Model Predictive Control for Underwater Robots in Ocean Waves

Daniel C. Fernández[1] and Geoffrey A. Hollinger[2]

***Abstract*— Underwater robots beneath ocean waves can benefit from feedforward control to reduce position error. This paper 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, showing a mean position error 44% lower than the noise-free feedback control case.**

I. Introduction

The coastal ocean is bounded by the shoreline and the 200 m isobath, and is a common experimental setting for field robotics. This is the most biologically productive area of ocean [1] and is subject to the majority of natural and industrial disasters. 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 robotic advancement in all of these coastal research areas predicates the need to better understand the dynamics of this environment.

Wave forces in the intermediate depths 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, and sufficient depths yield negligible disturbances [2]. 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, this assumption can quickly break down. As a result, increased sensor drift can hinder the quality of robotic observations, such as those needed to close SLAM loops in [3] and [4].

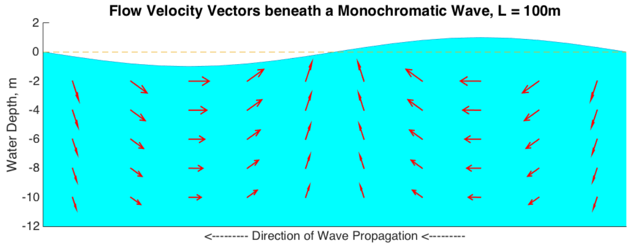

Fig. 1. A visualization of the flow field velocity at various depths beneath a monochromatic wave. These vector lengths are not to scale and are intended to conceptualize the direction and decay with depth of wave forces.

Traditional PID control techniques can be used to counter wave displacements, but reaction times for underwater robots are slow relative to the changing wave forces. Given the periodicity of waves, feedforward techniques should be explored. This paper 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 of the work presented in this paper is a control method that can forecast and compensate for future wave action. This is demonstrated through simulations of an underwater robot performing station keeping in a shallow 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 allows us to increase the quality of robotic observations in shallow ocean water, as well as reducing the risk of equipment damage while deployed. MPC shows to reduce an underwater robot 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.

The remainder of this paper is organized as follows: related work is highlighted in Section II. Next, the MPC method is described in Section III. Section IV describes the wave mechanics used in the system model. Section V outlines the vehicle model and overall system dynamics. Section VI details the algorithm structure and controller specifics. Section VII 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 Section VIII.

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

[1]D. C. Fernández, [2]G. A. Hollinger are currently with the Robotics Program at the School of Mechanical, Industrial and Manufacturing Engineering, Oregon State University, Corvallis, Oregon, 97331-6001, United States {fernanda, geoff.hollinger}@oregonstate.edu

II. Related Work

Path optimization is a key aspect of robotic path planning where the amount of information gain should be maximized while considering some cost function (e.g., mission duration). In [5] and [6], a sampling path is designed to minimize the distance between an AUV and specific oceanographic point of interests. The hybrid Fast Marching method, or FM*, employed in [7] uses an A* search heuristic to find a continuous path while minimizing drift. These methods incorporate ocean current models as a quadratic drag force, but do not include influence from wave forces. Localization challenges, such as the mid-column glider position uncertainty presented in [8], are further complicated when considering additive uncertainty due to wave disturbances.

Robotic maintenance of Wave Energy Converters (WEC), such as the platform detailed in [9], is an active area of research. In fact, MPC techniques have been explored by the wave energy community as a way to optimize WEC power generation. As shown in [10] and [11], 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 [12], 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 spectra data. 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.

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. The method in [14] uses visual servoing MPC to provide aperiodic, algorithm-generated control loop closures to an AUV influenced by ocean currents. In [15], energy efficient paths for a glider are generated by minimizing costs across stratified, spatially-distributed currents using an A* heuristic. In [16], 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 paper. These approaches show the value in state estimation for robotic path planning, but do not incorporate wave disturbances.

Wave-induced station keeping is explored in [17] and [18] where a dynamic Kalman filter is created to synthesize onboard sensor data and provide an estimate of wave-induced disturbances. The filter fuses ground speed from the Doppler Velocity Log (DVL), relative water speed from the Acoustic Doppler Current Profiler (ADCP), and attitude from the Inertial Measurement Unit (IMU) among other sensors to produce an estimate of the fluid velocity. This method highlights some of the difficulties of estimating *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 paper seeks to improve on these results by providing improved cost minimization techniques and emphasizing real-time control execution.

