

AN ABSTRACT OF THE THESIS OF

Daniel C. Fernández for the degree of Master of Science in Robotics presented on September 14, 2015.

Title: Model Predictive Control for Underwater Robots in Ocean Waves

Abstract approved: _____

Geoffrey A. Hollinger

Underwater robots beneath ocean waves can benefit from feedforward control to reduce position error. This thesis proposes a method using Model Predictive Control (MPC) to predict and counteract future disturbances from an ocean wave field. The MPC state estimator employs a Linear Wave Theory (LWT) solver to approximate the component fluid dynamics under a wave field. Wave data from deployed ocean buoys is used to construct the simulated wave field. The MPC state estimator is used to optimize a set of control actions by gradient descent along a prediction horizon. The optimized control input minimizes a global cost function, the squared distance from the target state. The robot then carries out the optimized trajectory with an emphasis on real-time execution. Several prediction horizons are compared, with a horizon of 0.8 seconds selected as having a good balance of low error and fast computation. The controller with the chosen prediction horizon is simulated and found to show a 74% reduction in position error

over traditional feedback control. Additional simulations are run where the MPC takes in noisy measurements of the wave field parameters. The MPC algorithm is shown to be resistant to sensor noise, providing a mean position error 44% lower than the noise-free feedback control case.

©Copyright by Daniel C. Fernández
September 14, 2015
All Rights Reserved

Model Predictive Control for Underwater Robots in Ocean Waves

by

Daniel C. Fernández

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented September 14, 2015
Commencement June 2015

Master of Science thesis of Daniel C. Fernández presented on September 14, 2015.

APPROVED:

Major Professor, representing Robotics

Head of the School of Mechanical, Industrial, and Manufacturing Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Daniel C. Fernández, Author

ACKNOWLEDGEMENTS

This work was developed for the NNMREC ALFA Project and is supported in part by Department of Energy grant DE-EE-0006816.0000.

Additional support was provided by PCC Structural (special thanks to Jim Barrett), Oregon Metals Initiative funding from the State of Oregon, the Office of Naval Research, and the W. M. Keck Foundation.

TABLE OF CONTENTS

	<u>Page</u>
1 Introduction	1
2 Background	5
2.1 Wave Energy	5
2.2 Wave Mechanics	7
2.3 Feedforward Control	8
2.4 Underwater Robotics	10
2.4.1 Path Planning	11
2.4.2 Localization	15
2.4.3 Perception	17
2.4.4 Combined Systems	20
3 Method	25
3.1 Model Predictive Control	25
3.2 Vehicle Model	27
3.3 State Space Form	31
3.4 Wave Field Model	32
3.5 MPC Algorithm Layout	36
4 Results	38
4.1 Determination of Prediction Horizon	38
4.2 MPC Performance	41
4.3 Sensor Noise Impact	44
5 Conclusion	46

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1.1 NNMREC is a partnership between the U.S. Department of Energy, Oregon State University, the University of Washington, and the University of Alaska Fairbanks. NNMREC operates and manages NETS and SETS among other facilities.	2
1.2 A visualization of the flow field velocity at various depths beneath a monochromatic wave. These vector lengths are not to scale and are intended to conceptualize the direction and decay with depth of wave forces.	3
2.1 A Webb Research Slocum Glider. This AUV uses only a buoyancy differential and small attitude adjustments to achieve an ultra low-wattage (0.5-1.0 Watts) and long range mission profile.	11
2.2 A Bluefin Hovering Autonomous Underwater Vehicle, or HAUV, as referenced in [29] and [26]. This vehicle was used along with a Didson sonar to reconstruct the <i>S.S. Curtiss</i> hull draft.	17
2.3 A YSI EcoMapper AUV as referenced in [47] and [23] which along with a side-scanning sonar was used to construct dense bathymetric maps of the floor of Puddingstone Lake, CA.	21
3.1 A SeaBotix vLBV300 ROV similar to the one modeled in this work. Along with IMU and DVL sensors, the vehicle has six angled thrusters which control it along five degrees of freedom: heave (vertical z), sway (lateral y), surge (lateral x), roll (rotation around x), and yaw (rotation around z).	29
3.2 A CAD model of the robot used to generate mass and hydrodynamic properties from SolidWorks and AQWA, respectively.	30
3.3 Four second sample sets of AWACS acoustic heave data from NETS, 2013. The simulated wave field is built with similar wave heights and gradients.	33
3.4 A time series of the wave profile over the 240 s simulated time window constructed using the parameters listed in Table 3.2.	34

LIST OF FIGURES (Continued)

<u>Figure</u>	<u>Page</u>
4.1 Simulator visualization of a displaced robot (the bold “X”) attempting to reach the target state (the faded “X”). Wave field profile data is provided as well as current vehicle state and velocity vector. . . .	39
4.2 Position error time series in global \mathbf{x} and \mathbf{z} coordinates when comparing a traditional feedback controller with a model predictive controller. Note the axis scale difference. As shown, MPC returns error values 74% lower than PD Control. RMS error values for these results are shown in Fig 4.3.	41
4.3 RMS errors for the three cases in Section 4.2. The MPC robot showed a 74% reduction in error compared to another using PD control.	42
4.4 MPC thruster inputs in global \mathbf{x} and \mathbf{z} coordinates over the simulated time series. The controller does not issue commands that saturate the thrusters for prolonged periods; however, it is prone to frequent direction changes.	43

LIST OF TABLES

<u>Table</u>		<u>Page</u>
3.1	Vehicle Parameters used in Simulation	28
3.2	Wave Field Parameters used in Simulation	34
4.1	Performance of Various Simulated Horizons	39
4.2	Performance of MPC with Noisy Wave Observations	45

LIST OF ALGORITHMS

<u>Algorithm</u>	<u>Page</u>
1 MPC Algorithm for Ocean Wave Station-Keeping	36

Chapter 1: Introduction

The coastal ocean is bounded by the shoreline and the 200m isobath, and is a common experimental setting for field robotics. This is the most biologically productive area of ocean and is subject to the majority of natural and industrial disasters [39]. Its proximity to the shore and coastal communities adds to a variety of economic, military, and energy research areas under development. The increased demand for advancement in all of these coastal research areas predicates the need to better understand the dynamics of this environment.

An area of particular coastal interest is in wave energy extraction. The increasing demand for alternative energy sources has created new opportunities for this constant and powerful energy source. However, its appeal is curbed when compared to the cost of deployment and maintenance of Wave Energy Converters (WEC). One goal of the National Northwest Marine Renewable Energy Center (NNMREC) ALFA project aims to explore new avenues to increase the cost-effectiveness of WEC maintenance by employing autonomous robotic platforms to inspect, monitor, and intervene. Field experiments are currently carried out in various areas of research at the NNMREC North Energy Test Site (NETS) two nautical miles off the Oregon Coast (as shown in Fig. 1.1).

One challenge for robotic maintenance of the WECs is proper station-keeping under the influence of a surface wave field. Wave forces in the intermediate depths



Figure 1.1: NNMREC is a partnership between the U.S. Department of Energy, Oregon State University, the University of Washington, and the University of Alaska Fairbanks. NNMREC operates and manages NETS and SETS among other facilities.

of the coastal ocean will displace a robot throughout the majority of the water column. These forces decay exponentially from the water surface (as shown in Fig. 1.2), and sufficient depths yield negligible disturbances [14]. Because of this decay, as well as their cyclic nature, wave forces are often neglected in robotic path planning. In field applications where there is a low operational depth, a persistent wave climate, and strict localization constraints, such as those found at NETS, this assumption can quickly break down.

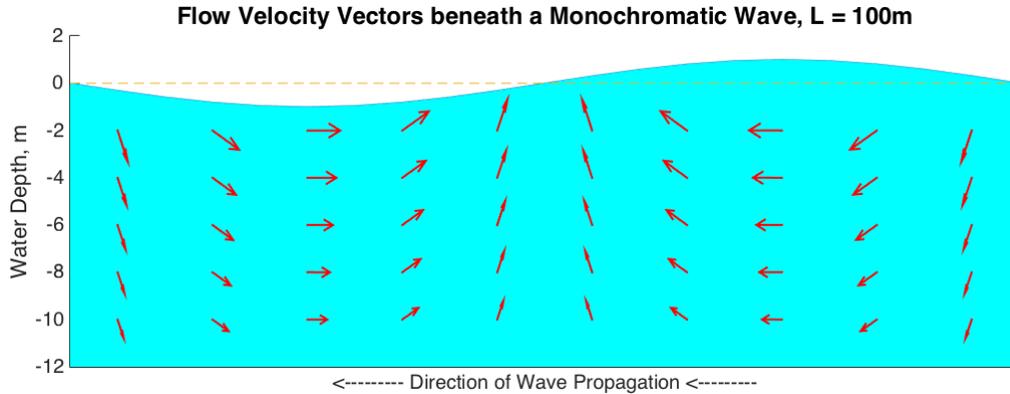


Figure 1.2: A visualization of the flow field velocity at various depths beneath a monochromatic wave. These vector lengths are not to scale and are intended to conceptualize the direction and decay with depth of wave forces.

Robotic inspection and manipulation tasks often require some form of sensory observation, be it acoustic, visual, etc., which has an associated computation time. These observations are paired with positional, or localization, data. Wave-induced disturbances lend to increased sensor drift which hinder the quality of robotic observations, such as those needed to close SLAM loops in [29] and [11].

Traditional Proportional Integral Derivative (PID) control techniques can be used to counter wave displacements, but reaction times for marine robots are slow relative to the changing wave forces. Given the periodicity of waves, feedforward techniques should be explored. This thesis outlines how Model Predictive Control (MPC) can reduce an underwater robot's position error when station-keeping under the influence of ocean waves. A wave field is decomposed to component velocities and used as input to the model. Using this model, an optimized control input is calculated over the desired time horizon. This optimized control is shown to resist wave displacement by using thruster force to counter impending disturbances.

The main novelty presented in this thesis is a robotic control method that can estimate and compensate for future wave action. This is demonstrated through simulations of an underwater robot performing station-keeping above an intermediate water bathymetry under the influence of a strong sea swell. The calculated control actions are optimized to actuate an underwater robot's thrusters in an anticipatory fashion so that the vehicle remains nearly stationary as the waves pass over it. This provides an increase in the quality of robotic observations in shallow ocean water, as well as reducing the risk of equipment damage while deployed. MPC reduces an underwater robot's position error in a station-keeping application when compared to traditional feedback control. Additionally, the algorithm is shown to be resistant to sensor noise of the observed wave field. Thus, the goals of this thesis are to:

- Determine the most effective prediction horizon for MPC in ocean waves
- Measure performance improvement from MPC over PD control
- Measure MPC resistance to noisy wave field observations

The remainder of this thesis is organized as follows: the problem background and some related work is highlighted Chapter 2. Next, the MPC method is described and outlined in Chapter 3, which includes the system model and algorithm specifics. Chapter 4 presents results of determining effective prediction horizons and analyzes system performance against a simulated wave field with noisy observations. Finally, concluding remarks are provided in Chapter 5.

