

Model Predictive Control of a Hybrid Autonomous Underwater Vehicle with Experimental Verification

L. V. Steenson¹, S. R. Turnock¹, A. B. Phillips¹, C. Harris², M. E. Furlong³, E. Rogers⁴, L. Wang⁵

1. Faculty of Engineering and the Environment, University of Southampton, Southampton SO17 1BJ, UK

2. School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK

3. National Marine Facilities, National Oceanography Center, Southampton, UK

4. Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

5. School of Electrical and Computer Engineering, RMIT University, Melbourne, Australia

Abstract

In this work model predictive control is used to provide transit and hover capabilities for an autonomous underwater vehicle where the description of the system dynamics used include terms measured experimentally. The resulting controller manoeuvres the vehicle in the presence of constraints on the actuators and results obtained from the deployment of the vehicle in an inland lake for the study of the Zebra mussel, an invasive species, are also given.

1 Introduction

Unmanned underwater vehicles (UUVs) may be characterised as either remotely operated vehicles (ROVs) or autonomous underwater vehicles (AUVs). The former are typically tethered to a ship or surface structure, with the tether providing power and communication between the vehicle and operator. The use of the tether enables a pilot on the surface to manoeuvre the vehicle accurately and intelligently in order to complete a complex task such as repairing an oil well riser. The disadvantage of an ROV is that, due to the tether, the range of the vehicle is short and any motion of the ship is coupled with the ROV.

Typical AUVs are of torpedo shape [1], with four control surfaces and a propeller at the stern of the vehicle. They are primarily used for long range survey type operations where the vehicle essentially acts like a bus for onboard sensors to log data [2]. Such missions include bathymetry, CTD (conductivity, temperature and depth), or mine detection surveys among many other survey type missions. Due to the actuator set on-board a typical AUV, the vehicle has a minimum speed below which the vehicle loses control authority [3]. Such a vehicle is incapable of hovering and thus unable to conduct detailed inspection type missions.

The next generation of AUV will require both the ability to transit long distances, typical of standard AUVs, but also slow down to conduct detailed inspections on areas of interest, missions currently performed by ROVs. These requirements require a new approach to designing the vehicle and the on-board actuators. To date a small number of hover capable hybrid AUVs have been developed including: the hover capable REMUS 100 [4], ODYSSEY IV [5], and C-SCOUT [6]. To maintain the ability to transit long distances a low drag hull form is desirable, while for inspections a hover capability is required. In addition to the physical design challenges there are also implementation challenges, such as controlling the vehicle using a multitude of actuators.

All operational AUVs, and most ROVs, use closed-loop low-level control systems to manoeuvre the vehicle within its environment. These controllers are used to calculate suitable actuator settings so as to reduce the error between the current and desired state of the vehicle. There has been a substantial amount of research on the topic of low-level control systems for UUVs since the 1980's. This has included work based on: proportional-integral-derivative (PID) control [7, 8, 9], sliding mode control [10, 11] and model predictive control [12, 13, 14, 15, 16].

This paper gives new results on the design and experimental evaluation of model predictive control laws for a hover capable hybrid AUV. After development of the model, including some parameters obtained from experimental data, model predictive control is used as the controller. One deployment for an AUV of this type is the efficient mapping of large lakes to detect, or monitor, environmental predators where, in general, the most efficient route would be to have a fleet of vehicles under co-ordination. Such a system requires the operating characteristics of each vehicle to be known and in this paper the performance of the single AUV in a large lake to detect the presence of the Zebra mussel is evaluated. The next section begins with a description of the vehicle.

2 Vehicle Characteristics

Delphin2 AUV [17] shown in Figure 1 is a prototype vehicle designed for the development of control and navigation systems. The maximum depth for this vehicle is 50 m at a speed of 2 m/s, an estimated range of 20 km and an endurance of 8 hours [18].



Figure 1: *Delphin2*: an over-actuated hover capable AUV.

The AUV is torpedo shaped and over-actuated with four through-body tunnel thrusters (two vertical and two horizontal), four rear mounted independently controlled control surfaces in a cruciform arrangement, and a rear propeller. The term ‘over-actuated’ is used as the vehicle has more actuators (nine) than degrees of freedom (six), but this is only true when the vehicle is moving as the control surfaces will not produce any force whilst the vehicle is stationary (relative to fluid). A typical AUV equipped with only rear control surfaces has a minimum velocity below which the vehicle becomes uncontrollable. For the *Delphin2* AUV, the through-body tunnel thrusters are used below this critical speed to maintain vehicle control at low and zero forward speeds [3]. The thrusters are the dominant actuator set when operating between minus 0.3 m/s to plus 0.5 m/s forward speed.

The tunnel thrusters use a 250 W brushless DC rim driven motors of 70 mm internal diameter. The use of external coils maximizes the blade area in the tunnel producing up to 25 N of thrust. The four control surfaces have a mean chord of 0.104 m, mean span of 0.0815 m with a NACA 0014 section shape, the control surfaces are independently controlled to $\pm 30^\circ$ and the rear propeller is driven by a 50 W brushless DC motor. Torque to the control surfaces and propeller is transmitted through magnetic couplings. Communications between the computer and the actuators is via USB-Serial. Details of the electronics can be found in [17].