III. Model Predictive Control

The term MPC does not refer to a specific strategy, but a variety of methods that all in some way incorporate the following [19]:

- 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 [20]; 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 [21], [22]. 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:

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

and

$$\mathbf{u}^*_{1:N} = \underset{\mathbf{u}_{1:N}}{\arg\min}\, J(\mathbf{u}_{1:N}), \tag{2}$$

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 $\mathbf{u}_k^2$ term in (1) acts to minimize energy usage where possible.

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 (2). This gradient descent method of cost evaluation and optimization is similar to that used by Medagoda et al. in [16].

IV. Wave Mechanics

According to Linear Wave Theory (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) [1], [2]. For reference, this paper instead uses the term waveheight (H), which is simply twice the amplitude ($H = 2a$). LWT assumes fluid flow is irrotational, incompressible, and inviscid, thus allowing for potential flow [23]. In practice, LWT is not only easily implemented, but has also been shown to produce surprisingly accurate results [2]. Thus, LWT is often employed for a reliable primary analysis before other nonlinear theories, and it forms the initial approximation for the wave forces modelled in this paper.

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) [24]. DFT solvers will output frequency parameters with associated energies correlating to signal amplitude. The phase information can then be extracted by taking the tan^{-1} of the imaginary over the real part of each component. For details on Fourier spectra of wave fields, see [25].

In this work, 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) for each wave (λ) in the set of waves (Λ) are solved for by way of the dispersion relation [2]. The superimposed wave field shown in Fig. 3 is then constructed by the water surface wave equation shown:

$$\eta(\mathbf{t}) = \sum_{\lambda=1}^{\Lambda} \frac{H(\lambda)}{2} cos(k(\lambda)\mathbf{x} - \omega(\lambda)\mathbf{t} + \phi(\lambda)). \tag{3}$$

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 [2]. 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, $d \approx L/2$ [2]. The per-wave, at-depth solutions to these velocity and acceleration equations are used as the primary inputs to the modeled system dynamics.

V. System Dynamics

For the scope of this paper, the term Remotely Operated Vehicle (ROV) will be used to describe a tethered unmanned vehicle whose operation may or may not be teleoperated [26]. The ROV modeled for this work is the SeaBotix vLBV300 shown in Fig. 2. In simulation, the ROV is not teleoperated and performs control actions autonomously.

A. Vehicle Model

The SeaBotix vLBV300 ROV has six vectored thrusters controlling motion along five degrees of freedom. For the scope of this work, and to test the initial feasibilty of MPC in ocean waves, only the surge and heave directions, or forward (global $\mathbf{x}$) and vertical (global $\mathbf{z}$) axes, are considered. See Fig. 2 for reference.

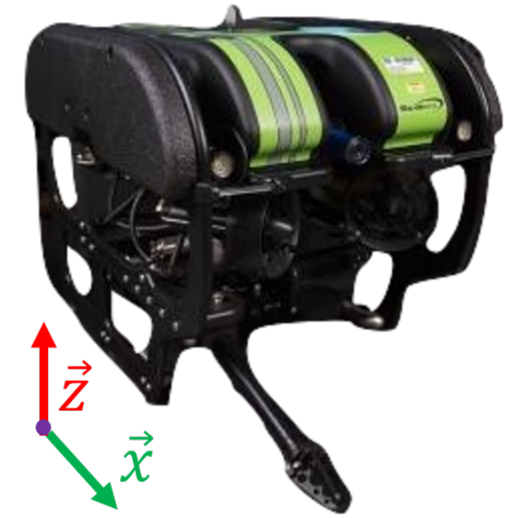

Fig. 2. A SeaBotix vLBV300 ROV similar to the one modeled in this work. Along with IMU and DVL sensors, the vehicle has six angled thrusters which control it along five degrees of freedom: heave (vertical z), sway (lateral y), surge (lateral x), roll (rotation around x), and yaw (rotation around z). For this work, only the x and z axes are considered.

