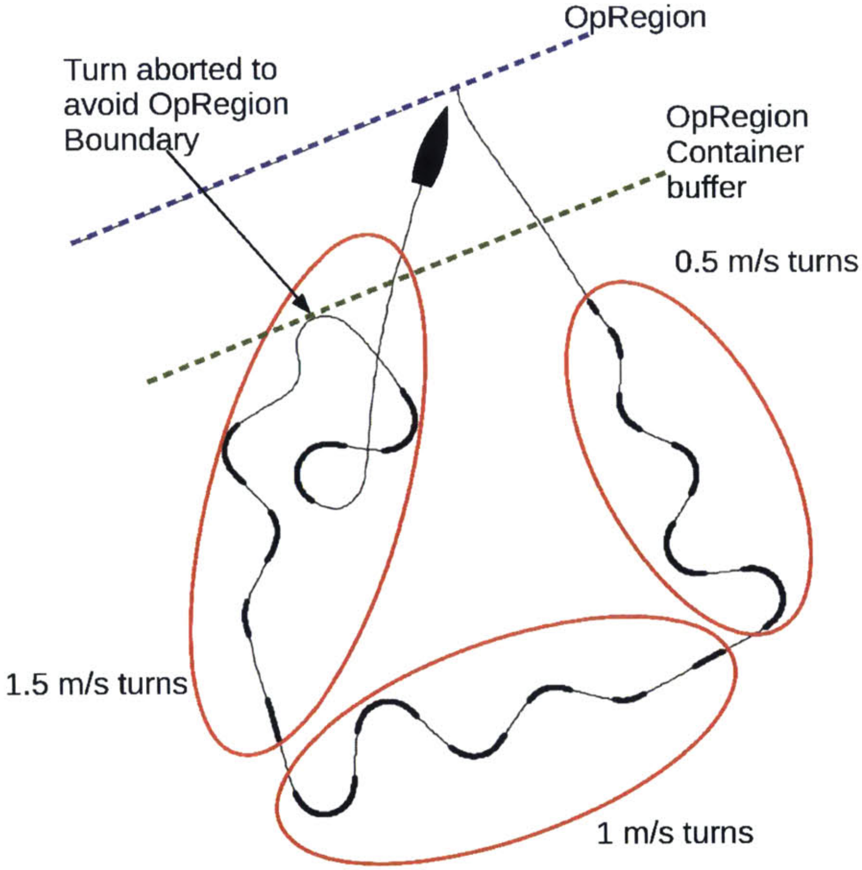

OpRegion
Turn aborted to avoid OpRegion Boundary
OpRegion Container buffer
0.5 m/s turns
1.5 m/s turns
1 m/s turns

Figure 4-1: **Simulation of Basic Mission:** The mission shown consists of 10, 40, 70, 100, 130, and 160 degree turns performed at 0.5, 1.0, and 1.5 m/s. The heavy lines show the recorded turns while the thin shows the path between turns. Between turns, there is a small delay while the vehicle regains steady speed. The OpReg Container behavior activates during the 1.5 m/s turns to keep the vehicle within the OpRegion. The Container behavior is deactivated at the end of the mission to allow the vehicle to return to the starting point.

4.6 Summary

This chapter has given a basic introduction to the MOOS-IvP autonomy software and described the applications and behaviors developed in this work to obtain AVK trajectories. A typical turn collection mission has been described. Variations of this basic turn collection mission will be used in the next chapter to obtain simulation and in-water AVK data.

Chapter 5

Experimental AVK Results

5.1 Results from Experimental Simulations

Simulations of the mission described in Section 4.5 were performed for debugging and analysis. In the simulation, the vehicle's dynamics are replaced by the MOOS application uSimMarine. The yaw characteristic in uSimMarine is given by:

$$\psi_{i+1} = K\left(\psi_i + C\delta\right) \tag{5.1}$$

where $100C =$ `turn_rate` (a configuration parameter) and K is a modification factor based on the percentage of full thrust requested at step i. K ranges from 0.5 at 0% thrust to 1.5 at 100% thrust, effectively making the vehicle turn faster at high speed. Key configuration parameters affecting the simulated yaw performance of the simulated vehicle are given in Table 5.1.

The nature of an AVK trajectory prediction is that it is a point estimate of a variable process. In simulation, variation is relatively small, introduced only by variations in inter-process communications. The asynchronous nature of the MOOS publish and subscribe architecture (Section 4.1) introduces a stochastic element into the vehicle's behavior, even in simulation. For example, PID control, performed by pMarinePID, was run at 4Hz, as was the propagation of the vehicle's position and pose, performed by uSimMarine. If pMarinePID runs immediately after uSimMarine, it will have rel-

Table 5.1: Key simulation parameters affecting maneuverability. In simulations for this work, the following parameters were used unless otherwise specified. A change in desired thrust is calculated once during every pMarinePID AppTick. The contribution from proportional gain is independent of time. If pMarinePID runs repeatedly with the same navigation information (i.e. if pMarinePID's AppTick is faster than uSimMarine's AppTick), it effectively raises the proportional gain of the speed controller.

App	Parameter	Values
uSimMarine	turn_rate	30
uSimMarine	AppTick	4
pMarinePID	YAW_PID_KP	1
pMarinePID	YAW_PID_KD	0
pMarinePID	YAW_PID_KI	0
pMarinePID	SPEED_PID_KP	1
pMarinePID	SPEED_PID_KD	0
pMarinePID	SPEED_PID_KI	0
pMarinePID	MAXRUDDER	40
pMarinePID	MAXTHRUST	100
pMarinePID	AppTick	4

atively "new" navigation data available to calculate the next control action. If, on the other hand, pMarinePID runs immediately before uSimMarine, the navigation data may be as much as 1/AppTick old (0.25 sec in this case). To quantify these interactions, a series of simulations were run where the AppTick of key applications (uSimMarine, pHelmIvP, and pDynamicsMonitor) was varied. Results are given in Figure 5-1. The main conclusion is that some degree of variation is expected, even in a simulation environment, simply due to the MOOS architecture. Variations in advance and transfer for simulated turns were small and can be attributed to this effect.

The variation shown in these results manifests primarily through several discrete results, with very minor variation in each discrete result. An example is given in Figure 5-2.

Given the relative consistency of simulated turns, the AVK table resulting from a simulated turn collection mission accurately represents the expected trajectories for turns initiated with constant heading.