Chapter 2: Background

This chapter is organized as follows: an introduction to wave energy and extraction methods is given in Section 2.1. Section 2.2 gives background information on the linear wave mechanics used to model the simulated wave field. Section 2.3 introduces Model Predictive Control (MPC) and related work on the topic. Lastly, Section 2.4 serves as a literature review, highlighting marine robotic advancements in path planning (Sec. 2.4.1), localization (Sec. 2.4.2), perception, (Sec. 2.4.3), and applied systems (Sec. 2.4.4).

2.1 Wave Energy

The world's oceans can produce close to 2 TW – roughly twice the current global usage – of usable wave energy [17]. Compared to solar and wind sources, wave energy is relatively predictable and available on a consistent basis. Some challenges facing wave energy extraction are: poor scaled economics, a high rate of infrastructure wear, and unclear effects to the coastal geomorphology [21]. All of these are active areas of research in the field.

According to Linear Wave Theory (LWT), the energy (E_L) in one water wavelength (L) is the sum of its potential and kinetic energies [14]. After some deriva-

tion, it is reduced to:

$$E_L = 1/8\rho gH^2L. \quad (2.1)$$

Though simplified, (2.1) illustrates a noteworthy point that neither the average potential nor kinetic energy per unit area depends on water depth, but instead is simply proportional to the squared wave height term (H). The rate at which energy is transferred is the energy flux. For LWT, it is the rate at which work is being done by change in energy density of a fluid over a vertical face [14].

Wave energy conversion as a whole is a young industry with many competing converter designs. Point absorbers are small devices that freely oscillate at the water surface, expanding and contracting some working fluid. An attenuator is a jointed body that floats parallel to the wave direction, generating energy from the relative motion at the joint. An Oscillating Water Column (OWC) device uses differences in atmospheric pressure to force trapped air through a turbine as it is forced out by wave action [17]. The arrays to be deployed at NNMREC test sites are yet to be determined; however, they will all require similar mooring and anchor systems. Thus, immediate robotic testing for ALFA is within reason.

Robotic maintenance of Wave Energy Converters (WEC), such as the platform detailed in [28], is an active area research. In fact, MPC techniques have been explored by the wave energy community as a way to optimize WEC power generation. As shown in [8] and [43], MPC can incorporate actuator limits and system constraints to provide optimal energy capture while benefiting from a variable prediction horizon. Wave prediction modeling is also an active area of research. In

[32], Ling provides a method of real time WEC force estimation, showing accurate predictions for horizons up to 15 s. This method is noteworthy since it does not require a network of sensors to provide wave information, as is often carried out. In [13], Colby uses an artificial neural network to estimate wave forces from a hydrodynamic model as inputs to an evolutionary algorithm to optimize WEC geometry.

2.2 Wave Mechanics

According to LWT, a wave field in a random sea is composed of a superposition of sinusoids. Once decomposed, each sinusoid can then be analyzed as a single monochromatic wave with unique period (T), amplitude (a), and phase offset (ϕ) [39, 14]. For reference, this thesis instead uses the term wave height (H), which is simply twice the amplitude ($H = 2a$). LWT assumes fluid flow is irrotational, incompressible, and inviscid, thus allowing for potential flow [7]. In practice, LWT is not only easily implemented, but has also been shown to produce accurate results [14]. Thus, LWT is often employed for a reliable primary analysis before other nonlinear theories, and it forms the wave action model used in this thesis.

Using LWT, a field of random sea waves, not unlike an electrical signal, can be decomposed to its component frequency spectra by way of a Discrete Fourier Transformation (DFT) [21]. DFT solvers will output frequency coefficients with associated energies correlating to signal amplitude. The phase information can be extracted by taking the *arctan* of the imaginary over the real part of each term.

For further details on Fourier spectra of wave fields, see [17].

In this thesis, the input wave field is assumed to be provided to the vehicle as an array of wave period, wave height, and phase offset (T, H, ϕ) . Component wavelengths (L) , wave numbers (k) , and frequencies (ω) are solved for by way of the dispersion relation [14]. Applying superposition, the wave field time series is then constructed by a water surface wave equation (3.10).

Beneath each component monochromatic wave, wave-induced particle displacements occur in a cyclical fashion. In deep water waves where depth is greater than approximately half of the wavelength ($d > L/2$), these displacements follow circular paths. As d decreases, the paths become more elliptical until they are nearly horizontal in the surf zone [14]. Associated particle velocities and accelerations can be derived from these trajectories. As stated earlier, the magnitude of these velocities and accelerations decay exponentially through the water column such that they are negligible ($< 0.04\%$) at a depth, $d \approx L/2$ [14]. The per-wave, at-depth solutions to these velocity and acceleration equations are used as the primary inputs to the modeled system dynamics.

2.3 Feedforward Control

Robotic control is an area of research under constant development. For trivial tasks, traditional feedback (PID, or PD) control remains widespread because of its ease of implementation and low computational cost. In applications requiring more precision; however, more advanced learning-based control methods have been

developed [30]. For the scope of this paper, it is assumed that the basics of feedback control are well understood and are not the focus of this background section.

Feedforward control describes a control method where a predefined command signal is passed to an actuated plant from an external source [9]. This differs from feedback control, as feedforward does not inherently adjust the command to account for its effect on the system output [37]. Model Predictive Control (MPC) seeks to synthesize the two by combining the forward estimation ability of a well-defined model with the error checking of feedback control.

MPC uses a dynamic model to forecast system behavior and optimizes that forecast to produce the best decision [42]. The robustness of this model is the centerpiece of MPC, as it must allow for accurate estimations of future states along a prediction horizon. MPC then can optimize an action by way of minimizing some global cost function.

MPC gained popularity in the late 1970s, particularly in chemical process industries, because of an impulse function which required less *a priori* information [9]. This results in little or no hand tuning of controller gains. MPC in state space form has also been explored. This is beneficial and allows for well known state space techniques to be extended to multivariable processes and systems with stochastic noise in measured variables [9]. In this thesis, model dynamics are developed in state space to take advantage of these formations.

Applying MPC for underwater robotics is a promising option, as the combination of model dynamics and cost function minimization requires minimal tuning of controller gains. In [27], energy efficient paths for a glider are generated by min-

imizing costs across stratified, spatially-distributed currents using an A* search heuristic, or an overshooting estimate. In [33], MPC with a least squares cost function is used to optimize sawtooth paths for an underwater glider with an emphasis on real-time execution. This cost minimization technique is similar to that used in this thesis. Both approaches show the value in state estimation for robotic path planning, but do not incorporate wave disturbances.

2.4 Underwater Robotics

Underwater vehicles can be classified into one of two generic categories: manned and unmanned vehicles. Unmanned Underwater Vehicles (UUV) are often labeled as synonymous with Autonomous Underwater Vehicles (AUV). For the scope of this thesis, the term AUV will be used for an untethered unmanned vehicle. The term Remotely Operated Vehicle (ROV) will be used to describe a tethered unmanned vehicle whose operation may or may not be teleoperated [12]. No manned vehicles will be discussed.

The term “glider” may on occasion be used to describe a type of AUV. Gliders such as the Slocum shown in Fig. 2.1 are designed to move efficiently through the water column by changing their weight [2]. Successive pitch adjustments up and down result in a sawtooth subsurface flight profile with no external propulsion.



Figure 2.1: A Webb Research Slocum Glider. This AUV uses only a buoyancy differential and small attitude adjustments to achieve an ultra low-wattage (0.5-1.0 Watts) and long range mission profile.

2.4.1 Path Planning

One of the more active areas of research in Field Robotics is optimal path planning. Planning methods combine discretized graphs, heuristic search, and dynamic programming which, when properly balanced, give close to optimal results at a reduced computational cost [30]. Underwater robotics is no different, albeit with its own unique challenges. One such challenge is vehicle localization in an environment absent GPS and long range wireless transmissivity. Relative measurements can be provided by onboard sensors and corrected for uncertainty. The method

employed by Galceran et al. [19] uses a 2-phase approach to produce a sonar generated bathymetric map. In it, a standard mow-the-lawn pattern is employed using an *a priori* input to distinguish the seafloor into regions of low and high slope with a user-influenced gradient. Once the map is split, the planar regions are covered exhaustively with a path generated in a Travelling Salesman (TSP) manner while high-sloped regions are ignored as obstacles. After, the high-slope regions can then be covered with a slicing algorithm, where the vehicle travels in a spiral fashion along greedily linked adjacency points. Localization is provided by an onboard Doppler Velocity Log (DVL) and Inertial Measurement Unit (IMU) while submerged, and GPS while at the surface. The vehicle path is adaptively replanned for uncertainty using a Stochastic Trajectory Optimization Motion Planning (STOMP) method to account for errors from its ground true position.

Some sampling techniques are analysed in [35] and compared with a prioritized cost-evaluation function. This function balanced three mission metrics: collected samples, energy consumption, and mission duration. Their results showed that in almost all sampling scenarios, a stratified random spiral pattern is the most effective sampling method. This method is similar to that employed by the GIRONA 500 AUV in [19] to measure high-slope areas. In [5], an informed path planning method is applied to a Slocum Glider, where the vehicle can maximize user-weighted information gain while avoiding high-traffic areas during specific time windows, traveling there only at night. The recursive algorithm uses a “diminishing returns” approach, rewarding travel to unexplored nodes. This work is expanded upon in [6] with a Branch and Bound (B&B) search algorithm which

chooses the best path for the glider based on a probabilistic model of measurement quality expected. This model is a generalization of the Gaussian probability distribution of [41] of whether a point in space adds to the “informativeness” of the scalar field being sampled. The specific goal of the algorithm is to find the path which minimizes the average variance reduction in the Gaussian Process (GP) model. The results show that though exhaustive search is optimal in an infinite horizon, this is not computationally efficient. By limiting the horizon and using B&B to keep track of these searches, costs are cut dramatically. The authors also suggest using a heuristic to increase the lower bound more quickly.

Another interesting path-planning solution for a Slocum Glider is presented by Smith et al. in [48]. Here, the unique glider dynamics are considered in creating an algorithm which will traverse areas of high interest, adjust its sampling density accordingly, and avoid areas of strong and/or variable currents. Unlike other underwater vehicles, a glider performs poorly when running a traditional mow-the-lawn pattern due to poor navigational accuracy and reliance on dead reckoning. Instead, a glider “flies” in a zigzag pattern, one where its sampling density is directly related to its pitch angle. In this approach, an expert-imported interest domain and water current data from a Regional Ocean Modeling System (ROMS) feed are input to the algorithm which is subjected to field trials. The most usable results were those of a trial in the Southern California Bight, where the algorithm scored 20% better than a precomputed trajectory. Simulator results on an A* derivative are carried out by Fernández-Perdomo et al. in [18] show that accounting for ROMS-generated ocean currents in path generation leads to reduced overall

travel times.