Depth is measured by a pressure transducer and a tilt compensated 3 axis digital compass measures heading, pitch and roll. An echo sounder gives the altitude above seabed and scanning sonar obstacle avoidance capability. Location on the surface is by GPS and there is a sensor for temperature for logging temperature with depth and location. Finally, two CCD colour cameras, forwards and downwards facing, are available for object detection and video survey. Physical details of the actuators and sensors employed can be found in [17].

The software onboard the vehicle is used for the autonomous operation of the vehicle including; artificial intelligence (AI), navigation, manoeuvring, sensor and actuator interfacing, and limited fault detection. Robotics Operating System (ROS), an open-source robotics platform [19], provides the underlying functionality of the software, which consists of several nodes and one library. Each node has a primary function, for example the compass sensor node reads data from the compass USB serial port and processes the data into usable information, before publishing it. The ROS platform provides the communication functionality that enables the nodes to publish information to topics. Nodes that require information, from another node, can then subscribe to these topics.

Nodes can be written in either the Python or C++ programming languages (other languages are currently being developed). Onboard this vehicle all the code was written using the high-level programming language Python. The reason for this is that Python enables rapid code development due to its relatively simple syntax, availability of open-source libraries and that the code does not require compiling. One disadvantage of using Python over C++ is that its computational efficiency is generally lower. Therefore to achieve equivalent performance using Python compared to C++ requires a more powerful computer that will, in general, use more electrical power. The hotel load of *Delphin2* at 50W is high for a vehicle of its size [20]. As this vehicle is designed as a research platform, electrical power consumption is not of primary concern and instead the rate of software development takes precedent. For commercial vehicles it would be advantageous to use C++ and a less powerful computer.

The software developed for this vehicle can be described as hierarchical with a mission planner at the highest layer and the actuator and sensor drivers at the lowest layer, Figure 2. The main components of the software structure are now described upwards from the lowest layer.

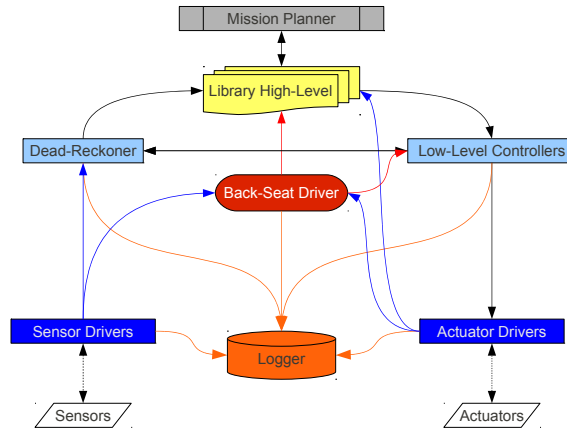


Figure 2: Flow-diagram illustrating the software architecture of the *Delphin2* AUV software.

Communication between the hardware and computer are handled by the sensor and actuator drivers. All of the hardware (with the exception of the frame-grabber) on *Delphin2* interfaces with the computer using the RS-232 serial protocol, using either a direct serial connection or a USB-serial adapter. Most of the sensor nodes only read information from the sensors and do not transmit. The actuator nodes transmit information to hardware, such as thruster demands, and also read information returned by the hardware, such as thruster speed feedback. The processed information read by the actuator and sensor nodes are published to topics specific to each node.

The second layer includes three main components; the dead-reckoner node, the low-level controller nodes, and the back-seat driver node. The dead-reckoner node computes navigational information such as (X,Y,Z) location and surge velocity. This is done by subscribing to information from the sensors and low-level controllers and then combining this information with a five DOF non-linear hydrodynamic model of the vehicle. The performance of the dead-reckoner is critical for navigation underwater when (X,Y) location information is not available (the GPS signal is available only whilst the AUV is on the surface).

There are typically two low-level controller nodes; a heading controller node, and a depth and speed controller node. As these controllers are the main focus of this research, the number of nodes varies throughout this work. The low-level controllers compute suitable actuator set-points so as to minimize the error between high-level demands (e.g. go to 2 metres depth) and the current vehicle states. Previous work with the *Delphin2* AUV has relied upon gain-scheduled proportional integral derivative (PID) controllers for manoeuvring the vehicle [21, 22]. These controllers are decoupled from each other, thus a controller exists for each degree of freedom that the operator wishes to control. Although these controllers have been reasonably successful, the lack of coupling between degrees of freedom and the reliance on fixed values, such as estimated buoyancy, can present problems when performing complex manoeuvres or if the system unexpectedly changes due for, example, to a loss or gain of buoyancy. The integral term within a PID controller has the ability to compensate for such changes but integral ‘wind-up’ during step changes in depth can result in undesirable overshoot.

The back-seat driver node continuously monitors sensor, actuator and mission information to ensure all the critical parameters are within predefined limits [23]. If any of these limits are exceeded, such as maximum depth, then the back-seat driver publishes an error flag. The library high-level (described in the next paragraph) and the low-level controllers both subscribe to the back-seat driver topic. In the event an error flag occurs, the low-level controllers immediately switch off the actuators (thrusters and rear propeller) and the library-high level informs the mission planner of the error so it can determine how to proceed.

The top layer will be described as two sub-layers; on the bottom sub-layer is the library high-level and on the top is the mission planner. The library high-level is a compilation of functions and system information that can be called by the mission planner and the purpose of the library high-level is to simplify and reduce the quantity of code in the mission planner, thus reducing the likelihood of a coding error.