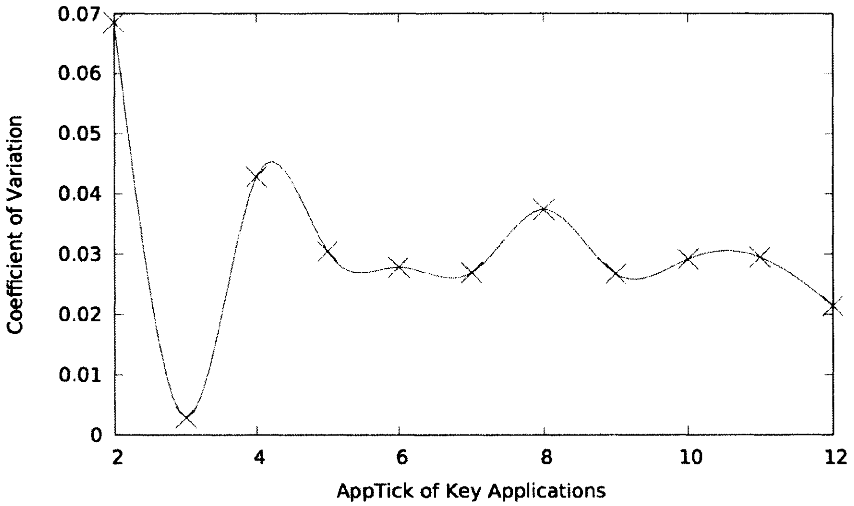


Figure 5-1: **AppTick influence on simulation consistency:** For each value of AppTick, approximately 70 turns (all 90 degrees and 1 m/s) were simulated. The varying nature of the turn trajectories is well summarized by differences in the advance for the turn. The y value is the coefficient of variation (mean/standard deviation) for turn advance at each AppTick. The results at AppTick = 2 and 3 are attributed to changes in the effective proportional gain in the speed controller. The increase at AppTick=8 is suspected to occur because it is an even multiple of pMarinePID's AppTick (4), but this was not verified.

5.2 In-Water Observed Characteristics

The data collection mission described in Section 4.5 was performed several times on an M200. There were several goals for this testing. First, testing served to verify that the ConditionSet behavior described in Section 4.4.1 works as intended with real world environmental effects and sensor noise. Second, testing was performed to evaluate the variability in observed turning characteristics over repeated trials.

With appropriate values for configuration parameters, the ConditionSet behavior performed as intended in real world conditions.

It is again important to stress that an observed trajectory, used as an AVK prediction, is a point estimate. The accuracy of the prediction depends on the variability of repeated turns. Unlike simulation, where only small variations from process communications were observed, variability in the M200 data was much higher. This is attributed primarily to environmental effects. Wind and current influence the turns, but are not constant in magnitude or direction. Furthermore, the vehicle's orientation with respect to the current varies between repeated evaluations of a given heading

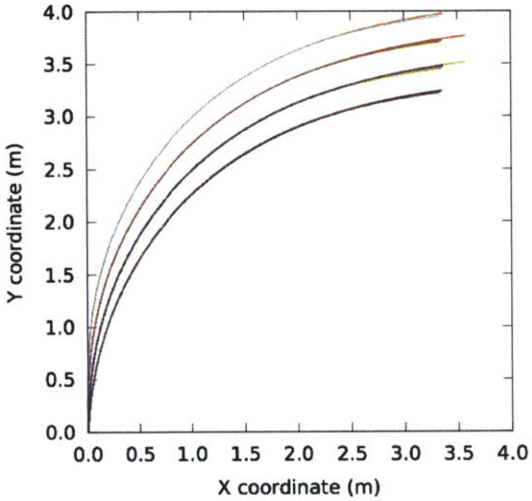


Figure 5-2: **Variation in simulation results:** A 90 degree turn at 1 m/s was simulated 77 times. Each result is plotted here in a separate color. Key applications used an AppTick of 4. Four discrete paths resulted, with minor variations in each path.

change. An example is given in Figure 5-3. These results are typical for all turns. Similar figures for all measured turns are given in Appendix A.

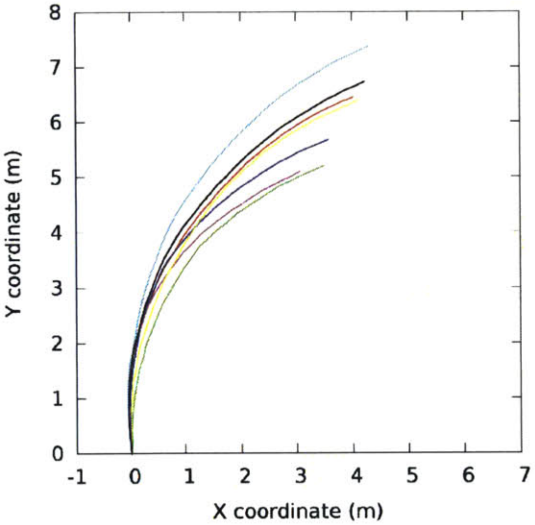


Figure 5-3: **M200 in-water result:** Observed trajectories for several 70 degree, 1 m/s turns. Each line shows the turn as observed in a separate mission.

The variability can be summarized in a few different ways. Figures 5-4 and 5-5 present advance and transfer values observed during test runs of the M200. This type of summary benefits from simplicity: each turn is reduced to these two numbers. An alternate possibility is to determine a "mean" turn. One challenge in defining a mean

Table 5.2: **M200 in-water result:** The average (μ_t) and standard deviation (σ_t) of the time spent establishing turn initial conditions for several data collection runs. The speed tolerance is given in the last column; in all case the maximum yaw rate was 1 deg/s. These results are considered adequate and validate the performance of the ConditionSet Behavior with these settings.

Mission	μ_t	σ_t	v_{tol}
24-Apr 0930	7.10	6.87	0.1
24-Apr 0943	7.62	6.60	0.1
27-Apr 1010	6.79	7.62	0.1
27-Apr 1051	4.81	5.90	0.2
27-Apr 1102	7.14	7.30	0.2
27-Apr 1115	3.83	3.89	0.2
27-Apr 1152	5.30	5.80	0.2
27-Apr 1209	5.04	5.56	0.2
27-Apr 1223	5.41	6.32	0.2

turn is choosing a domain over which to average. One option would be averaging both X and Y position at each time step. For the present analysis, the trajectories were resampled to a constant X spacing and an average Y coordinate found for each X. In order to provide a consistent X domain across individual turns, each turn was extended along the final heading goal to the maximum X value in the set. An example showing the calculated mean turn is given in Figure 5-6.