Path optimization is a pivotal aspect of robotic path planning. In general, the amount of information gain should be maximized while considering some cost function, often times the mission duration. This approach is shown in [49] where a sampling path is designed to track a specific oceanographic point of interest and update accordingly. Expanding upon this in [47], Smith et al. applied an Ecomapper AUV to autonomously track and move along Oceanographic Fronts. These fronts are gradients in the ocean water properties and are directly affiliated with plankton populations. The algorithm employed uses a prior estimation of the front locations and then aims to sample the front by repeatedly crossing the AUV through it. New waypoints are generated as the front status is ascertained and compared to GPS readings. The next waypoint is generated to either cross the front again or continue searching, whichever is more efficient. The results showed that preplanned routes were shorter than the adapted routes. This is because the algorithm tended to work conservatively, using a cubic spline which overestimated the front curvature. Otherwise, the vehicle moved well along waypoint contours.

Another efficient path planning approach is the hybrid Fast Marching (FM) method, or FM*, employed in [40]. This approach uses an A* search heuristic to find a continuous path through a discretized, static world, converting grid meshes to save on computation time. In addition, the algorithm accounts for the vehicle kinematics and adapts the trajectory for underwater currents. The simulator results are promising but presented without any field trials, listing that as an avenue for future work.

2.4.2 Localization

A central focus in greater robotics is the Simultaneous Localization And Mapping (SLAM) problem. This arises when a robot either does not have access to a map of its environment or it cannot determine its position in an *a priori* map [51]. SLAM techniques attempt to generate a map of the environment while simultaneously localizing a robot within it. It is a difficult and costly problem to solve as maps and relative positions must be estimated throughout. SLAM challenges in a marine environment are amplified as robots rarely remain stationary for very long and sensor observations are often noisy.

Efficient area coverage and good SLAM performance in navigation; however, are conflicting objectives. To provide efficient coverage, redundant overlapping trajectories need to be minimized while still accounting for vehicle/sensor “drift” over time. In [29], Kim and Eustice introduce an active visual SLAM method of perception-driven navigation which balances exploration and revisitation using a reward function. They used a visual saliency method to compute image scores and schedule revisits accordingly. The saliency score assumes that visual SLAM treats images unequally, a fair assumption especially underwater, where the spatial distribution of desired features is not consistently apparent. As a set of waypoints is generated, a reward for that path is computed based on its overall saliency. The reward is then compared with the cost of travel and the robot then moves.

Results for [29] are presented in a simulator and through HAUV field trials in a ship hull inspection role for the *SS Curtiss*. The AUV plans paths autonomously

and then revisits waypoints to close loops and minimize navigation uncertainty by comparing features at its “drifted” position with prior features. This uncertainty horizon controls how opportune revisits are and directly affects the strategy cost. This work is expanded upon in [11] where the loop-closing navigation uncertainty is probabilistically modeled using GP regression. This approach combines sampling-based plans with information filters to quickly search many paths given a utility function. Trials were again carried out on *SS Curtiss* and results were compared to [29]. The proposed method showed an improved path length with uncertainty levels similar to the “best possible” deterministic model.

Active SLAM as presented in [29] and [11] is effective in a hull inspection application as it can use previously recorded features to adjust for uncertainty. In a bathymetric mapping application; however, this is not a preferred approach as the seafloor is often largely featureless. In [4], Barkby et al. employ an efficient and featureless bathymetric SLAM with a Rao-Blackwell particle filter and GP Regression for depth uncertainty loop closures. This method overcomes feature dependency by generating a particle-based 2D depth map from successive identical point clouds of bathymetric observations. To save memory, the entire point cloud is not saved; the backtraceable trajectory of each child point is recorded to a parent map in a process called Distributed Particle Mapping (DPM). Each observed state is removed from the particle filter and tracked by a single shared extended Kalman filter. A bathymetric map can be input or initialized to zero. The algorithm was tested on a Sirius AUV, a JASON ROV, and a Sentry AUV at different locations, showing consistently small particle set sizes and generating similar maps to Ultra

Short Base Line (USBL) data at a minimum computation time.



Figure 2.2: A Bluefin Hovering Autonomous Underwater Vehicle, or HAUV, as referenced in [29] and [26]. This vehicle was used along with a Didson sonar to reconstruct the *S.S. Curtiss* hull draft.

2.4.3 Perception

Another active area of research in Field Robotics is interpreting real-world observations. Underwater environments increase the difficulty of robotic perception due to their unique characteristics. A plethora of methods on this topic have been published, and each carries its own applied merits and detriments. In [38], Papadopoulos et al. use a SCOUT Autonomous Surface Vehicle (ASV) with off-the

shelf sensors to obtain reconstructions for partially-submerged structures in rough sea conditions. The ASV used a LiDAR Camera to collect data above the water line, where the vehicle travelled around the target in a circle, recording data in a point cloud. This data is then passed through an Iterative Closest Point (ICP) minimizing algorithm and then filtered to reconstruct the observed surface. Additionally, the vehicle used a sonar emitter to collect data below water. The results, although not novel in their approach, are still interesting in their low cost execution. Another interesting takeaway was that this method did not require a pre-computed trajectory, like SLAM techniques. This allowed the vehicle to reconstruct slow-moving structures and even watercraft.

In addition to presenting the aforementioned path-planning techniques in [19], the authors also detail their surface reconstruction method. In it, the sonar sensor populates a triangular mesh grid with points and normal obtained as gradient samples of a minimized volumetric indicator function. When minimized, the surface can then be extracted as a zero level set using surface contours. The normals are estimated using a Hoppe Method, which fits a plane in a local k -neighborhood around each point. Another interesting approach is outlined in [15], where the authors attempt to use a Starbug AUV as a “mule” to transfer data from non-communicating submerged sensor nodes. A path is plotted from node to node using a TSP solver with uncertainty and the nodes are found using computer vision. This approach looks for a specific color range and maintains the object camera center while data is up/downloaded.

Lighting issues, camera contrast, and blurring all hinder camera operation in

an underwater application. By needing only non-orthogonal captured raw camera point sets, the method devised by Campos et al. in [10] is widely applicable. The process employs Restricted Delaunay Triangulation (RDT) meshing which takes a small set of points and iteratively constructs a coarse-to-fine triangle surface. This is done online and is shown to be able to process corrupted point sets with loose input requirements and a low memory footprint. RDT selects the points which lie in a local neighborhood and logarithmically filters out any outliers. Any intersections between local segments is answered and generates a finalized Local Bivariate Quadric (LBQ) surface. Results are presented on a number of different multi-beam sonar experiments on sources with varying complexities and added noise sets. The method is also experimented on an optical stereo multi-view seafloor reconstruction. Though the algorithm struggled with vertical walls at shallow observation heights, the post-processed results were still robust despite a high number of outliers.

Interpreting the novelty of each observation is equally important as making the actual observation. In [20], Girdhar et al. use an online topic modeling approach to compute the “surprise” score of an observation. Applied, a vehicle could explore an environment as a tourist might explore a new city: stop and observe if new, move along if not. Using this method, an underwater robot was able to recognize and record different species of coral while having its speed controlled by mapping this “surprise” score through a sigmoid function.

2.4.4 Combined Systems

By combining these path-planning and surface reconstruction techniques, a number of useful applications similar to the bathymetric mapping in [19] can then be performed. Another such application is modeling and inspecting the hull of a berthed ship as detailed in [24], [25], and [26]. Hollinger et al. seek to construct closed 3D meshes from sonar-induced point clouds and measure the uncertainty on the closed mesh using non-parametric Bayesian regression. Specifically, the method applies a GP implicit surface with augmented input vectors for uncertainty and use a probabilistic path planner to minimize uncertainty while maximizing the mesh coverage. Here, the surface uncertainty is a generalization of the probability distribution of whether a point in space is in fact on the mesh surface. The input vectors can be supplied in various ways; in this thesis, they are supplied by way of an initial coarse survey over the ship hull. [26] uses a similar Poisson surface reconstruction method for sonar imaging and 3D mesh modeling as [24]. The path planning is slightly different; however, as it uses a computationally inexpensive TSP/RRT solver to create a probabilistically complete and asymptotically optimal solution.

In [23], Hollinger and Sukhatme apply their prior work in [24] and [25] to a feature mapping role. Here, a YSI EcoMapper AUV is used to dive Puddingstone Lake, CA and generate dense bathymetric maps using a sidescan sonar. Their approach again models uncertainty in the form of a GP, where the actively planned path seeks to maximally reduce the variance while accounting for onboard budget

constraints. Paths are then selected greedily. In addition, they allow for dives to be adaptively replanned as more information becomes available to the vehicle. These re-planned dives are computed offboard and communicated to the vehicle after a human operator checks for safety. Results show that their planned path shows substantial improvement over standard lawnmower patterns and show limited improvement when adaptively re-planning, showing an 8% reduction in uncertainty over the first re-planning cycle and no improvement over the second re-planning cycle. They cite reducing the computational cost of adaptivity as an avenue of future work.



Figure 2.3: A YSI EcoMapper AUV as referenced in [47] and [23] which along with a side-scanning sonar was used to construct dense bathymetric maps of the floor of Puddingstone Lake, CA.

Ocean currents often lead to untethered AUV's such as the Slocum suffering from increasing positional uncertainty over their mission profiles. If the vehicle is equipped with a DVL and the water depth is shallow enough ($d \lesssim 200$ m), then localization becomes significantly more trivial. In the mid-water column, which is

the submerged space outside the DVL bottom lock range, the vehicle relies solely on dead reckoning to localize itself. In [34], Medagoda et al. propose a novel method of using a vehicle’s DVL as an Acoustic Doppler Current Profiler (ADCP) to instead measure water current relative to the vehicle. Applied, this can constrain the positional error to the initial current velocity uncertainty at the sea surface. In addition, if bottom lock is achieved at any point, the velocity history is constrained to 2σ by fusing the dead reckon and current estimates. The method is validated using field data from a Sentry AUV, where DVL and USBL inputs to the AUV are blacked out while the vehicle performs a series of lawnmower patterns. Results show that with no prior info about the true water currents, positional errors with the USBL are within 600 m at 17 km (3.5 %) for 8 hours and 8 km at 84 km (9.5 %) at 20 hours.

Onboard vehicle sensors are often used to compensate for accumulated position error. Another active avenue for autonomy in marine robotics; however, is the use of *in situ* data to allow for active replanning and to guide sampling decisions. A model is described in [46] which uses a Wave Glider AUV. Its goal is to learn a linear predictive regression model which inputs environment data to predict glider speed. The appeal is to use this data for better offline objective planning. Their results concluded that the dominant contributors to glider speed predictions were the significant wave height and peak wave period. This is not surprising, as they are the dominant contributors to orbital velocities in the surf zone [14]. Their work is expanded upon in [36] using a GP-based approach to mixed results.

This approach is expanded upon in [22] where Hitz et al. attempt to use the

onboard sensor data as the adaptive path planned threshold criteria. They employ a receding horizon path planner to reduce the uncertainty around this specific criteria, which generates optimal sampling paths connecting highlighted sites along a user-defined vertical transect plane. The uncertainty model is GP-derived and uses no *a priori* information on initialization. They provide simulator and field results on an ASV tracking toxic cyanobacteria. The vehicle traveled 18 km and showed reduced uncertainties of 68 % when compared to non-adaptive techniques.