TABLE I
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°

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 I. Moment of inertia and center of gravity data was provided by a CAD model input to Dassault Systèmes SolidWorks [27]. Added mass values were provided by the same model input to Ansys AQWA [28]. Drag coefficient data was provided by the manufacturer. The differential equation defining ROV motion is:

$$\mathbf{M}\dot{\mathbf{v}}_a = \mathbf{F}_{thrust} + \mathbf{F}_d + \mathbf{F}_g + \mathbf{F}_c, \tag{4}$$

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, $\mathbf{F}_d$ is the drag force from the water, $\mathbf{F}_g$ is the force of gravity, $\mathbf{F}_c$ is the Coriolis force, and $\mathbf{F}_{thrust}$ is the ROV thruster force. By assuming that the Coriolis force is negligible and that the vehicle is neutrally buoyant, the $\mathbf{F}_c$ and $\mathbf{F}_g$ terms are neglected. Decomposing the $\mathbf{M}$ term and substituting inertia and drag relations, (4) becomes:

$$m_{dry}\dot{\mathbf{v}}_a + m_{add}\dot{\mathbf{v}}_r = \mathbf{F}_{thrust} + \frac{1}{2}\rho_{sea}A_i c_d|\mathbf{v}_r|\mathbf{v}_r, \tag{5}$$

where $\mathbf{v}_r$ is the of the vehicle relative to the water, which acts on the added mass term of $\mathbf{M}$ in (4). Substituting heave, surge, and particle velocity components, (5) becomes:

$$\begin{aligned}\begin{bmatrix} m_{dry} + m_{add,\mathbf{x}} \\ m_{dry} + m_{add,\mathbf{z}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{thrust,\mathbf{x}} \\ \mathbf{F}_{thrust,\mathbf{z}} \end{bmatrix} + \begin{bmatrix} m_{add,\mathbf{x}} \\ m_{add,\mathbf{z}} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_{p,\mathbf{x}} \\ \dot{\mathbf{v}}_{p,\mathbf{z}} \end{bmatrix} \\ + \frac{\rho_{sea}}{2} \begin{bmatrix} A_{i,\mathbf{x}}C_{d,\mathbf{x}} \\ A_{i,\mathbf{z}}C_{d,\mathbf{z}} \end{bmatrix} \begin{bmatrix} |\dot{\mathbf{x}} - \mathbf{v}_{p,\mathbf{x}}|(\dot{\mathbf{x}} - \mathbf{v}_{p,\mathbf{x}}) \\ |\dot{\mathbf{z}} - \mathbf{v}_{p,\mathbf{z}}|(\dot{\mathbf{z}} - \mathbf{v}_{p,\mathbf{z}}) \end{bmatrix},\end{aligned} \tag{6}$$

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 $\dot{\mathbf{v}}_{a,\mathbf{x}}$ = $\ddot{\mathbf{x}}$, $\dot{\mathbf{v}}_{a,\mathbf{z}}$ = $\ddot{\mathbf{z}}$.

B. State Space Form

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

$$\dot{\mathbf{\Upsilon}} = \begin{bmatrix} \mathbf{x} & \dot{\mathbf{x}} & \mathbf{z} & \dot{\mathbf{z}} \end{bmatrix}^T = \mathbf{Ax} + \mathbf{Bu} + \mathbf{D}, \tag{7}$$

where

$$\mathbf{Ax} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \\ \mathbf{z} \\ \dot{\mathbf{z}} \end{bmatrix}, \tag{8}$$

$$\mathbf{Bu} = \frac{T_{max}}{m_{dry}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ cos\theta_f & cos\theta_f & -cos\theta_a & -cos\theta_a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -cos\theta_v & -cos\theta_v \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix}, \tag{9}$$

and

$$\mathbf{D} = \begin{bmatrix} 0 \\ \frac{\dot{\mathbf{v}}_{p,\mathbf{x}}}{m_{dry}} + \frac{\rho_{sea}A_{i,\mathbf{x}}C_{d,\mathbf{x}}}{2(m_{dry}+m_{add,\mathbf{x}})}|\dot{\mathbf{x}} - \mathbf{v}_{p,\mathbf{x}}|(\dot{\mathbf{x}} - \mathbf{v}_{p,\mathbf{x}}) \\ 0 \\ \frac{\dot{\mathbf{v}}_{p,\mathbf{z}}}{m_{dry}} + \frac{\rho_{sea}A_{i,\mathbf{x}}C_{d,\mathbf{z}}}{2(m_{dry}+m_{add,\mathbf{z}})}|\dot{\mathbf{z}} - \mathbf{v}_{p,\mathbf{z}}|(\dot{\mathbf{z}} - \mathbf{v}_{p,\mathbf{z}}) \end{bmatrix}. \tag{10}$$

The $\mathbf{u}$ vector in (9) 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.