The mission planner can then call functions, such as go to 2 metres depth, and the library high-level will publish the depth demand as well as the flags that enable the depth controller and actuator drivers. The mission planner contains states that handle tasks within the mission planner. The states are similar to nodes but are directly connected to the mission planner node. When each task finishes it returns one of three possible outcomes; succeeded, aborted or preempted. Succeeded means that the task completed successfully, aborted means it did not complete successfully (often due to a predefined time-out criterion being exceeded), and preempted means that the task has stopped due

to an error flag from the back-seat driver. The next task is defined by the outcome of the previous task. Future development of the mission planner will include more outcome options so as to enable a more reactive, and less linear, planner.

The logger node is not defined within a specific level. It subscribes to most of the available topics and writes the published information to individual files in the comma-separated values (CSV) format. These files are post-processed to analyse the vehicle performance.

The next section gives the model of the vehicle dynamics used for control design.

3 Modeling for Control

In this paper it is the lower level control that is considered, where there is a clear need to ensure that the AUV can achieve the commands sent from the higher level. At this level there are two main controllers; heading controller, in the form of a gain-scheduled PID algorithm, and a depth and speed controller for which Model Predictive Control (MPC) is used. It is this latter aspect that is considered in this paper and the first task is to build a model of the dynamics.

Most AUVs are designed to be positively buoyant (they float). Therefore to maintain depth an equal and opposite force to the vehicles buoyancy must be generated. Two modes of operation exist for *Delphin2*, hovering mode (using the tunnel thrusters) and flight-style mode (using the control surfaces). Flight style is the most common approach to depth control for AUVs, where the buoyancy is countered by downforce generated across the hull by pitching the vehicle nose down. Horizontal control surfaces (sternplanes) are used to control the vehicle pitch. The vehicle must be above the critical speed in order to generate sufficient pitching moment from the sternplanes to overcome the hydrostatic restoring moment. Details of the wind tunnel testing to measure lift, drag and pitching moments for different control surface and hull angles of attack can be found in [17]. The second mode of operation is hovering where the buoyancy is countered by thrust from the tunnel thrusters. Details of experiments to quantify the performance of the tunnel thrusters may be found in [24].

Following a similar approach to that adopted by Fossen [25] the depth and pitch model for control law design is based on linearisation about zero speed. For pitch and depth control, the AUV is modeled as two coupled second-order systems, see Figure 3. It is assumed that the pitch angles are small, less than 10° , the surge velocity is negligible and the buoyancy does not vary with depth. This gives the following pitch and depth models, respectively, with the nomenclature given in the appendix.

$$\dot{q}_v = -\frac{1}{I_y}[x_{Tvf}T_{vf} + x_{Tvr}T_{vr} - z_g W \sin \theta + \frac{1}{2}\rho V^{2/3}C_{Dq}|q_v|q_v] \quad (1)$$

$$q_v = \int_0^t \dot{q}_v dt, \theta = \int_0^t q_v dt \quad (2)$$

$$\dot{w}_v = \frac{1}{m_z}[T_{vf} \cos \theta + T_{vr} \cos \theta + (W - B) - \frac{1}{2}\rho V^{2/3}C_{Dw}|w_v|w_v] \quad (3)$$

$$w_v = \int_0^t \dot{w}_v dt, z = \int_0^t w_v dt \quad (4)$$

Equations (1) and (3) calculate the angular acceleration about the y (pitch) axis and linear acceleration on the z (depth) axis respectively. Both of these equations include non-linear terms that need to be linearised for use with the MPC algorithm as detailed below. For simplicity the damping terms will be described as quadratics despite the inclusion of the absolute value (used to maintain the correct sign).

The controller outputs are the force demands for each thruster (in Newtons) and to translate these force values into thruster speed set-points to send to the thruster controller, the inverse of the thrust equation is used, that is,

$$n = 60 \times \left[\frac{T_{demand}}{\rho K_T D^4} \right]^{0.5} \quad (5)$$

In this description the transient dynamics of the thrusters are not modelled for simplicity, nor the effects of the advance ratio [26].

To obtain a linearised model for control design, the trigonometric terms are replaced by their small angle approximations. Linearising the quadratic damping terms is more challenging. When the AUV has successfully arrived at the defined depth and pitch set-points, and is stable, the heave and pitch velocities are zero, thus the damping forces

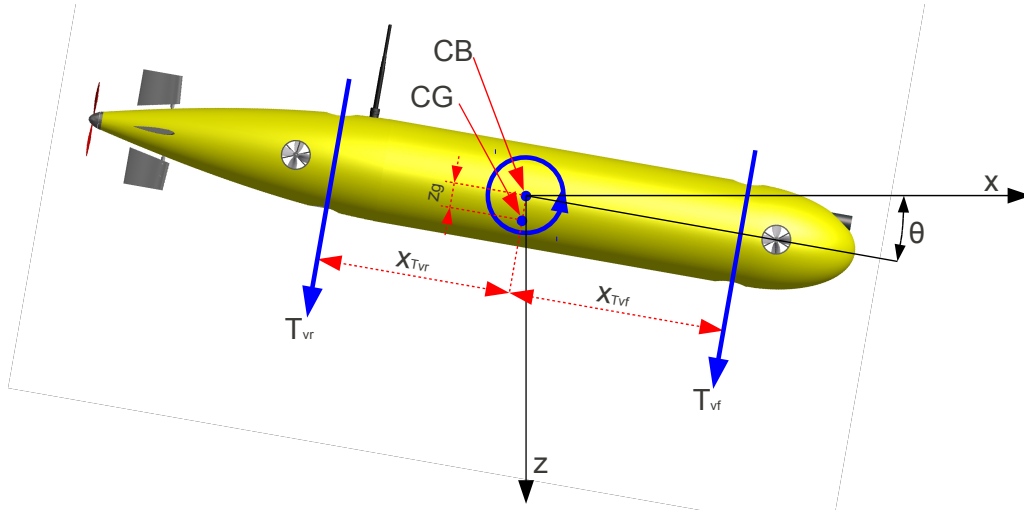


Figure 3: Illustration demonstrating the location of the thruster force vectors, moment arms, centre of gravity and buoyancy and the axis system. Note that the AUV in the illustration is at a pitch angle, θ , of -10° .