Though there is no ROV involved, the multi-robot approach in [16] proposes a novel solution to the ship-hull inspection problem. The three robots, a small quadcopter and two magnetic crawlers, employ a three phase approach. First, the Pelican Quadrotor gives a top level scan of the interior of a cargo bay using computer vision to find the more compromised areas. Second, a lightweight magnetic robot crawls along the walls using a Monte Carlo localization method and an onboard camera to inspect the coating corrosion and any cracks in the hull. Finally, the heavier Magnetic Autonomous Robotic Crawler (MARC) crawls along the walls with a similar localization method but uses an ultrasound sensor to measure any irregularities in the material thickness of the vessel. The work is preliminary and many locomotion, point cloud, and multi-agent improvements are suggested. Otherwise, it is an interesting solution, especially if paired with an outer hull-inspecting ROV.

A novel multi-robot approach is presented in [31] where a group of ten Slocum and Spray gliders are used to autonomously sample an area for oceanographic data. The different gliders have different sampling profiles; the Spray gliders patrol the

outside loop of a rectangle while the Slocum gliders make three loops through the inside of it. Paths are generated to maintain equal intervehicle spacing within each loop, with gliders adaptively shortening or expanding as necessary. The algorithm also can adapt for an adjusted number of gliders within each loop. Results show that the system provided excellent coverage of the area while maintaining relatively good vehicle spacing, which further improved the sampling coverage. Strong currents were found to be an issue but one that the system adapted for reasonably well.

Wave-induced station-keeping is explored in [45] and [44] where a dynamic Kalman filter is created to synthesize onboard sensor data to provide an estimate of wave-induced disturbances. This filter fuses ground speed from the Doppler Velocity Log (DVL), relative water speed from the an ADCP, and attitude from an IMU among other sensors to produce an estimate of the fluid velocity. This method highlights some of the difficulties of *in situ* wave parameters. One drawback of their experiments was the need for a subsurface sensor network to provide wave parameters to the controller. The work in this thesis seeks to improve on these results by providing improved cost minimization techniques and emphasizing real-time control execution.

Chapter 3: Method

This chapter is organized as follows: Model Predictive Control (MPC) is further detailed in Section 3.1, where the global cost function and minimization technique is explained. Section 3.2 gives background information on the robot modeled in this work and explains the force balance equations. Section 3.3 expands these equations into State Space form, readable by the simulator. Next, Section 3.4 details how the simulator constructs a wave field and how wave forces are extracted and input to the state estimator. Lastly, Section 3.5 synthesizes the cost function and wave field model into the MPC algorithm.

3.1 Model Predictive Control

The term Model Predictive Control (MPC) does not refer to a specific strategy, but a variety of methods that all in some way incorporate the following [9]:

- Modeled state estimator along some time horizon
- Cost function to minimize a control sequence
- Receding horizon as optimized control is carried out

MPC requires a model of the system dynamics that can estimate future states from a current state and set of control inputs along a time horizon. The model in

this case includes the vehicle dynamics and thruster forces, along with a disturbance matrix to model the wave disturbances on the system. By thresholding the thruster forces within the minimization function, the need for tuning gains, such as in PID control, is removed [42]; however, the number of steps in the prediction horizon must be judiciously chosen. Too little time will not allow for the full dynamics of the system to be accounted for while too much time will be computationally intensive.

The optimization objective is to find the sequence of input control actions to the state estimator that minimizes some global cost function. Cost functions can be formulated by balancing one or several metrics, e.g. mission duration, energy consumption, or number of sampled observations [30, 5]. Given the station-keeping objective, the cost function employed in this work is the sum of squared distances between the desired and predicted states added to its input control over the current horizon, or:

$$J = \sum_{k=1}^N [\boldsymbol{\Upsilon}_{target} - \boldsymbol{\Upsilon}_k(\mathbf{u}_k)]^2 + \mathbf{u}_k^2, \quad (3.1)$$

and

$$\mathbf{u}_{1:N}^* = \arg \min_{\mathbf{u}_{1:N}} J(\mathbf{u}_{1:N}), \quad (3.2)$$

where N is horizon length, k is the current time step in the N horizon, $\boldsymbol{\Upsilon}_{target}$ is the desired state, $\boldsymbol{\Upsilon}_k(u_k)$ is the state at time k from input \mathbf{u}_k , and $\mathbf{u}_{1:N}^*$ is the optimized control input that minimizes the cost, J .

The control action is then optimized by evaluating the Jacobian, which is the derivative of the cost with respect to the control action. The Jacobian is minimized

as the optimal control action is approached. At each optimization step, a new set of control actions is generated by perturbing the previous set according to the Jacobian. The new control action effects are then estimated along the horizon. This is repeated until a set of control actions that minimizes the cost function is calculated as in (3.2). This gradient descent method of cost evaluation and optimization is similar to that used by Medagoda et al. in [33].

3.2 Vehicle Model

For the scope of this thesis, the term Remotely Operated Vehicle (ROV) will be used to describe a tethered unmanned vehicle whose operation may or may not be teleoperated [12]. The ROV modeled for this work is the SeaBotix vLBV300 shown in Fig. 3.1. In simulation, the ROV is not teleoperated and performs control actions autonomously. In addition, the terms “ROV”, “vehicle”, and “robot” are used interchangeably.

The ROV has six vectored thrusters controlling motion along five degrees of freedom. For the scope of this work, only the surge and heave directions, or forward (global \mathbf{x}) and vertical (global \mathbf{z}) axes, are considered. See Fig. 3.1 for reference.

The vehicle is assumed to be a rigid body, and irrotational in the water flow. Since its size is well below the wavelength, its drift motion is assumed to follow that of a particle. Vehicle dimensions, mass parameters, and thruster forces are provided in Table 3.1. Moment of inertia and center of gravity data was provided by the CAD model shown in Fig. 3.2 input to Dassault Systèmes SolidWorks [50].

Table 3.1: Vehicle Parameters used in Simulation

Parameter	Symbol	Value
Density of Seawater	ρ_{sea}	1030 kg/m^3
Incident Area, \mathbf{x}	$A_{i,\mathbf{x}}$	0.156 m^2
Incident Area, \mathbf{z}	$A_{i,\mathbf{z}}$	0.273 m^2
Moment of Inertia, \mathbf{x}	I_{xx}	0.62 $kg\ m^2$
Moment of Inertia, \mathbf{z}	I_{zz}	1.60 $kg\ m^2$
Dry Mass	m_{dry}	22.2 kg
Added Mass, \mathbf{x}	$m_{add,x}$	8.1 kg
Added Mass, \mathbf{z}	$m_{add,z}$	36.7 kg
Drag Coefficient, \mathbf{x}	$c_{d,x}$	0.84
Drag Coefficient, \mathbf{z}	$c_{d,z}$	1.06
Max Thruster Force	T_{max}	29.7 N
Thruster Angle, Forward	θ_f	35°
Thruster Angle, Aft	θ_a	45°
Thruster Angle, Vertical	θ_v	20°



Figure 3.1: A SeaBotix vLBV300 ROV similar to the one modeled in this work. Along with IMU and DVL sensors, the vehicle has six angled thrusters which control it along five degrees of freedom: heave (vertical z), sway (lateral y), surge (lateral x), roll (rotation around x), and yaw (rotation around z).

Added mass values were provided by the same model input to Ansys AQWA [1]. Drag coefficient data was provided by the manufacturer.

The differential equation defining ROV motion is:

$$\mathbf{M}\dot{\mathbf{v}}_a = \mathbf{F}_{thrust} + \mathbf{F}_d + \mathbf{F}_g + \mathbf{F}_c, \quad (3.3)$$

where \mathbf{M} is a mass term containing the dry and added masses, m_{dry} and m_{add} , \mathbf{v}_a is the absolute velocity of the vehicle with respect to the inertial reference frame, \mathbf{F}_{thrust} is the ROV thruster force, \mathbf{F}_d is the drag force from the water, \mathbf{F}_g is the force of gravity, and \mathbf{F}_c is the Coriolis force. By assuming that the Coriolis force

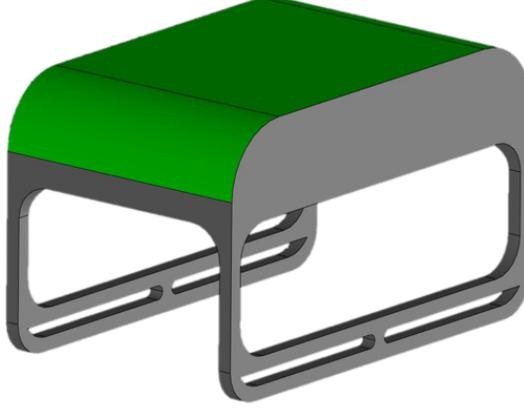


Figure 3.2: A CAD model of the robot used to generate mass and hydrodynamic properties from SolidWorks and AQWA, respectively.

is negligible and that the vehicle is neutrally buoyant, the \mathbf{F}_c and \mathbf{F}_g terms are neglected. Substituting inertia and drag relations, (3.3) becomes:

$$m_{dry}\dot{\mathbf{v}}_a + m_{add}\dot{\mathbf{v}}_r = \mathbf{F}_{thrust} + \frac{1}{2}\rho_{sea}A_iC_d|\mathbf{v}_r|\mathbf{v}_r, \quad (3.4)$$

where \mathbf{v}_r is the velocity of the vehicle relative to the water, which acts on the added mass term of \mathbf{M} in (3.3). The added mass term is used to measure the inertia of the volume of water displaced by the vehicle. Substituting heave, surge, and particle velocity components, (3.4) becomes:

$$\begin{bmatrix} m_{dry} + m_{add,x} \\ m_{dry} + m_{add,z} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{thrust,x} \\ \mathbf{F}_{thrust,z} \end{bmatrix} + \begin{bmatrix} m_{add,x} \\ m_{add,z} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_{p,x} \\ \dot{\mathbf{v}}_{p,z} \end{bmatrix} + \frac{\rho_{sea}}{2} \begin{bmatrix} A_{i,x}C_{d,x} \\ A_{i,z}C_{d,z} \end{bmatrix} \begin{bmatrix} |\dot{\mathbf{x}} - \mathbf{v}_{p,x}|(\dot{\mathbf{x}} - \mathbf{v}_{p,x}) \\ |\dot{\mathbf{z}} - \mathbf{v}_{p,z}|(\dot{\mathbf{z}} - \mathbf{v}_{p,z}) \end{bmatrix}, \quad (3.5)$$

where \mathbf{v}_p is the particle velocity at a time and position beneath a monochromatic wave. For added reference, $\mathbf{v}_a = \mathbf{v}_r + \mathbf{v}_p$, and $\dot{\mathbf{v}}_{a,x} = \ddot{\mathbf{x}}$, $\dot{\mathbf{v}}_{a,z} = \ddot{\mathbf{z}}$.