VI. Simulator Setup

The ROV was simulated along a 240 s long time vector with 0.2 s discretizations. All robot thruster actuations and wave disturbances occur along the global $\mathbf{x}$ and $\mathbf{z}$ axes. Each motor input simulated by the $\mathbf{Bu}$ matrix in (9) is a thresholded value between [-1,1], which correlates to a percentage of maximum possible thrust. All forward-looking state estimations and robot movements use the MATLAB function ODE45 which uses a variable step Runge-Kutta Method to numerically solve the system differential equations [29].

TABLE II

Wave field parameters used in simulation

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

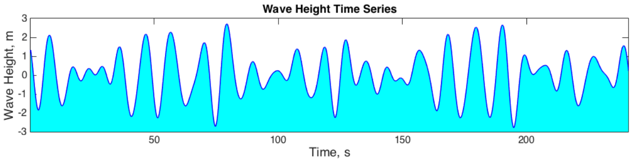

Fig. 3. A time series of the wave profile over the 240 s simulated time window constructed using the parameters listed in Table II.

A. 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 [12]. The operational depth at NETS is approximately 50 m. The wave field used in this work was qualitatively verified using AWAC acoustic measurement data deployed at NETS in August 2013. Each 40 min time series was recorded at 2 Hz.

The wave field used by the simulator is the set of eight different periods, heights, and phases (T, H, ϕ) detailed in Table II. By assuming LWT and using superposition, a reconstructed time series using (3) is shown in Fig. 3. Because of the 2-dimensional analysis along only surge and heave motions, wave angles are set to 0.

Prior to any forward state estimation or any simulated vehicle motion, the wave action, or particle accelerations and velocities, are calculated for the robot's current position in time and space. These accelerations and velocities are calculated for each wave and summed by superposition in the function GETPARTICLES shown in Algorithm 1.

B. 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 $\mathbf{\Upsilon}_i$ respectively. The initial control vector, $\mathbf{u}_i$, is a set of PD control actions along the N horizon. This initial control

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

```
procedure $MPC$( $t$, $\lambda$, robot, $\Upsilon_{target}$ )
    $n \leftarrow 1$
    $\eta \leftarrow$ LOADSEASTATE( $t$, $\lambda$, $\Upsilon_{initial}$ )
    while $n <$ simulatorOff do
        input $\leftarrow$ GETFORECAST($t$, robot, $\lambda$, $\Upsilon_{target}$, $n$)
        robot $\leftarrow$ MOVEROBOT( $t$, robot, $\lambda$, input, $n$ )
        $n \leftarrow n + 1$
function GETFORECAST( $t$, robot, $\lambda, \Upsilon_{target}, count$ )
    $i \leftarrow 1$
    $N \leftarrow t.Horizon$ / $t.Discretization$
    $J_i$, $u_i$, $\Upsilon_i \leftarrow$ INITIALIZEINPUT( robot )
    $\delta \leftarrow$ INITIALIZEDELTA( )
    while $i$ < maxIterations and $\delta$ > exitCriteria do
        $u_{i+1} \leftarrow u_i - \delta$
        for $k \in [1, 2, ..., N]$ do
            $\mathbf{v}_p, \dot{\mathbf{v}}_p \leftarrow$ GETPARTICLES( $t$, $\Upsilon_i(k)$, $\lambda$ )
            $\Lambda \leftarrow \mathbf{v}_p, \dot{\mathbf{v}}_p$
            $\Upsilon_{i+1}(k) \leftarrow$ GETSTATE($t$, robot, $\Lambda$, $u_{n+1}(k)$)
            $J_{i+1}(k) \leftarrow$ GETCOST( $\Upsilon_{target}$, $\Upsilon_{i+1}(k)$ )
        $\delta \leftarrow$ GETJACOBIAN( $J_i$, $J_{i+1}$, $u_i$, $u_{i+1}$ )
        $u_i \leftarrow u_{i+1}$
        $J_i \leftarrow J_{i+1}$
        $\Upsilon_i \leftarrow \Upsilon_{i+1}$
        $i \leftarrow i + 1$
    return $u_{i+1}$
```

vector serves as a first guess to the estimator to reduce optimization iterations, producing an added benefit of using PD control actions in smaller wave states (η), where net displacement is neglible. 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 GETSTATE which estimates the resulting predicted state, $\boldsymbol{\Upsilon}_{i+1}$. Lastly, $\boldsymbol{\Upsilon}_{i+1}$ is evaluated with respect to the target state, $\boldsymbol{\Upsilon}_{target}$ in GETCOST by way of the cost function (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.