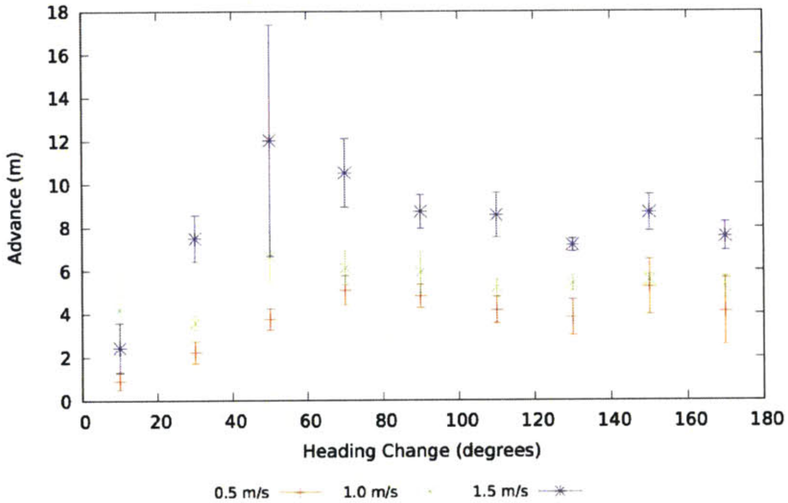


Figure 5-4: **M200 in-water result:** Observed advance characteristics from 7 experiments on the M200 operating at various speeds. The mean advance from the 7 runs is shown with error bars indicating the standard deviation in the observed measurements. Several outlier turns contributed significantly to the overall statistics. (For example, at 50 deg/1.5 m/s)

As a basis for comparison between techniques, the calculated mean turns are

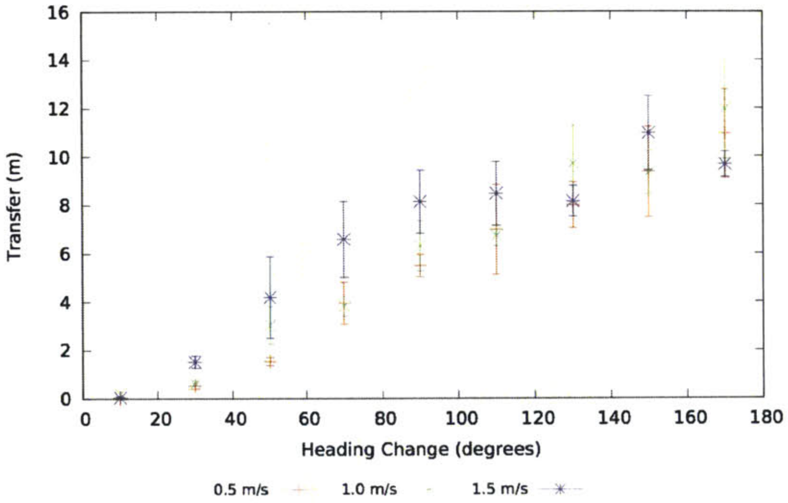


Figure 5-5: **M200 in-water result:** Observed transfer characteristics from 7 experiments on the M200 operating at various speeds. The mean advance from the 7 runs is shown with error bars indicating the standard deviation in the observed measurements.

considered to represent the true characteristics of the vehicle. Comparisons are made using mean square error (MSE) between y values using an x domain like that described above to find the mean turn. The AVK approach, as described, relies on a single observed trajectory. The average MSE between each of the measured turns and the corresponding mean turn is therefore an appropriate error metric. For the black box yaw model, a trajectory is simulated using a discrete change in desired heading. The MSE between the modeled trajectory and the mean turn follows. Finally, the current practice is to simply assume instantaneous turns. MSE between the mean turn and the instantaneous turn (a straight line) can also be calculated. An example is given in Figure 5-7. Results for 1 m/s turns are given in Figure 5-8. Results for 0.5 m/s and 1.5 m/s are included in Appendix A.

5.3 Summary

This chapter has described analysis performed on simulation and in-water AVK data. The inherent variability in an AVK trajectory, even in simulation, has been demonstrated. The performance of the basic data collection mission developed in this thesis has been validated by in-water experiments using the M200 USV. Data from sev-

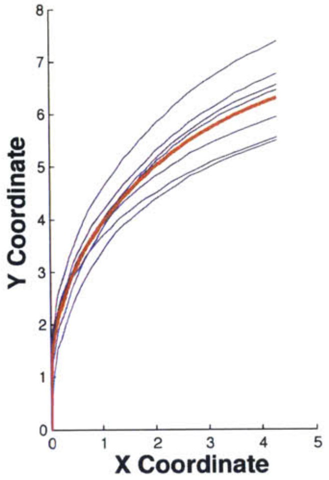


Figure 5-6: **Analysis of in-water results:** The experimentally observed turns shown in Figure 5-3 (shown here in blue) were extended along the goal heading angle (70 degrees) and sampled every 0.01 m in the x domain. The resulting mean turn is shown in red. The average MSE between the input turns and the mean turn is 0.23 m^2.

eral data collection missions was aggregated and used to show the variability of turn kinematics when real world factors such as environmental forces are included. Mean kinematic trajectories associated with certain turns on the M200 USV have been determined from this data. Finally, the accuracy of the AVK approach and the black box model developed in Chapter 3.4 were compared to the current instantaneous turn assumption.

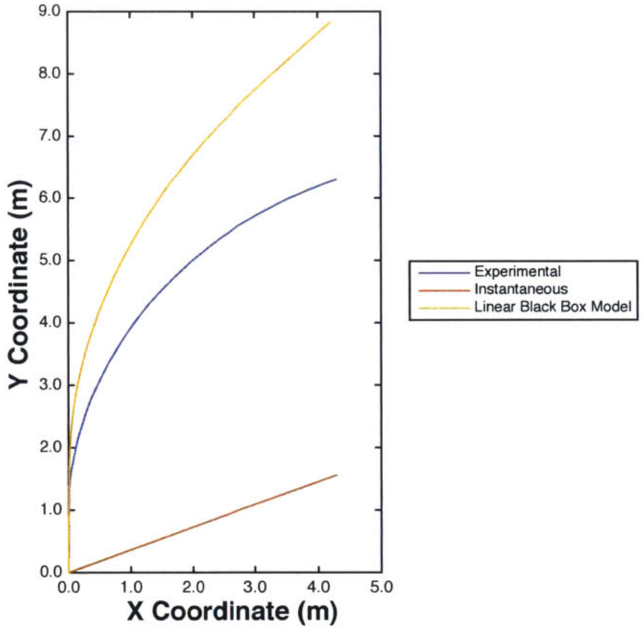


Figure 5-7: **Comparison of prediction methods:** The mean turn determined from the experimental data in Figure 5-6 is compared against the model prediction and the current practice of assuming an instantaneous turn. The 0.23m^2 average MSE between the source turns and the mean reflects the expected accuracy of an AVK estimate (if measured under reasonable conditions). The MSE between the model prediction and the mean turn is 3.2 m^2. The MSE between the instantaneous assumption and the mean turn is 16.8 m^2. This suggests that the prediction accuracy can be improved by 80% using the model and over 98% (on average) using an observed AVK trajectory.