3.3 State Space Form

The differential equations are now rewritten in state space form solvable by the simulator. This takes the form:

$$\dot{\Upsilon} = \begin{bmatrix} \dot{\mathbf{x}} & \ddot{\mathbf{x}} & \dot{\mathbf{z}} & \ddot{\mathbf{z}} \end{bmatrix}^T = \mathbf{A}\Upsilon + \mathbf{B}\mathbf{u} + \mathbf{D}, \quad (3.6)$$

where

$$\mathbf{A}\Upsilon = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \\ \mathbf{z} \\ \dot{\mathbf{z}} \end{bmatrix}, \quad (3.7)$$

$$\mathbf{B}\mathbf{u} = \frac{T_{max}}{m_{dry}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \cos\theta_f & \cos\theta_f & -\cos\theta_a & -\cos\theta_a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\cos\theta_v & -\cos\theta_v \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix}, \quad (3.8)$$

and

$$\mathbf{D} = \begin{bmatrix} 0 \\ \frac{\dot{\mathbf{v}}_{p,\mathbf{x}}}{m_{dry}} + \frac{\rho_{sea}A_{i,\mathbf{x}}C_{d,\mathbf{x}}}{2(m_{dry}+m_{add,\mathbf{x}})}|\dot{\mathbf{x}} - \mathbf{v}_{p,\mathbf{x}}|(\dot{\mathbf{x}} - \mathbf{v}_{p,\mathbf{x}}) \\ 0 \\ \frac{\dot{\mathbf{v}}_{p,\mathbf{z}}}{m_{dry}} + \frac{\rho_{sea}A_{i,\mathbf{z}}C_{d,\mathbf{z}}}{2(m_{dry}+m_{add,\mathbf{z}})}|\dot{\mathbf{z}} - \mathbf{v}_{p,\mathbf{z}}|(\dot{\mathbf{z}} - \mathbf{v}_{p,\mathbf{z}}) \end{bmatrix}. \quad (3.9)$$

The \mathbf{u} vector in (3.8) refers to motor inputs of each of the vehicle’s six vectored thrusters: u_1 and u_2 are the forward thrusters, u_3 and u_4 are the aft thrusters, and u_5 and u_6 are the vertical thrusters.

3.4 Wave Field Model

The bathymetry and wave climate selected for analysis is similar to that of the National Northwest Marine Renewable Energy Center (NNMREC) North Energy Test Site (NETS) in the coastal Pacific Ocean two nautical miles out of Newport, Oregon [32]. The operational depth at NETS is approximately 50 m. The wave field used in this work was verified using AWAC acoustic measurement data deployed at NETS from August to October 2013. Each 40 min time series was recorded at 2 Hz. Figure 3.3 provides three sample excerpts of wave data from that time. The times chosen are during more elevated sea states in the early autumn when stronger sea swells are expected as the oceanographic climate transitions into winter.

The wave field used by the simulator is the set of eight different periods, heights, and phases (T, H, ϕ) detailed in Table 3.2. It was constructed to appear qualitatively similar to a real wave field at NETS during the winter transition. The superimposed wave field shown in Fig. 3.4 is constructed by the water surface wave equation shown below:

$$\eta(\mathbf{x}, \mathbf{t}) = \sum \frac{H}{2} \cos(k\mathbf{x} - \omega\mathbf{t} + \phi). \quad (3.10)$$

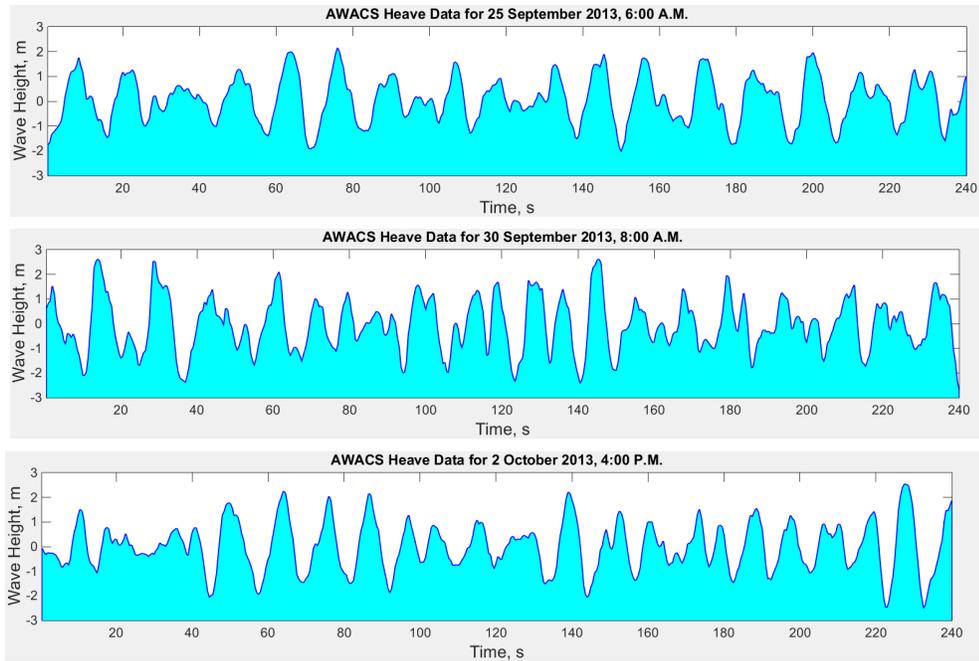


Figure 3.3: Four second sample sets of AWACS acoustic heave data from NETS, 2013. The simulated wave field is built with similar wave heights and gradients.

Component wavelengths (L), wave numbers (k), and frequencies (ω) are solved for by way of the dispersion relation [14] below:

$$\omega^2 = gk \tanh(kd), \quad (3.11)$$

where g is the acceleration due to gravity, d is the water depth, $k = 2\pi/L$, and

$$L = \frac{gT^2}{2\pi} \left[(\tanh \frac{\omega^2 d}{g})^{3/4} \right]^{2/3}. \quad (3.12)$$

Because of the 2-dimensional analysis along only surge and heave motions, wave angles are set to zero [21].

Table 3.2: Wave Field Parameters used in Simulation

Component Wave	1	2	3	4	5	6	7	8
Wave Period, T , s	10	8	12	11	6	7	9	25
Wave Height, H , m	1.8	0.9	1.6	1.3	0.4	0.5	1.1	0.7
Phase, ϕ , rad	$-\frac{\pi}{2}$	$-\frac{\pi}{4}$	$-\frac{5\pi}{8}$	$\frac{4\pi}{13}$	$-\frac{\pi}{15}$	$\frac{\pi}{3}$	$-\frac{\pi}{18}$	$-\frac{7\pi}{4}$

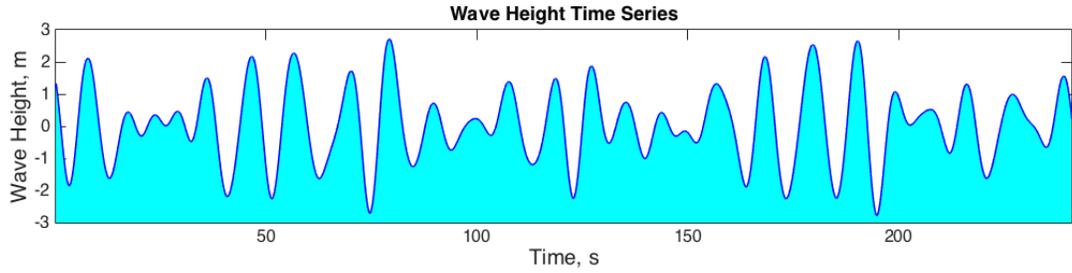


Figure 3.4: A time series of the wave profile over the 240 s simulated time window constructed using the parameters listed in Table 3.2.

Prior to any forward state estimation or any simulated vehicle motion, the wave action, or particle velocities and accelerations (\mathbf{v}_p and $\dot{\mathbf{v}}_p$), are calculated for the robot's current position in time and space. For the lateral (\mathbf{x}) case in intermediate water this is:

$$\mathbf{v}_{p,\mathbf{x}} = \frac{HgT}{2L} \frac{\cosh \frac{2\pi(\mathbf{z}+d)}{L}}{\cosh \frac{2\pi d}{L}} \cos(k\mathbf{x} - \omega\mathbf{t} + \phi), \quad (3.13)$$

and its derivative is:

$$\dot{\mathbf{v}}_{p,\mathbf{x}} = \frac{g\pi H}{L} \frac{\cosh \frac{2\pi(\mathbf{z}+d)}{L}}{\cosh \frac{2\pi d}{L}} \sin(k\mathbf{x} - \omega\mathbf{t} + \phi). \quad (3.14)$$

The particle velocity for the vertical (\mathbf{z}) direction in intermediate water depth is:

$$\mathbf{v}_{p,\mathbf{z}} = \frac{HgT}{2L} \frac{\sinh \frac{2\pi(\mathbf{z}+d)}{L}}{\cosh \frac{2\pi d}{L}} \sin(k\mathbf{x} - \omega\mathbf{t} + \phi), \quad (3.15)$$

and its derivative is:

$$\dot{\mathbf{v}}_{p,\mathbf{z}} = -\frac{g\pi H}{L} \frac{\sinh \frac{2\pi(\mathbf{z}+d)}{L}}{\cosh \frac{2\pi d}{L}} \cos(k\mathbf{x} - \omega\mathbf{t} + \phi). \quad (3.16)$$

These velocities and accelerations are calculated for each wave and summed by superposition in the function GETPARTICLES shown in Algorithm 1.

Because the vehicle length is much smaller than the component wavelengths, its displacement is assumed to follow that of a particle. This assumption allows for the fluid velocity to be modeled as an instantaneously uniform flow field where the drag force is defined by the quadratic drag law:

$$\mathbf{F}_d = \frac{1}{2} \rho_{sea} |\mathbf{v}_r| \mathbf{v}_r C_d A_i, \quad (3.17)$$

where F_d is the fluid drag force, ρ_{sea} is the density of seawater, \mathbf{v}_r is the velocity of the vehicle relative to the water, C_d is an experimentally determined drag coefficient, and A_i is the incident Area normal to the fluid flow [21].