VII. Results

One challenge when implementing MPC is determining a preferred prediction horizon. Section VII-A details how the horizon that balances solution accuracy with computation time is selected for this work. Section VII-B evaluates the controller with the chosen prediction horizon against a PD controller comparable to those employed in similar robotic tasks in the same wave climate. Finally, in Section VII-C, 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 $\mathbf{z} = -15\,\mathrm{m}$ 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.

A. Determination of Prediction Horizon

A desirable 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 0.2 s to 3.0 s. Table III shows total RMS errors and computation times per horizon.

TABLE III
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 III, two potential choices for prediction horizon are 0.8 and 1.0 s. Both of these offer low RMS error at reasonable computation times. For the results presented in this work, the chosen horizon time was 0.8 s 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 several times more calculation time than the chosen horizon of 0.8 s. This is likely 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.

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 faster optimization and programming techniques to further increase the robot horizon.

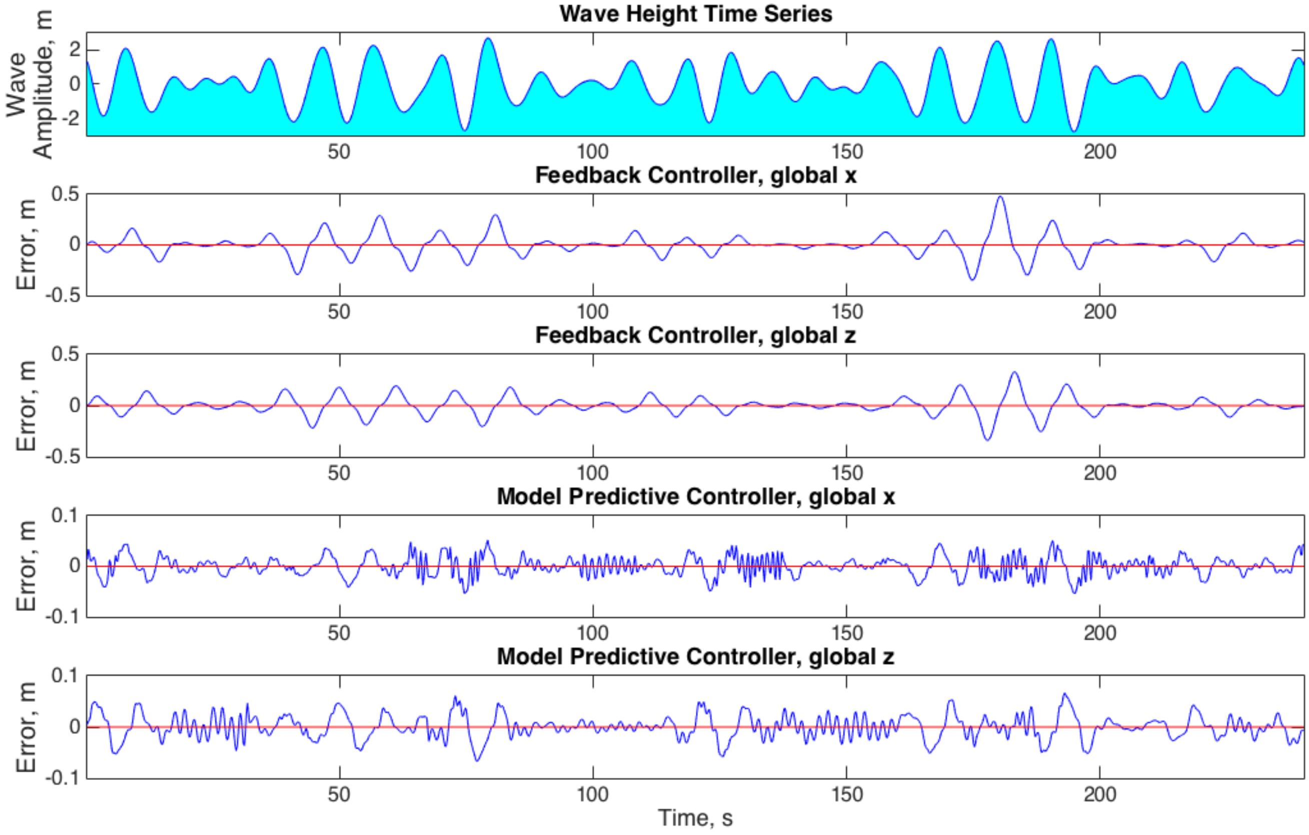

Fig. 4. Position error time series in global $\mathbf{x}$ and $\mathbf{z}$ coordinates when comparing a traditional feedback controller with a model predictive controller. As shown, MPC returns error values 74% lower than PD Control. RMS error values for these results are shown in Fig 5.