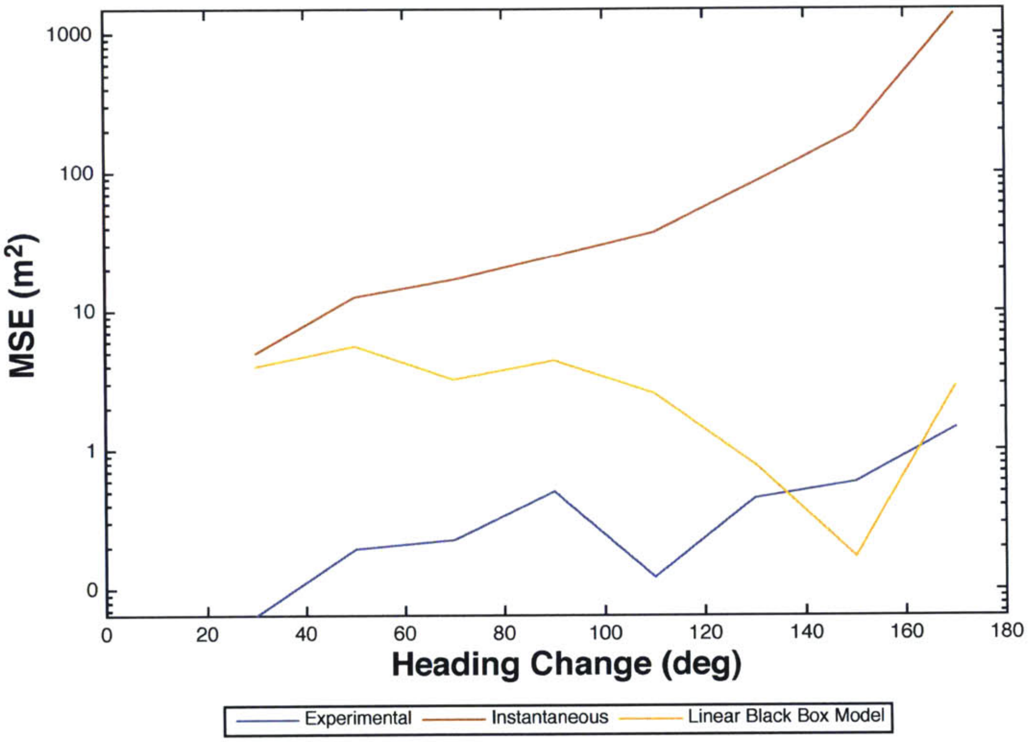


Figure 5-8: **Comparison of prediction methods:** The MSE of the experimental data (blue) shows the inherent variability in simply measuring a turn in the presence of environmental forces. Note that the significant error in assuming an instantaneous turn for large turns requires using a log scale to compare values. The trend in MSE for instantaneous turns is as expected - larger turns have longer trajectories before steadying on course, resulting in very large errors during the latter part of the turn trajectory. The trajectories determined using the black box model offer substantial improvements over the current practice, but still have non-trivial errors, especially for smaller turns.

Chapter 6

Future Work

6.1 Utility of AVK

The present work has focused on collecting and processing AVK data. The examples given in Section 2.1.3 show notional cases where this knowledge could be directly useful in improving guidance actions. Work to integrate AVK predictions into decision making is ongoing, and the effects on decision making will be studied and reported in separate work.

In particular, clear test cases where poor guidance decisions result from the assumption of instantaneous turns must be identified. This would provide a clear basis for evaluating the benefit afforded by adding accurate kinematic predictions. Furthermore, it could provide insight into appropriate weights used for better parameter estimation cost functions as described in Section 3.4.3.

6.2 Vehicle State

In Section 3.2, the initial vehicle state prior to a turn was restricted to $\dot{\psi} = 0$. This is adequate for initial assessments of potential maneuvers for a vehicle on a steady course. However, the utility of the resulting kinematic trajectories is significantly limited by this assumption. Two separate cases are considered, with consideration for how to extend the AVK approach.

First is the ongoing evaluation of a maneuver. Typical obstacle avoidance maneuvers ($\Delta\psi \leq 90$) are on the order of 30 seconds long. The typical guidance system operates multiple times per second, and once a maneuver is chosen must continually monitor whether it is adequate. As the vehicle proceeds along the turn trajectory, a new trajectory prediction cannot be made under the existing set of AVK assumptions. However, provided that the goal heading has not changed, the original trajectory prediction remains valid. The position of the vehicle along the trajectory can be estimated as time passes, and the remaining portion of the trajectory remains useful.

However, in many cases a new maneuver may be considered when the vehicle is not steady on course. If $\dot{\psi}_0$ is in direction as the intended direction of turn, the $\dot{\psi} = 0$ trajectory, although not accurate, could still be used as a conservative estimate with respect to evaluating obstacle avoidance. On the other hand, if initial yaw rate is opposite to the intended direction of turn, the vehicle's advance will be larger than the case where $\dot{\psi} = 0$ and therefore non-conservatively predicts the ability to avoid obstacles.

To handle situations where $\dot{\psi} \neq 0$, the CKT could be expanded into a third dimension. Just as trajectories resulting from certain $[u, \Delta\psi]$ combinations were recorded with $\dot{\psi} = 0$, similar data can be obtained with $\dot{\psi} \neq 0$. Similar interpolation techniques to those used in the $\Delta\psi$ domain could be used across the $\dot{\psi}$ domain to minimize the additional data collection requirements to obtain this third dimension. The black box model, if refined to improve accuracy, offers a convenient alternative to attempting to gather high quality data to fill a three dimensional CKT. Even if model-based trajectory prediction is too slow to run in real time inside the guidance algorithms, it could be used to easily populate a 3D CKT.

The current approach also assumes that the commanded speed is constant throughout a turn. In reality, it may often be appropriate to slow down while turning, especially for close quarters situations where the turn kinematics are significant. Future work should include a technique to approximate the kinematics of a simultaneous speed and heading change.

6.3 Dealing with External Forces

Vehicle kinematics in the presence of wind and current can vary substantially from the case with no external forces. This influences both data collection and use of kinematic approximations. As with complicating factors like yaw rate and speed change, the use of a dynamics model would simplify generating accurate predictions with external forces.

An additional challenge for low cost vehicles is that they often lack the navigation equipment necessary to determine the influence of external forces. For example, a vehicle with only GPS navigation data can only determine course and speed in the earth fixed reference frame. A compass and speed measurement (or at least estimate) on the vehicle are necessary to quantify external forces.

Finally, if an accurate AVK table has been established, a vehicle could compare actual trajectories to the AVK predictions and use the differences to estimate external forces.

Chapter 7

Conclusions

A new approach to estimating the kinematics of marine vehicles has been described. This approach affords the ability to approximate a vehicle's kinematics and can provide improved accuracy in determining expected standoff distance from a close range obstacle or contact. The AVK table can be determined in the field with any vehicle configuration and using data available to the backseat driver and is therefore compatible with payload autonomy.