Algorithm 1 MPC for ocean wave station-keeping where t is a time vector, λ contains all relevant wave spectra data, and Υ represents states in time and space.

```

1: procedure MPC(  $t, \lambda, \text{robot}, \Upsilon_{target}$  )
2:    $n \leftarrow 1$ 
3:    $\eta \leftarrow \text{LOADSEASTATE}( t, \lambda, \Upsilon_{initial} )$ 
4:   while  $n < \text{simulatorOff}$  do
5:     input  $\leftarrow \text{GETFORECAST}( t, \text{robot}, \lambda, \Upsilon_{target}, n )$ 
6:     robot  $\leftarrow \text{MOVEROBOT}( t, \text{robot}, \lambda, \text{input}, n )$ 
7:      $n \leftarrow n + 1$ 
8:   function GETFORECAST(  $t, \text{robot}, \lambda, \Upsilon_{target}, \text{count}$  )
9:      $i \leftarrow 1$ 
10:     $N \leftarrow t.Horizon / t.Discretization$ 
11:     $J_i, u_i, \Upsilon_i \leftarrow \text{INITIALIZEINPUT}( \text{robot} )$ 
12:     $\delta \leftarrow \text{INITIALIZEDELTA}( )$ 
13:    while  $i < \text{maxIterations}$  and  $\delta > \text{exitCriteria}$  do
14:       $u_{i+1} \leftarrow u_i - \delta$ 
15:      for  $k \in [ 1, 2, \dots, N ]$  do
16:         $\mathbf{v}_p, \dot{\mathbf{v}}_p \leftarrow \text{GETPARTICLES}( t, \Upsilon_i(k), \lambda )$ 
17:         $\Lambda \leftarrow \mathbf{v}_p, \dot{\mathbf{v}}_p$ 
18:         $\Upsilon_{i+1}(k) \leftarrow \text{STATEESTIMATE}( t, \text{robot}, \Lambda, u_{i+1}(k) )$ 
19:         $J_{i+1}(k) \leftarrow \text{GETCOST}( \Upsilon_{target}, \Upsilon_{i+1}(k) )$ 
20:       $\delta \leftarrow \text{GETJACOBIAN}( J_i, J_{i+1}, u_i, u_{i+1} )$ 
21:       $u_i \leftarrow u_{i+1}$ 
22:       $J_i \leftarrow J_{i+1}$ 
23:       $\Upsilon_i \leftarrow \Upsilon_{i+1}$ 
24:       $i \leftarrow i + 1$ 
25:    return  $u_{i+1}$ 

```

3.5 MPC Algorithm Layout

The MPC algorithm shown in Algorithm 1 requires four inputs. The first, t , is a time object containing the overall time vector and horizon and discretization parameters. The second input, λ , is an object containing all relevant wave spectra data. The third input is the robot object, which contains the vehicle dynamics,

states, and error history. The last input, Υ_{target} , is the target state.

Once all inputs are passed through, MPC runs by first generating the initial sea state at `LOADSEASTATE` from the wave spectra data in λ . Then, and until some cessation criteria, `GETFORECAST` generates the optimized motor inputs for each thruster while `MOVEROBOT` passes those inputs and moves the robot one step forward along that control vector.

In `GETFORECAST`, the function `INITIALIZEINPUT` generates initial cost, control, and state vectors, or J_i , \mathbf{u}_i , and Υ_i respectively. The initial control vector, \mathbf{u}_i , is a set of PD control actions along the N horizon. `INITIALIZEDELTA` creates a value, δ , to perturb \mathbf{u}_i and generates a new input, \mathbf{u}_{i+1} , which is then evaluated. First, the wave forces along the input trajectory, \mathbf{v}_p and $\dot{\mathbf{v}}_p$, are calculated through `GETPARTICLES`. Next, \mathbf{v}_p , $\dot{\mathbf{v}}_p$, and the control vector, \mathbf{u}_{i+1} are passed in to `STATEESTIMATE` which estimates the resulting predicted state, Υ_{i+1} . Lastly, Υ_{i+1} is evaluated with respect to the target state, Υ_{target} in `GETCOST` by way of the cost function in (3.1).

After the cost, J_{i+1} , is evaluated over the entire N horizon, the function `GETJACOBIAN` evaluates the Jacobian, or $\partial\mathbf{J}/\partial\mathbf{u}$, and returns a new δ value. This value translates the rate at which cost changes with respect to a change in control inputs, and a δ value growing smaller implies that a control vector is nearing a local minimum. At the end of each evaluation, if δ does not meet some *exitCriteria*, a new input vector is generated and the process is repeated. If it does; however, the optimized control vector, \mathbf{u}_{i+1} , is returned and the robot moves forward along that trajectory.

Chapter 4: Results

One challenge when implementing MPC is determining a preferred prediction horizon. Section 4.1 details how the horizon that balances solution accuracy with computation time is selected for this work. Section 4.2 evaluates the controller with the chosen prediction horizon against a standard PD controller in the same wave climate. Finally, in Section 4.3, sensor observation noise is inserted to the MPC and is compared with the deterministic PD controller.

For all simulations, the robot was to maintain a depth of $\mathbf{z} = -15$ m below the surface, or $\Upsilon_{target} = [0, -15]$. The simulator is designed to simulate wave forces at any depth; however, simulations were not run at varied depths as the scaled results offered little additional insight. The *exitCriteria* for an optimized trajectory used in Algorithm 1 was set to 5 mm. Visualizations of simulator data are provided by MATLAB figures as shown in Fig.4.1.

4.1 Determination of Prediction Horizon

The desired prediction horizon for MPC should reasonably balance computation time against position error. This is a discretionary characteristic as certain implementations may have stricter tolerances than others. In this work, simulations were run with a horizon ranging from 0.2s to 3.0s. Table 4.1 shows total RMS

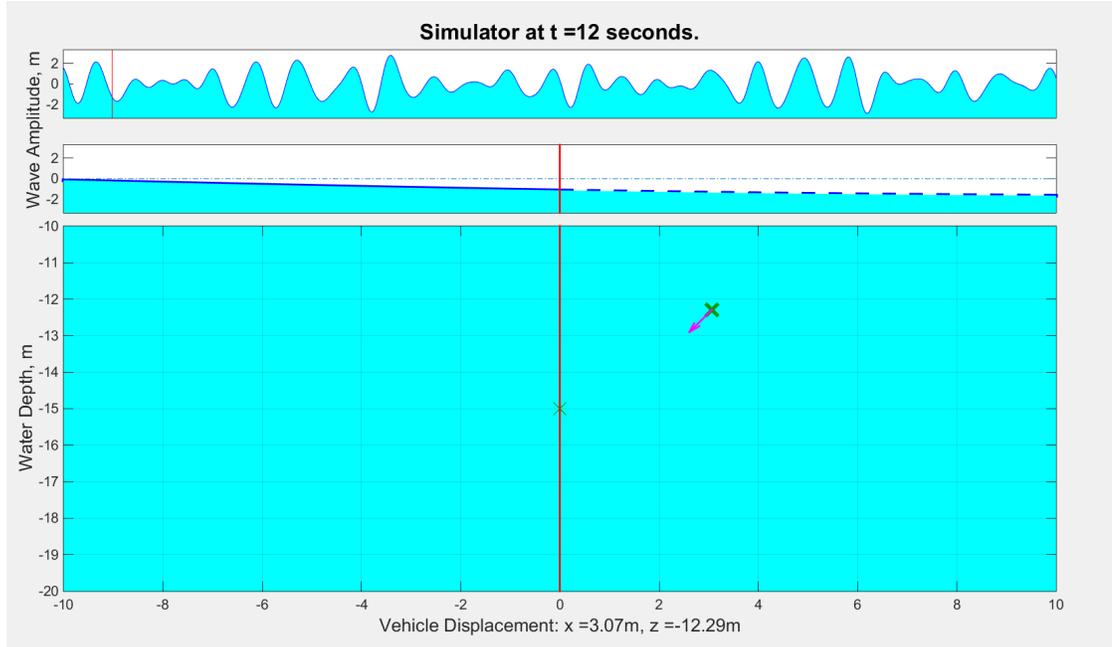


Figure 4.1: Simulator visualization of a displaced robot (the bold “X”) attempting to reach the target state (the faded “X”). Wave field profile data is provided as well as current vehicle state and velocity vector.

errors and computation times per horizon.

Table 4.1: Performance of Various Simulated Horizons

Horizon, s	0.2	0.4	0.8	1.0	1.6	2.0	3.0
ϵ_{RMS} , m	5.02	2.11	0.79	0.65	0.29	0.05	$9.0E-6$
$\bar{t}_{Calc/step}$	1.38	0.42	0.08	0.19	6.51	18.24	84.38

Based on the information in Table 4.1, two potential choices for prediction horizon are 0.8 and 1.0s. Both offer low error at reasonable computation times. For the results presented in this work, the chosen horizon time was 0.8s as it yielded only a 6 cm difference in error for less than half the total computation time.

With regards to the 1.0 s horizon, it is worth noting that the 240 s long time vector used in simulation is just longer than the 233.2 s run time needed. This presents challenges when considering real-time implementation of this method. One solution is to recalculate the next optimized trajectory not at each 0.2 s time step, but one or multiple steps later. In practice, the robot would use the first two or three inputs from the optimized control vector instead of just the first, while simultaneously calculating the next trajectory. This would lead to fewer calculations but may increase overall error due to sensor drift.

Another result worth noting is the long run times from the shorter horizon controllers (0.2 and 0.4 s). They showed poor performance, resulting in significantly larger tracking error and requiring more calculation time than the chosen horizon of 0.8 s. This is due to an induced "myopia" where: 1) the robot does not properly account for its own inertia and 2) the robot does not take proper advantage of changes in flow direction. The initial guess for optimized trajectories is the PD control along the horizon. Therefore, the initial guess for the lower horizons is so inaccurate that the optimizer uses many more steps to generate a trajectory.

As expected, the longer horizons yield reduced errors at the expense of computation time. None of the later three time horizons (1.6, 2.0, and 3.0 s) were considered because of unreasonable time costs. An obvious avenue for future work is to implement more efficient optimization and programming techniques to further increase the planning horizon.

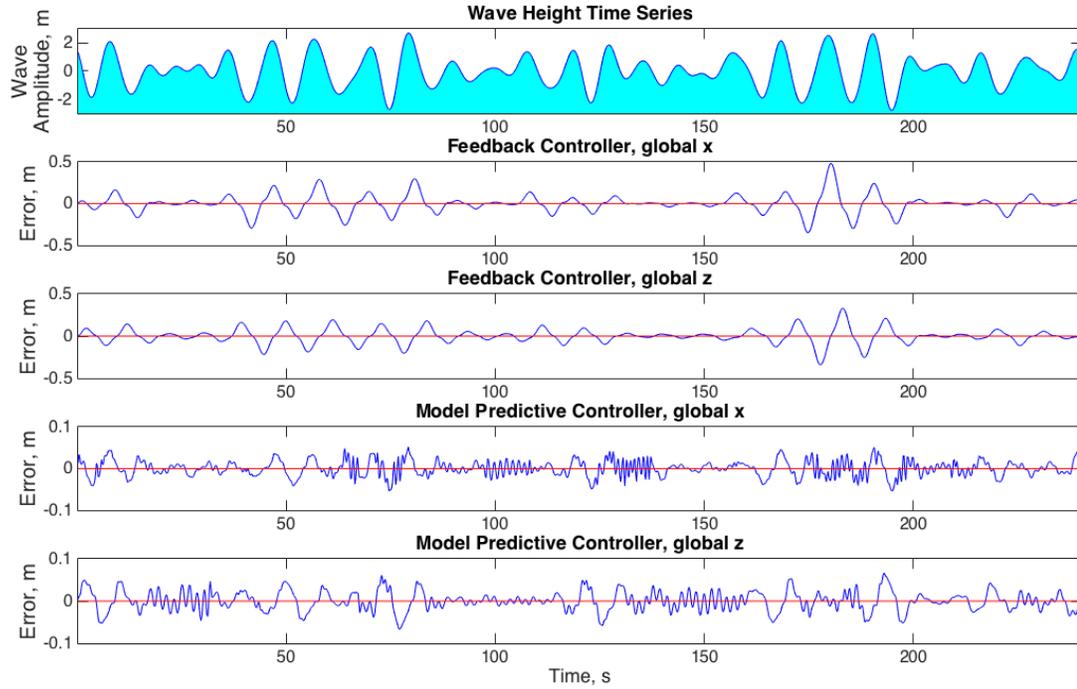


Figure 4.2: Position error time series in global \mathbf{x} and \mathbf{z} coordinates when comparing a traditional feedback controller with a model predictive controller. Note the axis scale difference. As shown, MPC returns error values 74% lower than PD Control. RMS error values for these results are shown in Fig 4.3.