B. MPC Performance

Another set of simulations was run where a robot employing an 0.8 s horizon MPC was compared against one using a feedback (PD) controller, which was originally structured as a PID. The integral term was dropped, though, as it did not contribute to performance in the submerged environment. Both MPC and PD controllers were tested against the same wave field. Additionally, a free-floating, non-actuated robot disturbed by the same waves was simulated and compared for reference. Figure 4 details the positional errors for the PD and MPC controllers over the length of simulated time and Fig. 5 shows the RMS error for each case.

As shown in Fig. 4, MPC gives a 74% reduction in position error over PD 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, 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 [2], 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.

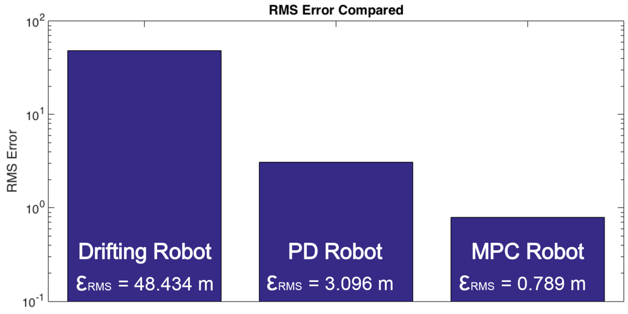

Fig. 5. RMS errors for the three cases in Section VII-B. The MPC robot showed a 74 % reduction in error compared to another using PD control.

Figure 6 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 (1). One concern about the inputs shown is that there are frequent changes in direction. This can lead to unrealistic results as thruster actuation time was not considered in this work. A more accurate model of thruster dynamics as well as an updated cost function to penalize large changes in thruster input are both avenues for future work.

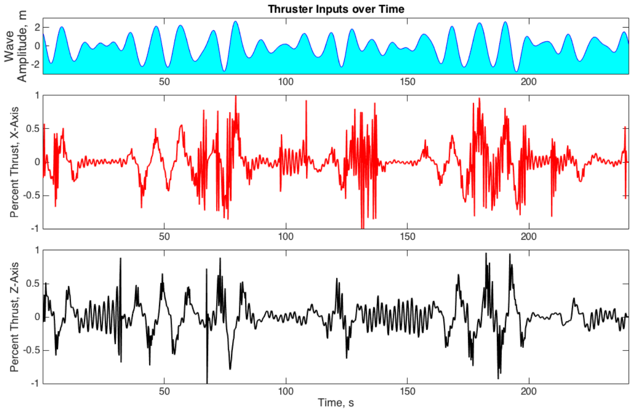

Fig. 6. MPC thruster inputs in global $\mathbf{x}$ and $\mathbf{z}$ coordinates over the simulated time series. The controller does not issue commands that saturate the thrusters for prolonged periods; however, it is prone to frequent direction changes.

C. 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) has lower resolution than the Doppler Velocity Log (DVL), thus it is more likely that the IMU would give inaccurate observations on 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 VII-B 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 [24]. 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.

TABLE IV

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

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

VIII. CONCLUSIONS AND FUTURE WORK

This paper presented a Model Predictive Control (MPC) approach to reducing underwater robot position error under the influence of shallow water under 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.

Results show that the most effective prediction horizon is 4 steps, or 0.8 s forward. This horizon reasonably balanced solution accuracy and computation time. Simulations show that a station-keeping robot disturbed by the same wave field can use our proposed MPC method to reduce its error by 74% over traditional PD control. Additional simulations were run where the MPC takes in noisy measurements of the wave field parameters. The algorithm was found to be resistant to sensor noise, showing a mean position error 44% lower than the deterministic feedback control case.

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. Also, 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. In addition, 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, other optimization techniques than gradient descent should be explored while more efficient programming techniques could be incorporated prior to field trials. This simulation was built in MATLAB to test the algorithm's feasibility and could be rewritten in Python or C++ to improve computational efficiency.

REFERENCES

[1] O. Phillips, *The Dynamics of the Upper Ocean*. Cambridge University Press, 1977.