As an alternative to explicitly measuring each turn trajectory, a simple black box model for the yaw kinematics of the vehicle was developed. As with the AVK table approach, this model only requires data available in the backseat.

Performance of the data collection mission was verified with on water testing. Both the AVK approach and the black box model were capable of significantly reducing the error in short range turn predictions.

Appendix A

M200 Summary Data

The following figures present the collected M200 data. The individual turns arc shown in blue. The "mean turn" developed using the procedure outlined in Section 5.2 is shown in red.

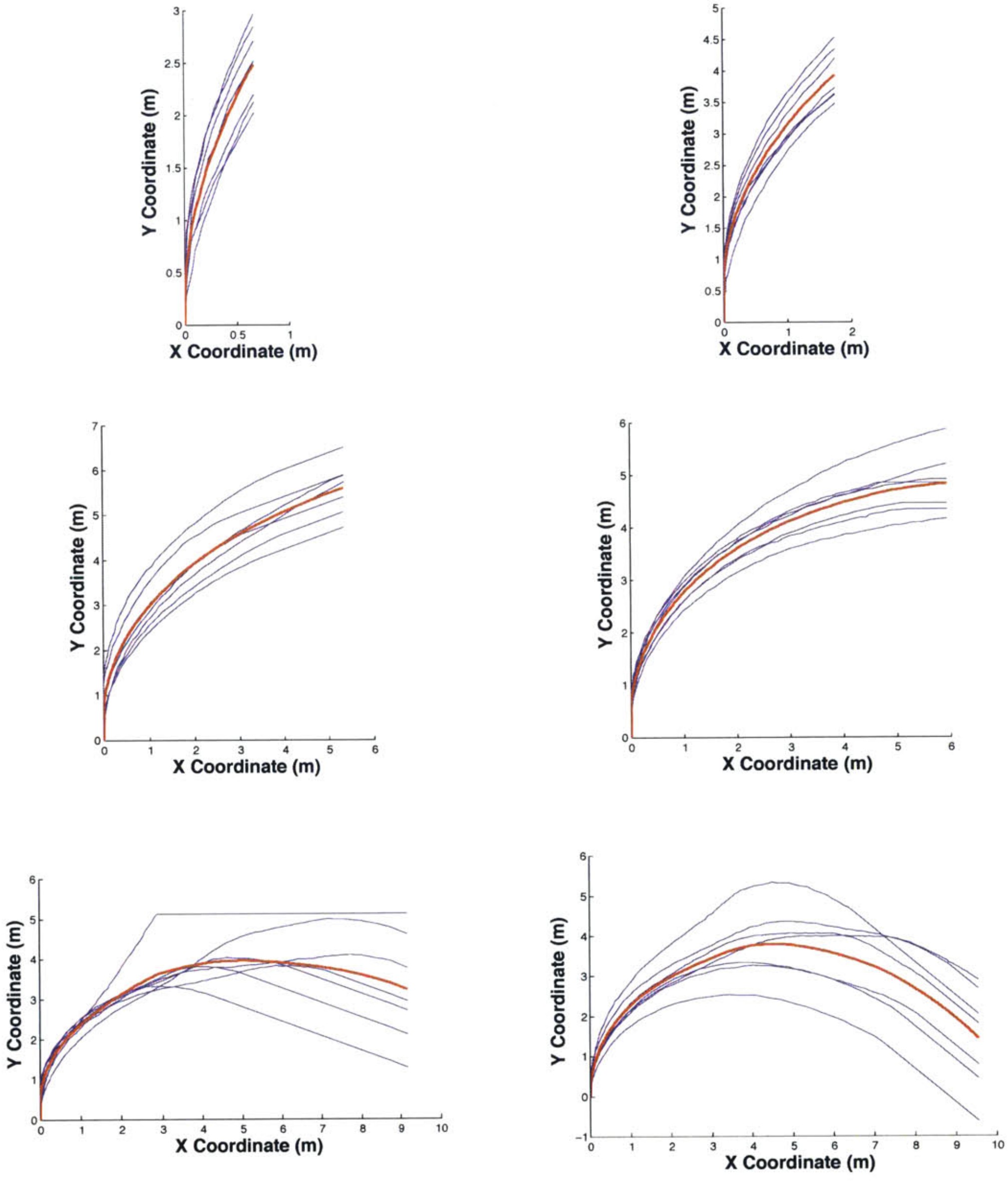


Figure A-1: **M200 in-water results:** M200 turns at 0.5 m/s. From left to right and top to bottom, heading changes of 30, 50, 70, 90, 110, and 130 degrees