4.2 MPC Performance

Another set of simulations was run where a robot employing an 0.8 s horizon MPC was compared against one using a traditional PD controller. Both were tested under the influence of the same wave field. Additionally, a free-floating, non-actuated robot disturbed by the same waves was simulated and compared for reference. Figure 4.2 details the positional errors for the PD and MPC controllers over the length of simulated time and Fig. 4.3 shows the RMS error for each case.

As shown in Fig. 4.2, MPC gives a 74% reduction in position error over PD

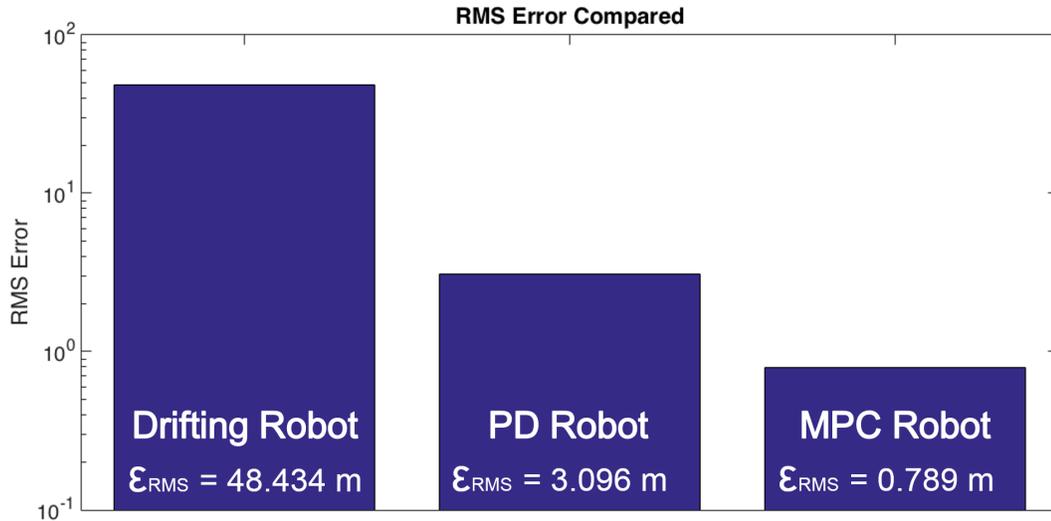


Figure 4.3: RMS errors for the three cases in Section 4.2. The MPC robot showed a 74 % reduction in error compared to another using PD control.

control. This substantial reduction is attained because the robot state estimator minimizes cost by using thrusters in an anticipatory action. In practice, the robot would thrust "against" the wave to reduce net displacement. Without any forward-looking state estimation, the PD controller can only choose a direct trajectory towards the desired state that is always deviated by the wave action.

In Fig. 4.2, the wave field time series is provided to show the correlation between wave height, H , which is directly proportional with wave forces in the water column [14], and position error below the waves. Both the MPC and PD controllers record little error as H approaches zero, which is expected. As H increases; however, the PD errors increase at a larger rate than the increase in MPC errors. This again results from the state estimator's ability to predict changes in flow direction and the impending effects of those changes on the robot's inertia.

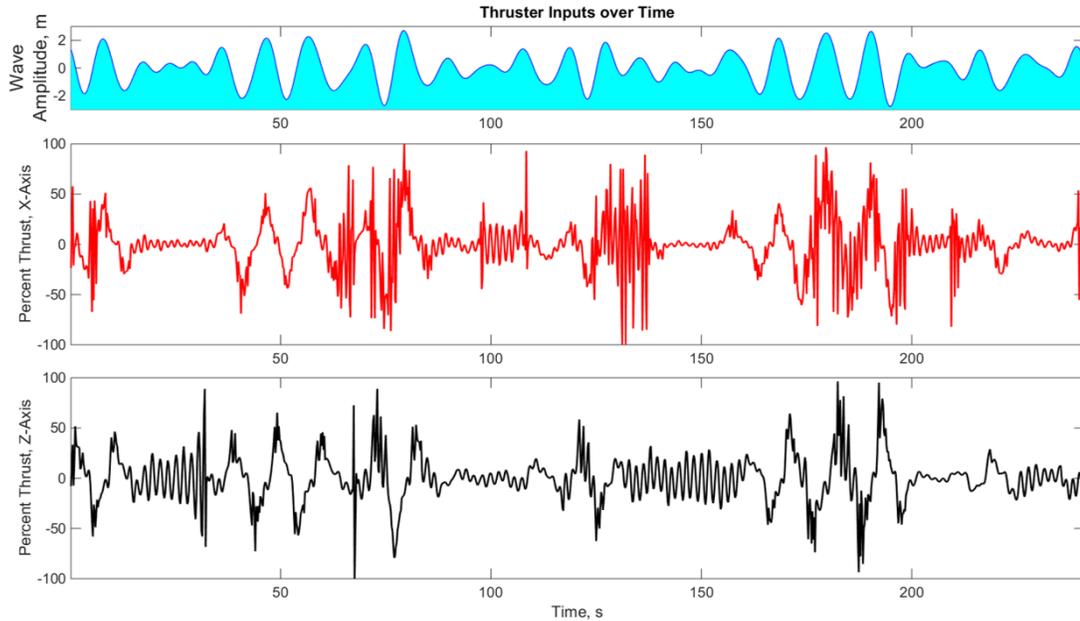


Figure 4.4: MPC thruster inputs in global \mathbf{x} and \mathbf{z} coordinates over the simulated time series. The controller does not issue commands that saturate the thrusters for prolonged periods; however, it is prone to frequent direction changes.

Figure 4.4 shows a time series of thruster inputs in percentages of maximum thrust. As expected, the controller issues less thrust when the wave action is reduced. This is a result of the added input term to the cost function minimized in (3.1). One concern about the inputs shown is that there are frequent changes in direction. This can potentially lead to unrealistic results as thruster actuation time was not considered in this work. A better model of thruster dynamics as well as an updated cost function to penalize large changes in thruster input are both avenues for future work.

4.3 Sensor Noise Impact

Simulations were also carried out to compare optimized trajectories to the effects of simulated sensor observation noise. For this set, Gaussian noise is injected to the vehicle’s perceived value of particle velocities and accelerations, \mathbf{v}_p and $\dot{\mathbf{v}}_p$. The wave field parameters were selected instead of other forms of localization noise because the vehicle Inertial Measurement Unit (IMU), which measures water pressure, has lower resolution than the Doppler Velocity Log (DVL), thus it is more likely that the IMU would give inaccurate observations of the pressure field above it (the wave action) than the DVL with bottom lock would on vehicle localization. By extension, this assumption allows for the PD control data from Section 4.2 to serve as a deterministic basis of comparison.

The additive, independent, and identically-distributed Gaussian noise is provided by the MATLAB function `AWGN`. This function injects noise to the desired signal according to a Signal-to-Noise Ratio (SNR) which is related to the noise variance. For this simulation, the wave height, H , was assigned the highest variance to account for the moderately noisy heave data provided by the deployed buoy sensors [21]. Noise with a smaller variance is injected into the wave period, T , and phase terms, ϕ , to account for error in Fourier coefficients and their resulting phase transformations.

50 simulations of the MPC Algorithm 1 were run where at each n^{th} step, the function `GETFORECAST` is run with a noisy estimation of the wave field. This outputs a trajectory whose first element is then carried out in the `MOVEROBOT`

Table 4.2: Performance of MPC with Noisy Wave Observations

Scenario	ϵ_{RMS} , m
Model Predictive Control	0.789
Mean MPC with Gaussian noise	1.737
Feedback (PD) Control	3.096
Drifting Robot	48.434

function. The process is repeated with a new noisy wave field estimation. Table 4.2 shows the RMS errors for MPC with noise against PD, MPC, and drift values. The MPC with noise shows a mean error of 1.737 m with standard deviation of 0.059 and gives a 43.9% reduction in position error over feedback control.

Chapter 5: Conclusion

This thesis presented a Model Predictive Control (MPC) approach to reducing underwater robot position error under the influence of water waves. Our method employed a Linear Wave Theory (LWT) solver to approximate the component fluid dynamics under a wave field. These fluid flow velocities and accelerations are input to a model state estimator which predicts robot state along a finite horizon. A set of control actions which minimizes a cost function is generated and optimized via gradient descent. Thus, the contributions of this thesis are:

- A feedforward control (MPC) method that can estimate and compensate for impending wave action.
- Comparison of MPC to PD control for a station-keeping robot disturbed by the same wave field, where position error was reduced by 74 %. The algorithm was further found to be resistant to noisy sensor observations of wave field parameters, showing a mean position error 44 % lower than the deterministic feedback control case.
- Determination of the most effective prediction horizon as 4 steps, or 0.8s forward as it reasonably balanced solution accuracy and computation time.

For future work, the dynamics of the system could be expanded to incorporate all five vehicle degrees of freedom. With this extension, the wave field would better model a 3-dimensional random sea. Second, the vehicle model could be expanded to include effects from Response Amplitude Operators (RAO). RAOs are frequency-domain solutions for a vehicle response to a defined wave field along all reference axes. Hydrodynamic LWT solvers such as Ansys AQWA are potential tools for incorporating RAOs for a given vehicle model. With a well-modeled RAO, neuro-evolutionary possibilities emerge. Thirdly, simplifications were made in this work to help the robot predict the wave field. Sensor detection for spectral data as well as their decomposition methods are two areas of interest when implementing this method. Fourthly, analyses on system dependence on wavenumber should be worked out using sets of monochromatic waves. This is important so see if there are any specific harmonics which make the system unstable or uncontrollable. Lastly, more efficient optimization techniques than gradient descent should be explored.

Continued areas of research in the marine robotics community include autonomous underwater path planning, localization, and perception. Efficient path planning helps reduce overall mission cost and time by optimizing methods of navigating. Localization issues in an environment absent GPS and long range wireless transmissivity prioritize the need for well-developed SLAM techniques with minimal input. Robotic perception in the underwater domain further complicates research efforts. In addition, multi-agent research, such as the multi-glider work in [31], is an excellent demonstration of the value of autonomy in performing oceanographic monitoring.

Advancements in underwater autonomy will be pivotal in the development of offshore energy arrays, since low-cost robotic platforms inspecting, monitoring, and manipulating infrastructure can reduce deployment costs drastically. Over the course of the NNMREC ALFA project, robust algorithms for these marine platforms to support WECs will lead to improved scaled economics and further global wave energy development. As more challenges are addressed, this will help secure wave energy extraction as the premier sustainable energy source for the 21st century.