[2] R. Dean and R. Dalrymple, *Water Wave Mechanics for Scientists and Engineers*, ser. Advanced Series on Ocean Engineering. World Scientific, 1991, vol. 2.

[3] A. Kim and R. M. Eustice, "Active visual SLAM for robotic area coverage: Theory and experiment," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 457–475, 2015.

[4] S. Chaves, A. Kim, and R. Eustice, "Opportunistic sampling-based planning for active visual SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014*, Sept 2014, pp. 3073–3080.

[5] R. Smith, Y. Chao, B. Jones, D. Caron, P. Li, and G. Sukhatme, "Trajectory design for autonomous underwater vehicles based on ocean model predictions for feature tracking," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, A. Howard, K. Iagnemma, and A. Kelly, Eds. Springer Berlin Heidelberg, 2010, vol. 62, pp. 263–273.

[6] 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, Jun 2014.

[7] C. Pêtrès, Y. Pailhas, P. Patrón, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 331–341, April 2007.

[8] L. Medagoda, J. 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*, May 2015, pp. 2091–2098.

[9] 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*, Sept 2013, pp. 1–10.

[10] T. Brekken, "On model predictive control for a point absorber wave energy converter," in *PowerTech, 2011 IEEE Trondheim*, June 2011, pp. 1–8.

[11] M. Richter, M. Magana, O. Sawodny, and T. Brekken, "Nonlinear model predictive control of a point absorber wave energy converter," *Sustainable Energy, IEEE Transactions on*, vol. 4, no. 1, pp. 118–126, Jan 2013.

[12] B. Ling and B. 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*, May 2015.

[13] M. Colby, E. 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] S. Heshmati-Alamdari, A. Eqtami, G. Karras, D. Dimarogonas, and K. Kyriakopoulos, "A self-triggered visual servoing model predictive control scheme for under-actuated underwater robotic vehicles," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 3826–3831.

[15] V. Huynh, M. Dunbabin, and R. 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.

[16] L. Medagoda and S. Williams, "Model predictive control of an autonomous underwater vehicle in an in situ estimated water current profile," in *OCEANS, 2012 - Yeosu*, May 2012, pp. 1–8.

[17] J. Riedel and A. Healey, "Shallow water station keeping of AUVs using multi-sensor fusion for wave disturbance prediction and compensation," in *OCEANS '98 Conference Proceedings*, vol. 2, Sep 1998, pp. 1064–1068 vol.2.

[18] J. 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.

[19] E. Camacho and C. Bordons, *Model Predictive Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer International Publishing, 2013.

[20] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill, 2009.

[21] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[22] 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*, May 2010, pp. 4791–4796.

[23] B. Bosma, T. Brekken, H. Ozkan-Haller, and S. Yim, "Wave energy converter modeling in the time domain: A design guide," in *Technologies for Sustainability (SusTech), 2013 1st IEEE Conference on*, Aug 2013, pp. 103–108.

[24] Y. Gōda, *Random Seas and Design of Maritime Structures*, ser. Advanced Series on Ocean Engineering. World Scientific, 2010, vol. 33.

[25] J. Falnes, *Ocean Waves and Oscillating Systems: Linear Interactions including Wave-Energy Extraction*. Cambridge University Press, 2002.

[26] R. Christ and R. Wernli Sr, *The ROV Manual: A User Guide for Remotely Operated Vehicles*. Butterworth-Heinemann, 2013.

[27] SolidWorks, "User's guide," *Dassault Systèmes SolidWorks Corp, Concord, Massachusetts*, 2012.

[28] ANSYS AQWA, "User's manual (ver. 14.0)," *ANSYS Incorporated, Canonsburg, PA*, 2012.

[29] J. Dormand and P. Prince, "A family of embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

[30] G. Hollinger, B. Englot, F. Hover, U. Mitra, and G. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," vol. 32, no. 1, pp. 3–18, 2013.

[31] R. Ballard, L. Stager, D. Master, D. Yoerger, D. Mindell, L. Whitcomb, H. Singh, and D. Piechota, "Iron age shipwrecks in deep water off ashkelon, israel," *American Journal of Archaeology*, vol. 106, no. 2, pp. 151–168, 2002.

[32] G. Hollinger, B. Englot, F. Hover, U. Mitra, and G. Sukhatme, "Uncertainty-driven view planning for underwater inspection," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 4884–4891.