and moments are equal to zero. Linearising about zero heave and pitch velocities results in the damping coefficients also equalling zero, therefore eliminating damping from the model. Removing damping from the model would force the MPC algorithm to reduce system velocities using the system inputs (thrusters) rather than the natural dynamics, causing the controller to perform slowly and inefficiently. It is also intuitive that if the damping terms are estimated to be too high then the controller will overshoot the set-point.

The linear damping approximations need to be a compromise between low and high speed operation and in this work the linearised and non-linear damping terms are compared using the square of the normalised error:

$$(\text{Normalised error})^2 = \left[\frac{\text{non-linear term} - \text{linear term}}{\text{non-linear term}} \right]^2 \quad (6)$$

Moreover, sum of the squared normalised errors, from zero to the maximum velocities (estimated), is then minimised and the value of the linear damping coefficient that at the minimum is taken as the coefficient in this work. Figure 4 compares the linear and non-linear damping terms.

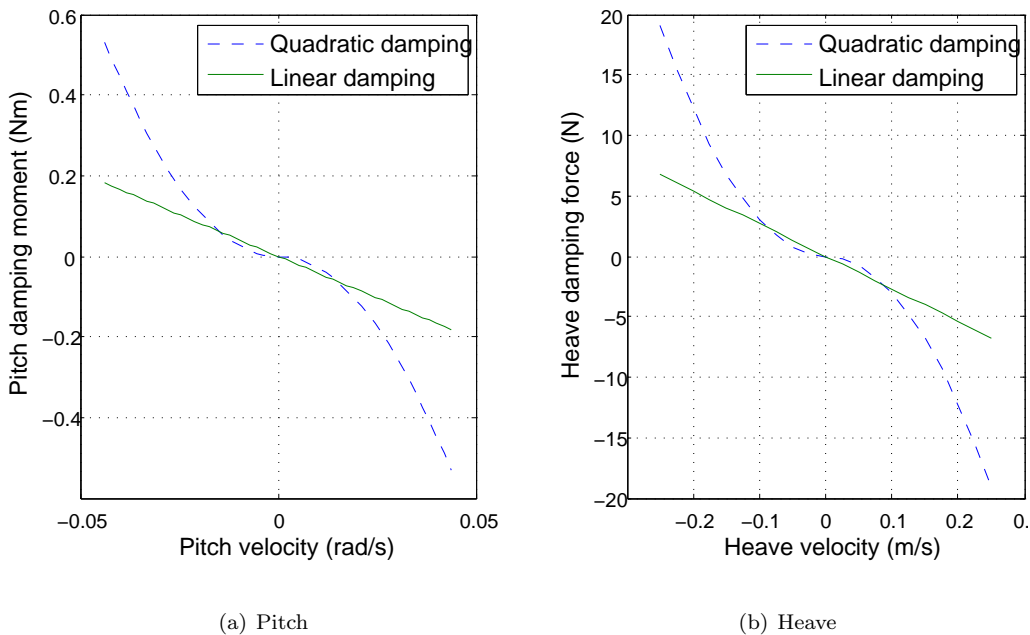


Figure 4: Comparison between the linear and non-linear damping forces and moments.

Combining equations (1) to (5) the AUV dynamics can be approximated by a time invariant state-space model

(assuming zero initial conditions):

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{\rho V^{2/3} k_z}{2m_z} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-z_q W}{I_y} & \frac{\rho V^{2/3} k_q}{2I_y} \end{bmatrix} \begin{bmatrix} z(t) \\ w_v(t) \\ \theta(t) \\ q_v(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{m_z} & \frac{1}{m_z} \\ 0 & 0 \\ \frac{-x_{Tvf}}{I_y} & \frac{-x_{Tvr}}{I_y} \end{bmatrix} \begin{bmatrix} T_{vf}(t) \\ T_{vr}(t) \end{bmatrix} \quad (7)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} z(t) \\ w_v(t) \\ \theta(t) \\ q_v(t) \end{bmatrix} \quad (8)$$

where k_z and k_q are the linear damping coefficients. The values of the parameters used in this state-space model are given in Table 1.

Parameter	Value
ρ	$1000kg/m^3$
V	$0.08m^3$
m_z	$167.5kg$
I_y	$70kg.m^2$
W	$540N$
k_z	8.62×10^{-2}
k_q	1.31×10^{-2}
x_{Tvf}	$0.55m$
x_{Tvr}	-0.49

Table 1. Linear state-space model parameters.

In the next section the MPC design is detailed.