Bibliography

- [1] ANSYS AQWA. “User’s Manual (ver. 14.0)”. In: *ANSYS Incorporated, Canonsburg, PA* (2012).
- [2] R. Bachmayer, N. E. Leonard, J Graver, E Fiorelli, P. Bhatta, and D. Paley. “Underwater gliders: recent developments and future applications”. In: *International Symposium on Underwater Technology (UT), 2004*. 2004, pp. 195–200.
- [3] R. D. Ballard, L. E. Stager, D. Master, D. Yoerger, D. Mindell, L. L. Whitcomb, H. Singh, and D. Piechota. “Iron Age Shipwrecks in Deep Water off Ashkelon, Israel”. English. In: *American Journal of Archaeology* 106.2 (2002), pp. 151–168. ISSN: 00029114.
- [4] S. Barkby, S. Williams, O. Pizarro, and M. Jakuba. “Bathymetric Particle Filter SLAM Using Trajectory Maps”. In: *The International Journal of Robotics Research* (2012).
- [5] J. Binney, A. Krause, and G. Sukhatme. “Informative path planning for an autonomous underwater vehicle”. In: *IEEE International Conference on Robotics and Automation (ICRA), 2010*. 2010, pp. 4791–4796.

- [6] J. Binney and G. S. Sukhatme. “Branch and bound for informative path planning”. In: *IEEE International Conference on Robotics and Automation (ICRA), 2012*. 2012, pp. 2147–2154.
- [7] B. Bosma, T. K. A. Brekken, H. T. Ozkan-Haller, and S. C. Yim. “Wave energy converter modeling in the time domain: A design guide”. In: *2013 1st IEEE Conference on Technologies for Sustainability (SusTech)*. 2013, pp. 103–108.
- [8] T. K. A. Brekken. “On Model Predictive Control for a point absorber Wave Energy Converter”. In: *PowerTech, 2011 IEEE Trondheim*. 2011, pp. 1–8.
- [9] E. F. Camacho and C. Bordons. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer International Publishing, 2013.
- [10] R. Campos, R. Garcia, P. Alliez, and M. Yvinec. “A surface reconstruction method for in-detail underwater 3D optical mapping”. In: *The International Journal of Robotics Research* (2014).
- [11] S. M. Chaves, A. Kim, and R. M. Eustice. “Opportunistic sampling-based planning for active visual SLAM”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014*. 2014, pp. 3073–3080.
- [12] R. D. Christ and R. L. Wernli Sr. *The ROV Manual: A User Guide for Remotely Operated Vehicles*. Butterworth-Heinemann, 2013.

- [13] M. K. Colby, E. M. Nasroullahi, and K. Tumer. “Optimizing ballast design of wave energy converters using evolutionary algorithms”. In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 1739–1746.
- [14] R. G. Dean and R. A. Dalrymple. *Water Wave Mechanics for Scientists and Engineers*. Vol. 2. Advanced Series on Ocean Engineering. World Scientific, 1991.
- [15] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus. “Data muling over underwater wireless sensor networks using an autonomous underwater vehicle”. In: *IEEE International Conference on Robotics and Automation (ICRA), 2006*. 2006, pp. 2091–2098.
- [16] M. Eich, F. Bonnin-Pascual, E. García-Fidalgo, A. Ortiz, G. Bruzzone, Y. Koveos, and F. Kirchner. “A Robot Application for Marine Vessel Inspection”. In: *Journal of Field Robotics* 31.2 (2014), pp. 319–341. ISSN: 1556-4967.
- [17] J. Falnes. *Ocean Waves and Oscillating Systems: Linear Interactions including Wave-Energy Extraction*. Cambridge University Press, 2002.
- [18] E. Fernández-Perdomo, J. Cabrera-Gámez, D. Hernández-Sosa, J. Isern-González, A. C. Dominguez-Brito, A. Redondo, J. Coca, A. G. Ramos, E. A. Fanjul, and M. García. “Path planning for gliders using Regional Ocean Models: Application of Pinzón path planner with the ESEOAT model and the RU27 trans-Atlantic flight data”. In: *OCEANS 2010 IEEE - Sydney*. 2010.

- [19] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao. “Coverage Path Planning with Real-time Replanning and Surface Reconstruction for Inspection of Three-dimensional Underwater Structures using Autonomous Underwater Vehicles”. In: *Journal of Field Robotics* 32.7 (2015), pp. 952–983. ISSN: 1556-4967.
- [20] Y. Girdhar, P. Giguère, and G. Dudek. “Autonomous Adaptive Underwater Exploration using Online Topic Modeling”. English. In: Springer Tracts in Advanced Robotics 88 (2013). Ed. by J. Desai, G. Dudek, O. Khatib, and V. Kumar, pp. 789–802.
- [21] Y. Gōda. *Random Seas and Design of Maritime Structures*. Vol. 33. Advanced Series on Ocean Engineering. World Scientific, 2010.
- [22] G. Hitz, A. Gotovos, F. Pomerleau, M. E. Garneau, C. Pradalier, A. Krause, and R. Y. Siegwart. “Fully autonomous focused exploration for robotic environmental monitoring”. In: *IEEE International Conference on Robotics and Automation (ICRA), 2014*. 2014, pp. 2658–2664.
- [23] G. A. Hollinger, U. Mitra, and G. S. Sukhatme. “Active and Adaptive Dive Planning for Dense Bathymetric Mapping”. English. In: *Experimental Robotics*. Vol. 88. Springer Tracts in Advanced Robotics. Springer International Publishing, 2013, pp. 803–817. ISBN: 978-3-319-00064-0.
- [24] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme. “Active planning for underwater inspection and the benefit of adaptivity”. In: *The International Journal of Robotics Research* 32.1 (2013), pp. 3–18.

- [25] G. A. Hollinger, B. Englot, F. Hover, U. Mitra, and G. S. Sukhatme. “Uncertainty-driven view planning for underwater inspection”. In: *IEEE International Conference on Robotics and Automation (ICRA), 2012*. 2012, pp. 4884–4891.
- [26] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard. “Advanced perception, navigation and planning for autonomous in-water ship hull inspection”. In: *The International Journal of Robotics Research* 31.12 (2012), pp. 1445–1464.
- [27] V. T. Huynh, M. Dunbabin, and R. N. Smith. “Predictive motion planning for AUVs subject to strong time-varying currents and forecasting uncertainties”. In: *IEEE International Conference on Robotics and Automation (ICRA), 2015*. 2015.
- [28] J. Joslin, E. Celkis, C. Roper, A. Stewart, and B. Polagye. “Development of an Adaptable Monitoring Package for marine renewable energy”. In: *Oceans - San Diego, 2013*. 2013, pp. 1–10.
- [29] A. Kim and R. M. Eustice. “Active visual SLAM for robotic area coverage: Theory and experiment”. In: *The International Journal of Robotics Research* 34.4-5 (2015), pp. 457–475.
- [30] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [31] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang. “Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay”. In: *Journal of Field Robotics* 27.6 (2010), pp. 718–740. ISSN: 1556-4967.

- [32] B. A. Ling and B. A. Batten. “Real time estimation and prediction of wave excitation forces on a heaving body”. In: *ASME International Conference on Ocean, Offshore and Arctic Engineering (OMAE), 2015*. 2015.
- [33] L. Medagoda and S. B. Williams. “Model predictive control of an autonomous underwater vehicle in an in situ estimated water current profile”. In: *OCEANS, 2012 - Yeosu*. 2012, pp. 1–8.
- [34] L. Medagoda, J. C. Kinsey, and M. Eilders. “Autonomous underwater vehicle localization in a spatiotemporally varying water current field”. In: *IEEE International Conference on Robotics and Automation (ICRA), 2015*. 2015, pp. 2091–2098.
- [35] A. Mora, C. Ho, and S. Saripalli. “Analysis of Adaptive Sampling Techniques for Underwater Vehicles”. In: *Autonomous Robots* 35.2-3 (2013), pp. 111–122. ISSN: 0929-5593.
- [36] P. Ngo, J. Das, J. Ogle, J. Thomas, W. Anderson, and R. N. Smith. “Predicting the speed of a Wave Glider autonomous surface vehicle from wave model data”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014*. 2014, pp. 2250–2256.
- [37] N. S. Nise. *Control Systems Engineering*. John Wiley & Sons, 2011.
- [38] G. Papadopoulos, H. Kurniawati, A. S. B. M. Shariff, L. J. Wong, and N. M. Patrikalakis. “Experiments on Surface Reconstruction for Partially Submerged Marine Structures”. In: *Journal of Field Robotics* 31.2 (2014), pp. 225–244. ISSN: 1556-4967.

- [39] O. M. Phillips. *The Dynamics of the Upper Ocean*. Cambridge University Press, 1977.
- [40] C. Pêtrès, Y. Pailhas, P. Patrón, Y. Petillot, J. Evans, and D. Lane. “Path Planning for Autonomous Underwater Vehicles”. In: *IEEE Transactions on Robotics on 23.2* (2007), pp. 331–341. ISSN: 1552-3098.
- [41] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [42] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill, 2009.
- [43] M. Richter, M. E. Magaña, O. Sawodny, and T. K. A. Brekken. “Nonlinear Model Predictive Control of a Point Absorber Wave Energy Converter”. In: *Sustainable Energy, IEEE Transactions on 4.1* (2013), pp. 118–126. ISSN: 1949-3029.
- [44] J. S. Riedel. “Shallow water stationkeeping of an autonomous underwater vehicle: the experimental results of a disturbance compensation controller”. In: *OCEANS 2000 MTS/IEEE Conference and Exhibition*. Vol. 2. 2000, pp. 1017–1028.
- [45] J. S. Riedel and A. J. Healey. “Shallow water station keeping of AUVs using multi-sensor fusion for wave disturbance prediction and compensation”. In: *OCEANS '98 Conference Proceedings*. Vol. 2. 1998, 1064–1068 vol.2.

- [46] R. N. Smith, J. Das, G. Hine, W. Anderson, and G. Sukhatme. “Predicting Wave Glider speed from environmental measurements”. In: *OCEANS 2011*. 2011, pp. 1–8.
- [47] R. Smith, P. Cooksey, F. Py, G. Sukhatme, and K. Rajan. “Adaptive Path Planning for Tracking Ocean Fronts with an Autonomous Underwater Vehicle”. In: *Proceedings, International Symposium on Experimental Robotics*. Morocco, 2014.
- [48] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme. “Persistent ocean monitoring with underwater gliders: Adapting sampling resolution”. In: *Journal of Field Robotics* 28.5 (2011), pp. 714–741. ISSN: 1556-4967.
- [49] R. N. Smith, Y. Chao, B. H. Jones, D. A. Caron, P. P. Li, and G. S. Sukhatme. “Trajectory Design for Autonomous Underwater Vehicles Based on Ocean Model Predictions for Feature Tracking”. English. In: *Field and Service Robotics*. Ed. by A. Howard, K. Iagnemma, and A. Kelly. Vol. 62. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2010, pp. 263–273. ISBN: 978-3-642-13407-4.
- [50] SolidWorks. “User’s Guide”. In: *Dassault Systèmes SolidWorks Corp, Concord, Massachusetts* (2012).
- [51] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT press, 2005.
- [52] G. Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.