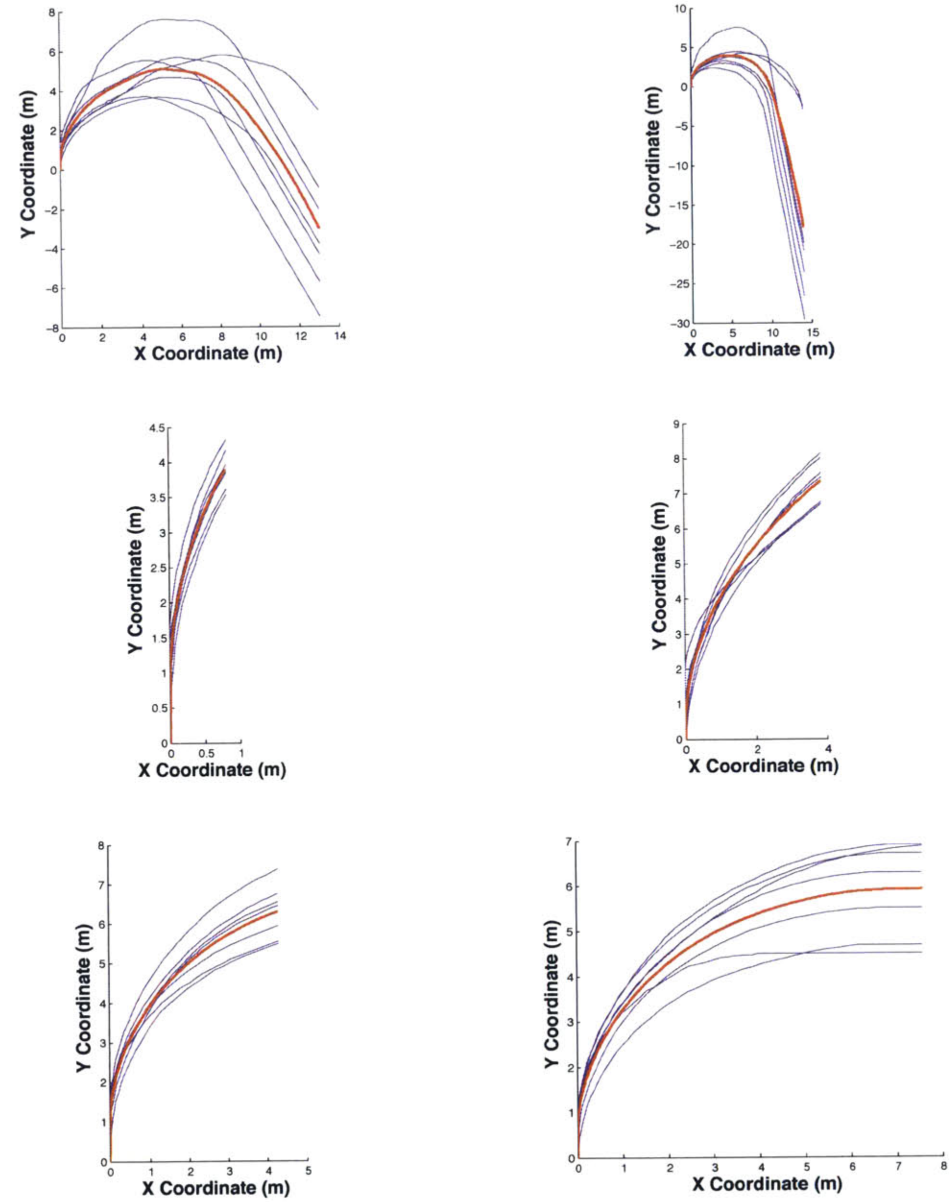


Figure A-2: **M200 in-water results:** From left to right and top to bottom, 0.5 m/s turns at 150 and 170 degrees, then 1 m/s turns at of 30, 50, 70, and 90 degrees

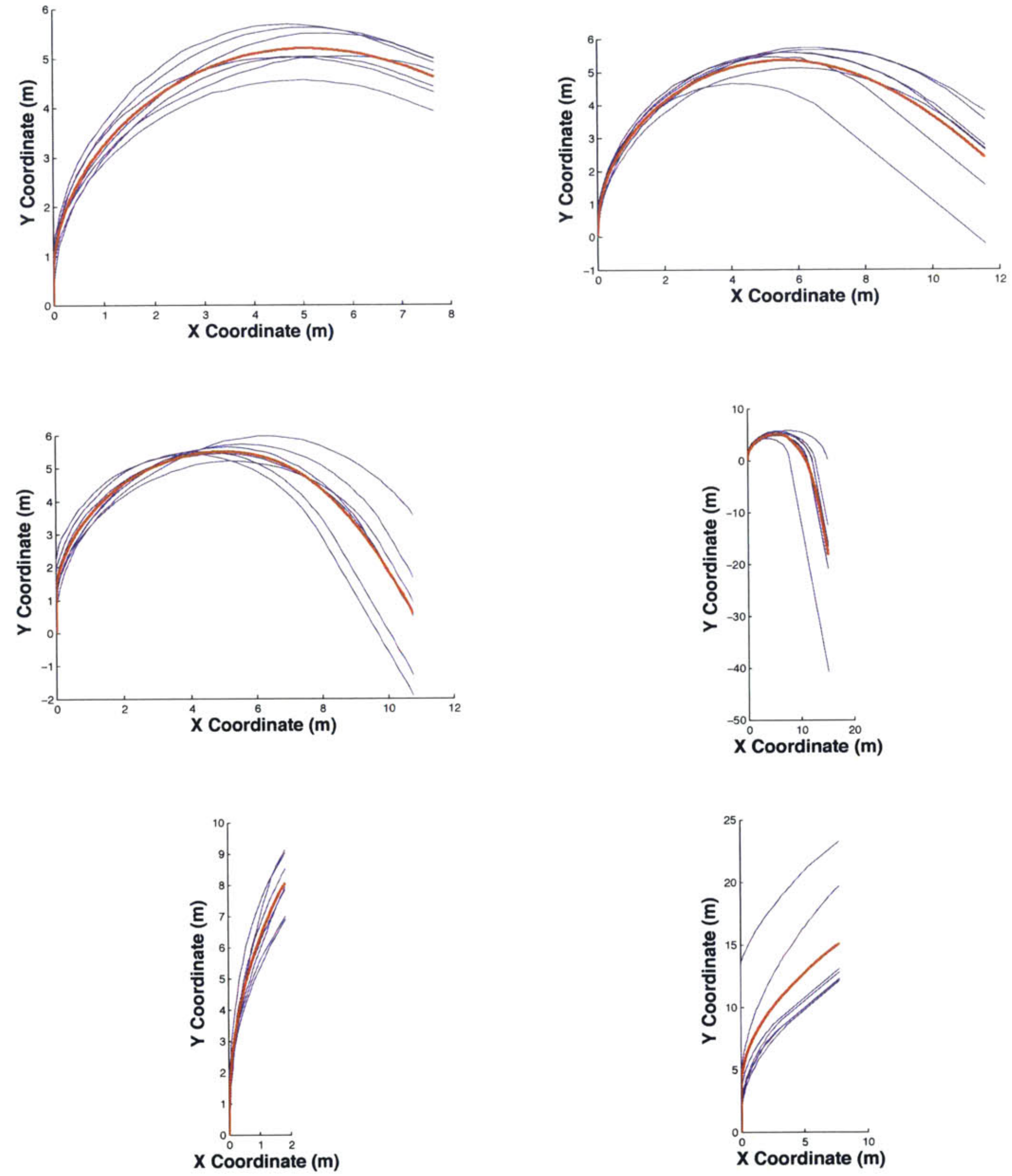


Figure A-3: **M200 in-water results:** From left to right and top to bottom, 1 m/s turns at 110, 130, 150 and 170 degrees, then 1.5 m/s turns at 30 and 50 degrees