4 Control Design

The use of MPC in this application is due its ability to handle multiple degrees of freedom using a model of the system with constraints, where the control algorithm calculates the optimal control inputs over a finite number of time steps in order to reduce the error. The constraints are specified within the controller, thus enabling the optimal control to be calculated within the stated constraints. Since the vehicle is positively buoyant, the vertical thruster limits are set such that the thrusters only rotate in one direction and the thrusters do not stop while submerged. The reasoning for avoiding zero speed is due to the nonlinear dead-band associated with a motor operating around zero rpm.

A full derivation of the algorithm used is given in [27]. Here the steps required to design and implement the model predictive controller will be described. First the continuous-time state-space model is sampled to result in a discrete state-space model with state, input and output matrices denoted by A_d, B_d and C_d respectively. The MPC algorithm embeds an integrator to deal with model inaccuracies and ensures zero steady-state error for set-point following. This is done by replacing the system input vector u by Δu , constructed by taking the difference between the current and previous state and input vectors and likewise for the state vector, that is, in terms of Δu and Δx

$$\Delta x(k) = x(k) - x(k-1), \Delta u(k) = u(k) - u(k-1) \quad (9)$$

where k denotes the sampling instant. The state-space model used for MPC design is

$$\begin{bmatrix} \Delta x(k+1) \\ y(k+1) \end{bmatrix} = A \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} + B \Delta u(k) \\ y(k) = C \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} \quad (10)$$

where

$$A = \begin{bmatrix} A_d & 0 \\ C_d A_d & I \end{bmatrix}, B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix}, C = [0 \quad I] \quad (11)$$

and for the remainder of this paper the null and identity matrices with compatible dimensions are denoted by 0 and I respectively.

In this model it is $\Delta u(k)$ that is optimized by the predictive control algorithm. In the steady-state, all entries in $\Delta x_m(k)$ are zero and the steady-state values of the output vector $y(k)$ will be taken as the set-point signals. Therefore, with the inclusion of an integrator in the predictive controller algorithm, the steady-state values are not required, leading to simplification at the implementation stage.

The predictive controller is designed using the receding horizon control principle, where future state vector is calculated for a prediction horizon of N_p samples for a future control trajectory of N_c samples where $N_c \leq N_p$. The prediction will be denoted as starting from sample number $k_i > 0$. Let N_c and N_p denote the control and prediction horizons, respectively, where $N_c \leq N_p$. Also introduce

$$\begin{aligned} U &= [\Delta u^T(k) \quad \Delta u^T(k+1) \quad \dots \quad \Delta u^T(k+N_c-1)]^T \\ X(k) &= [x^T(k+1|k) \quad x^T(k+2|k) \quad \dots \quad x^T(k+N_p|k)]^T \end{aligned} \quad (12)$$

Then the state space model (10) can be used to recursively compute the future state vectors and output vectors, or in a more compact form,

$$Y = FX(k) + \Phi U \quad (13)$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \quad \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

and

$$Y = [y^T(k+1|k) \quad \dots \quad y^T(k+N_p|k)]^T \quad (14)$$

Introduce the set-point or reference vector of length N_p as

$$R_s^T = [I \quad I \quad \dots \quad I] r(k) \quad (15)$$

where $r(k)$ is the reference vector at sample instant k . Then the cost function for MPC design is

$$J = (R_s - Y)^T (R_s - Y) + U^T R U \quad (16)$$

where R is a symmetric positive-definite matrix to be selected. Routine analysis [27] now gives the minimizing control in the absence of constraints as

$$U = (\Phi^T \Phi + R)^{-1} (\Phi^T R_s - \Phi^T F x(k)) \quad (17)$$

where the required matrix inverse is assumed to exist.

Using receding horizon control, only the entries in ΔU corresponding to $\Delta u(k_i)$ are used and the actual control signal applied to the plant is computed using

$$u(k) = u(k-1) + \Delta u(k) \quad (18)$$

where both the current optimal control $\Delta u(k)$ and past value $u(k-1)$ are used. Since the current and past control signals have the same steady-state value, if the first sample of the control signal is taken as the actual plant input signal before the closed-loop controller is in operation, the computation of the control signal using (18) leads the actual control signal for direct implementation. Hence the control signal has included its steady-state value, which is also part of the simplification in the implementation of the predictive controller.

Magnitude and rate constraints on the control signals can be achieved by minimizing the cost function J of (16) in real-time with constraints imposed. In particular, control amplitude constraints for the state-space model at sampling instant k and in this application these can be written in the form

$$u^{min} \leq u(k) \leq u^{max}, \quad (19)$$

where u^{min} and u^{max} are $p \times 1$ data vectors containing the required lower and upper limits of the control amplitude amplitudes allowed. If constraints on $\Delta u(k) = u(k) - u(k-1)$ are also required then in this MPC setting these take the form

$$\Delta u^{min} \leq \Delta u(k) \leq \Delta u^{max}, \quad (20)$$

where Δu^{min} and Δu^{max} correspond to the minimum and maximum changes in the thruster set-points, respectively.

In this work constraints are applied to the system inputs, on both the rate of change and absolute limits. The rate of change of thrust demand should be chosen to be lower than the maximum rate that the thruster can change its speed (this is an approximation, as thrust is proportional to thruster speed squared). If the controller demands larger

changes of thrust than the thruster is capable of then the thruster dynamics will effectively act like a low-pass filter on the real system which may prove detrimental to controller performance. For the design in this paper, the rate of change of set-points for each thruster are set at ± 2 N per sample. The maximum absolute thrust value is set at ± 10 N, corresponding to a thruster speed of approximately 1800 rpm. The thrusters are capable of driving at higher speeds, producing more thrust, but it is desirable to reduce the peak thrust so as to reduce peak electrical power consumption. The minimum absolute thrust value is set at $+0.7$ N, corresponding to a thruster speed of approximately 500 rpm. This is the minimum thruster speed below which the thruster dynamics become more non-linear due to the performance of the thruster controller (hardware) and the motor dead-band around zero thruster speed. Avoiding these non-linearities is the reason for setting the limit above zero speed. Both thruster constraints are positive and the assumption is that the buoyancy force of the vehicle is at least twice the minimum thrust constraint since otherwise the vehicle will be unstable in depth.

The predictive controller is obtained by minimizing the cost function (16) subject to (19) and (20) by direct application of quadratic programming algorithms in, for example, [27] and the relevant references cited in this text. In this application, implementation of the control law requires direct measurement of all entries in $[\Delta x_m(k)^T y^T(k)]^T$ at each sampling instant k . The difficulty is that the state vector component $\Delta x_m(k)$ is not measurable but could be approximated using $\Delta x_m(k) = x_m(k) - x_m(k-1)$. If, however, there is a significant amount of noise corruption in $x_m(k)$, the computed difference $\Delta x_m(k)$ will amplify the effects of noise. Hence an observer is used to estimate $\Delta x_m(k)$ as

$$\hat{x}(k+1) = A\hat{x}(k) + B\Delta u(k) + K_{ob}(y(k) - C\hat{x}(k)) \quad (21)$$

where the observer gain matrix K_{ob} is chosen such that the closed-loop observer error system matrix $A - K_{ob}C$ has all eigenvalues strictly inside the unit circle of the complex plane, where in this work K_{ob} is designed using discrete-linear quadratic regulator theory with weighting matrices Q_{ob} and R_{ob} . If $R_{ob} = r_{ob}I$, the effects of measurement noise in $y(k)$ can be reduced by selecting a large value of r_{ob} . The initial state vector of the observer is set to be the zero vector if the output vector $y(k)$ is zero. Otherwise, the initial $\hat{\Delta x}_m(0)$ is set to zero and the initial $\hat{y}(0)$ is set to the actual measurement of $y(0)$.

The initial development of the controller is MATLAB based and the controller is then translated into the Python programming language within the *Delphin2* control software as a separate node. To ensure that no mistakes have been made in translating the code, the controller node is coupled with a simulation node. This simulation node is identical to the dead-reckoner node, see below, but without sensor feedback. The Python simulation node and MATLAB simulator are identical, therefore the controller is assumed to be working and correct when the simulation results from the Python code matches the results from the previous MATLAB simulations.

Figure 5 is a schematic of the basic flow of information into and out of the MPC controller. Controller demands, e.g. depth demand, are published from the mission planner. The MPC controller then subscribes to the different demands for which it is designed to control. Feedback is published by the dead-reckoner node and the controller subscribes to the dead-reckoner topic. Publishers are set up within each controller to enable the controller to publish the necessary demands to the actuators.

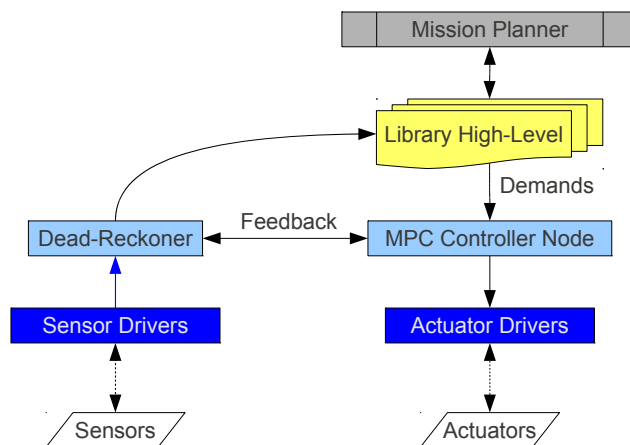


Figure 5: Flow diagram of information into and out of the model predictive controller in the *Delphin2* AUV.

5 Results

Prior to the experimental tests, an in-depth simulation exercise was undertaken. These simulations were performed in MATLAB using the non-linear model of the system in which realistic levels of measurement noise were added to the depth and pitch feedback signals, further details of these simulations is provided in [17]. From the simulation results three controller parameter sets were found corresponding to an aggressive, conservative and balanced system response, see Table 1, previous simulations showed that the control horizon, N_c , appears to have the least significant effect on system performance, with stability only compromised with low values of N_c at higher N_p values.

Controller set	N_p	N_c	r_w
Aggressive	60	8	0.5
Balanced	80	8	4.0
Conservative	100	8	8.0

Table 1: Linear state-space model parameters

The simulation studies were also used to determine the observer gain matrix by examining the effect of the observer gain scalar, r_o , on system performance. Although system performance does vary with r_o , it did not show any system instability despite a wide range of values tested. For the experimental results given in the rest of this section $r_o = 50$ was used.

Figure 6 gives the simulation results when the AUV is diving from 0 to 1 depth whilst maintaining a zero pitch angle. The depth converges quickly with minimal overshoot and pitch does oscillate slightly about zero degrees. Note that the noise still effects the controller performance but within acceptable limits.