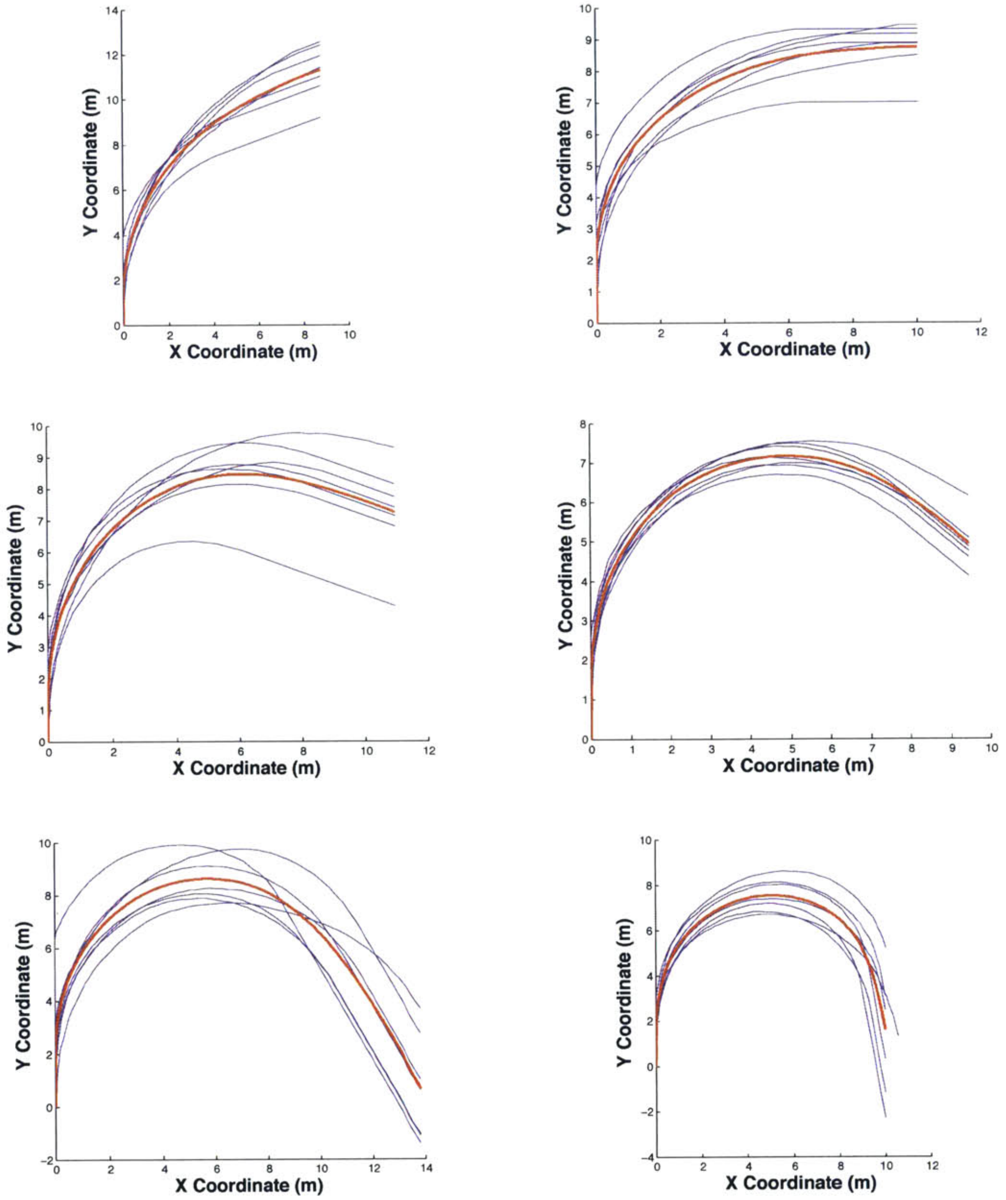
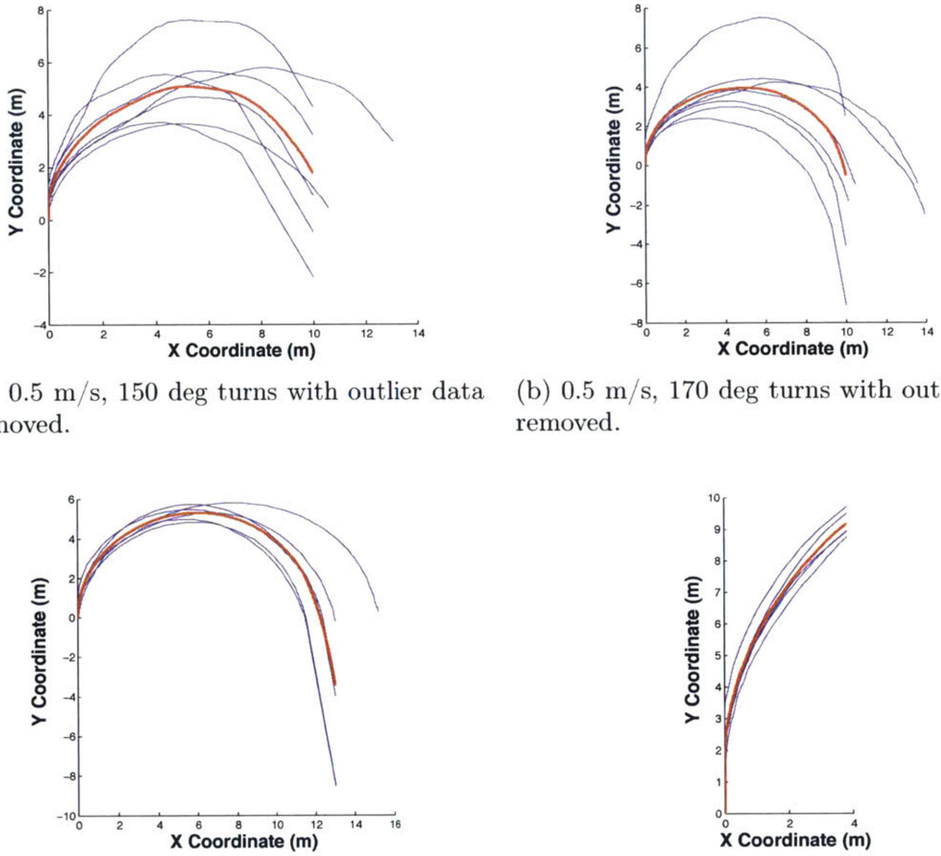


Figure A-4: **M200 in-water results:** From left to right and top to bottom, 1.5 m/s turns at 70, 90, 110, 130, 150, and 170 degrees



(a) 0.5 m/s, 150 deg turns with outlier data removed.

(b) 0.5 m/s, 170 deg turns with outlier data removed.

(c) 1.0 m/s, 170 deg turns with outlier data removed.

(d) 1.5 m/s, 50 deg turns with outlier data removed.

Figure A-5: **M200 in-water results:** Turns with outlier data removed. This is the same data given above, but with the x domain bounded or outlier turns removed.