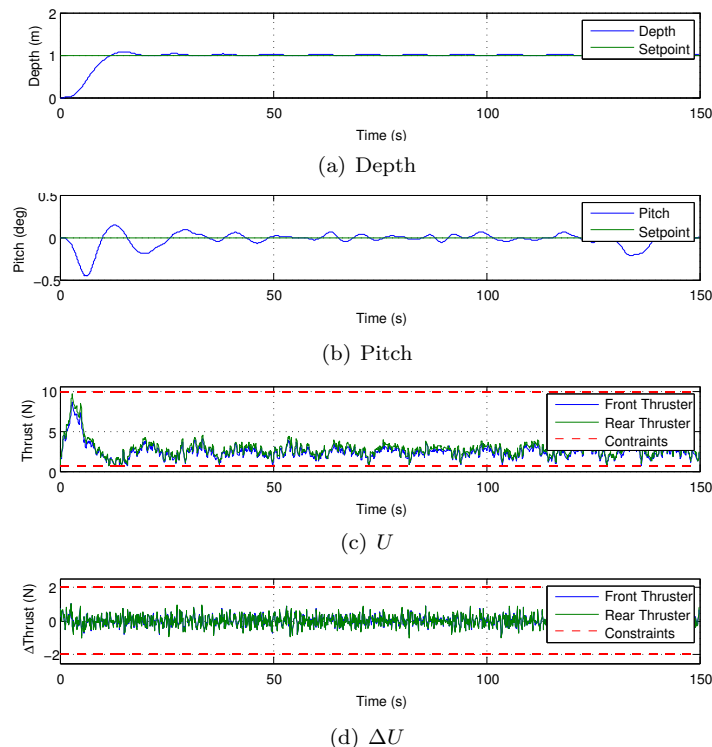


Figure 6: Simulation results of the MPC controller using the observer with an r_o value of 10.

To test the MPC controller on the *Delphin2* AUV, the MATLAB code was translated into the Python programming language and then integrated within the vehicles control software. The experimental results were obtained in an acoustic tank measuring $8.0 \times 8.0 \times 4$ m deep. Each test consists of three depth set-points of 1.0 m, 3.0 m and 2.0 m with the pitch set-point set at 0.0° for all the tests. Once the AUV has stayed within ± 0.2 m of the depth set-point for 60 s it moves on to the next depth set-point.

Figures 7, 8 and 9, respectively, show the experimental results for the aggressive, balanced and conservative tuning parameter sets of Table 1, where the depth, pitch, u and Δu signals are plotted against time.

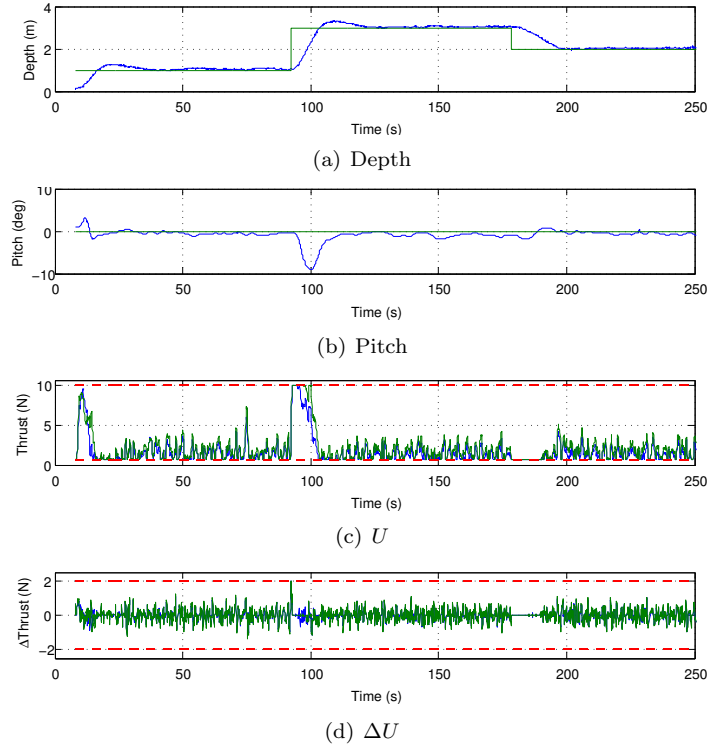


Figure 7: Experimental results of the MPC controller with depth set-points of 1 m, 3 m and 2 m. The pitch set-point remains fixed at 0.0° . Controller parameters: $N_p = 60$, $N_c = 8$, $r_w = 0.5$, $r_o = 50$.

Overall, these results provide clear evidence that the model predictive controller with constraints is capable of high quality performance in this application area, where all three controller parameter sets provide stable control of both depth and pitch. Further research is required to maximize this and the following discussion highlights areas where such research should be directed.

The first aspect is the overshoot of the depth demand for all three parameter sets. This trend results in lower overshoot values for the larger values r_w with a minimum overshoot of 0.15 m with a r_w value of 8. This means that at the first depth demand the more conservative model predictive controller provides a lower overshoot. For the second depth demand of 3.0 m this trend is reversed, with the largest overshoot of 0.382 m using a r_w value of 8. The cause of this reversal in trend is due to the magnitude of the step change in depth demand, where the lower r_w values cause the system to behave more aggressively with rapid changes in u values, whilst the higher N_p values along with higher r_w values causes the controller to perform more conservatively. For the first step change of 1.0 m, the more aggressive controller with a r_w value of 0.5 along with an N_p value of 60 causes the system to accelerate quickly, resulting in a large heave velocity which in turn results in it overshooting the depth demand. In contrast the more conservative controller parameters do not generate high heave velocities and hence corresponding overshoots are lower. This variation in controller 'aggressiveness' leads to the reversal in overshoot trends when the depth step change is greater. In this case all the controller parameters cause high heave velocities, but as the vehicle approaches the depth set-point the conservative controller parameters are slower to reduce the thrust magnitude and so their heave velocities decrease slower than the more aggressive controller parameters resulting in a greater overshoot.