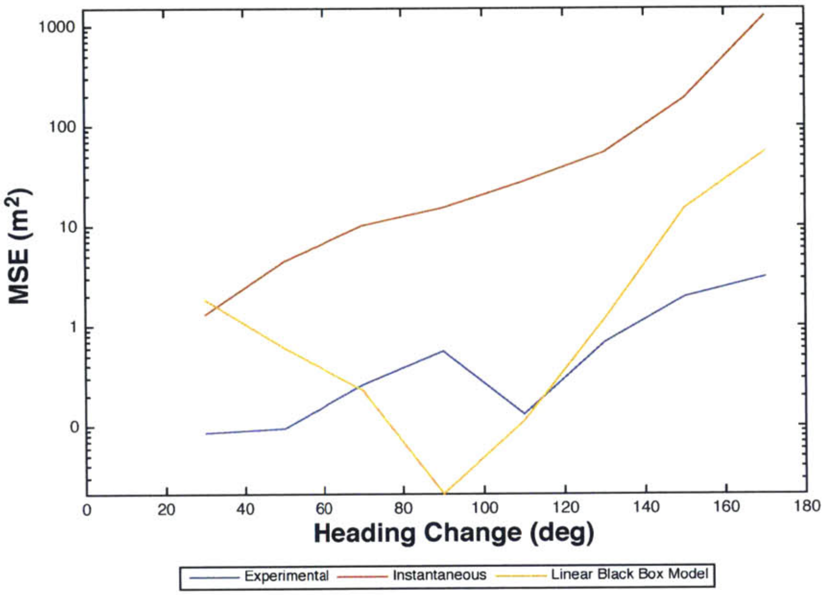


Figure A-6: **Analysis of M200 in-water results at 0.5 m/s:** The MSE of the experimental data (blue) shows the inherent variability in simply measuring a turn in the presence of environmental forces. The trend in MSE for instantaneous turns is as expected - larger turns have longer trajectories before steadying on course, resulting in very large errors during the latter part of the turn trajectory. The trajectories determined using the black box model offer substantial improvements over the current practice, but still have non-trivial errors, especially for smaller turns.

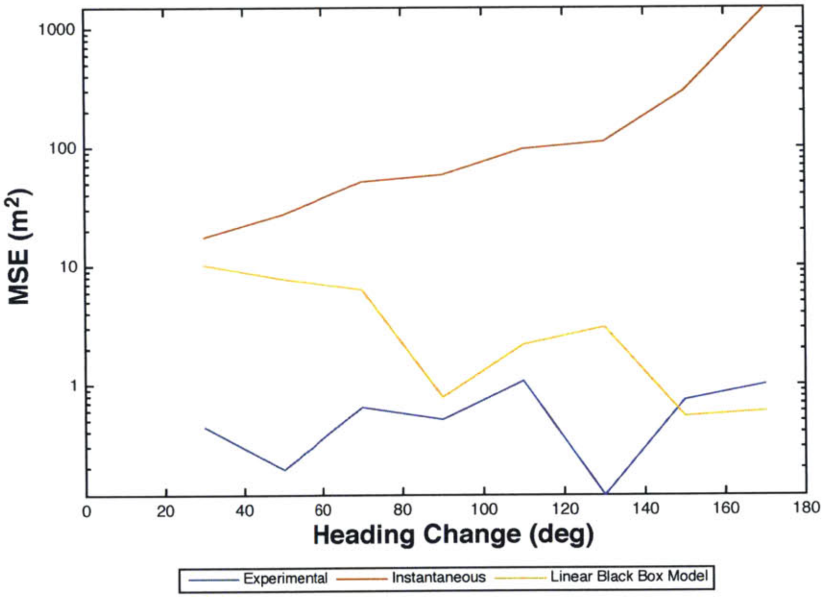


Figure A-7: **Analysis of M200 in-water results at 1.5 m/s:** The black box model had better performance at 1.5 m/s, but still tends to over-estimate small turns.

Bibliography

[1] A. Alessandri, M. Caccia, G. Indiveri, and G. Veruggio. Application of LS and EKF techniques to the identification of underwater vehicles. In *Proceedings of the 1998 IEEE International Conference on Control Applications, 1998*, volume 2, pages 1084–1088 vol.2, September 1998.

[2] Hashem Ashrafiuon, Kenneth R. Muske, and Lucas C. McNinch. Review of nonlinear tracking and setpoint control approaches for autonomous underactuated marine vehicles. In *American Control Conference (ACC), 2010*, pages 5203–5211. IEEE, 2010.

[3] Karl Johan Åström and C. G. Källström. Identification of ship steering dynamics. *Automatica*, 12(1):9–22, 1976.

[4] Michael R. Benjamin. MOOS-IvP Online Documentation : Helm - Behavior Avd Collision. `http://oceanai.mit.edu/ivpman/pmwiki/pmwiki.php?n=Helm.BehaviorAvdCollision`, 2015.

[5] Michael R. Benjamin, Henrik Schmidt, Paul M. Newman, and John J. Leonard. Nested autonomy for unmanned marine vehicles with MOOS-IvP. *Journal of Field Robotics*, 27(6):834–875, 2010.

[6] Michael Richard Benjamin. *Interval programming: a multi-objective optimization model for autonomous vehicle control.* PhD thesis, Brown University, 2002.

[7] N. Bowditch. *The American Practical Navigator: An Epitome Of Navigation.* The National Imagery and Mapping Agency, bicentennial edition, 2002.

[8] S. Campbell, W. Naeem, and G. W. Irwin. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2):267–283, December 2012.

[9] U. C. G. Commandant. International regulations for prevention of collisions at sea (COLREGS). *US Department of Transportation, US Coast Guard, COMMANDANT INSTRUCTION M16672*, 1999.

[10] Thor I. Fossen. *Guidance and control of ocean vehicles*, volume 199. Wiley New York, 1994.

[11] Jacoby Larson, Michael Bruch, Ryan Halterman, John Rogers, and Robert Webster. Advances in autonomous obstacle avoidance for unmanned surface vehicles. Technical report, DTIC Document, 2007.

[12] Steven M LaValle. *Planning algorithms.* Cambridge university press, 2006.

[13] P. Miotto, J. Wilde, and A. Menozzi. UUV on-board path planning in a dynamic environment for the Manta test vehicle. In *OCEANS 2003. Proceedings*, volume 5, pages 2454–2461 Vol.5, September 2003.

[14] Nikola Mišković, Zoran Vukić, Marco Bibuli, Gabriele Bruzzone, and Massimo Caccia. Fast in-field identification of unmanned marine vehicles. *Journal of Field Robotics*, 28(1):101–120, January 2011.

[15] Clement Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane. Path Planning for Autonomous Underwater Vehicles. *IEEE Transactions on Robotics*, 23(2):331–341, April 2007.

[16] P. Svec, B.C. Shah, I.R. Bertaska, J. Alvarez, A.J. Sinisterra, K. von Ellenrieder, M. Dhanak, and S.K. Gupta. Dynamics-aware target following for an autonomous surface vehicle operating under COLREGs in civilian traffic. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3871–3878, November 2013.

[17] Petr Švec, Atul Thakur, Eric Raboin, Brual C. Shah, and Satyandra K. Gupta. Target following with motion prediction for unmanned surface vehicle operating in cluttered environments. *Autonomous Robots*, 36(4):383–405, November 2013.

[18] Kyle Woerner. *COLREGS-compliant autonomous collision avoidance using multi-objective optimization with interval programming.* Thesis, Massachusetts Institute of Technology, 2014.