It is also noted that for the second step change, the constraints on the magnitude of thruster movement and its incremental movement have become active (i.e. used maximum u and Δu) when the aggressive predictive controller is used (see Figure 7). As a result, the settling time is about 25 s, which is about half of the response time in comparison with the cases when the balanced and conservative predictive controllers are used (see Figures 8 and 9). This means that by deploying constrained control, the AUV system can provide the fastest response to the depth set-point signal while maintaining all operational constraints. Furthermore, if the AUV encountered a large disturbance, then by deploying constrained control, maximum thruster movement may help the system quickly reject the disturbance and

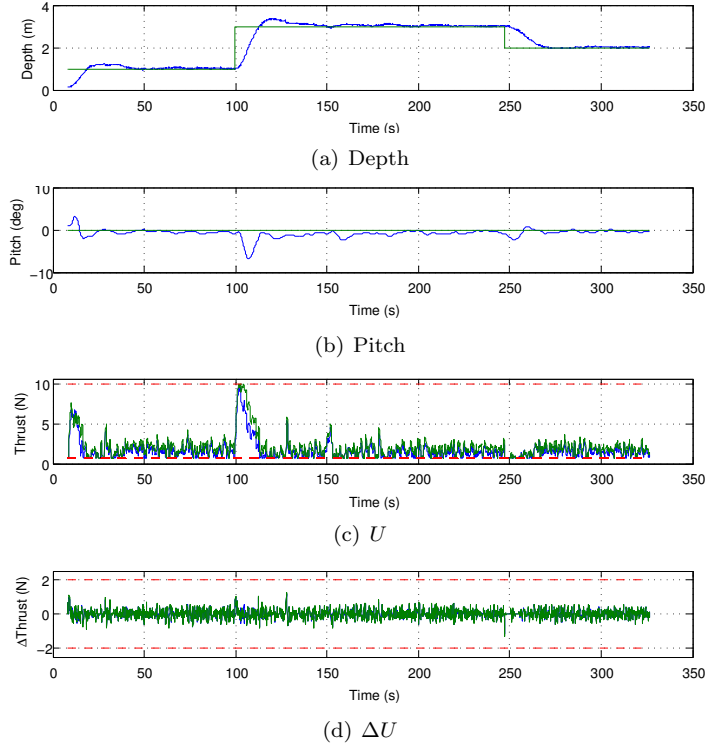


Figure 8: Experimental results of the MPC controller with depth set-points of 1 m, 3 m and 2 m. The pitch set-point remains fixed at 0.0° . Controller parameters: $Np = 80$, $Nc = 8$, $r_w = 4.0$, $r_o = 50$.

maintain closed-loop stability. The advantage of using constrained control in disturbance rejection will be tested experimentally in the future report.

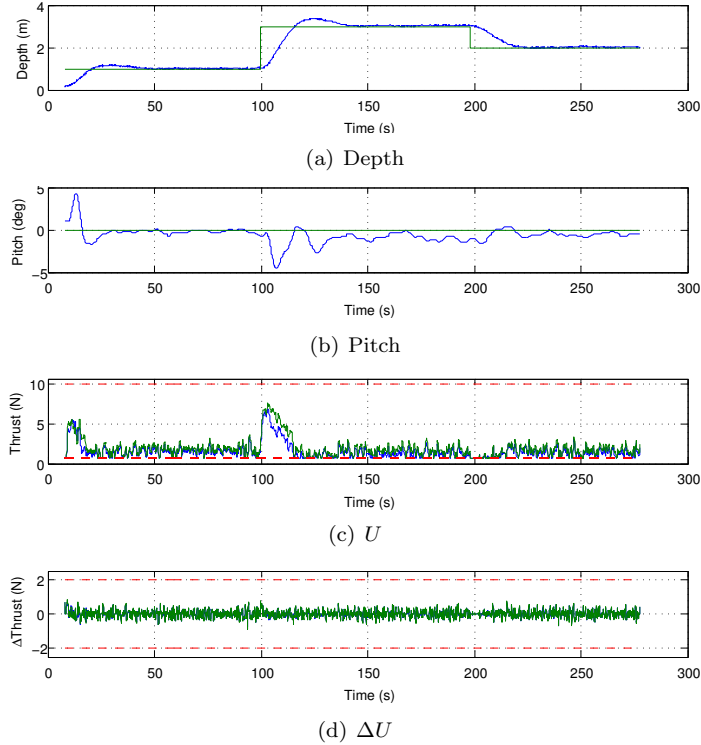


Figure 9: Experimental results of the MPC controller with depth set-points of 1 m, 3 m and 2 m. The pitch set-point remains fixed at 0.0° . Controller parameters: $Np = 100$, $Nc = 8$, $r_w = 12.0$, $r_o = 50$.

6 Lake Test

The *Delphin2* AUV has been used to collect video data of Lower Lough Erne to study the Zebra mussel population in the Lough, consisting of a video survey of the Lough bottom at Carrickreagh Bay. Zebra mussels are small striped freshwater shellfish native to Russia, since 1996 this species has been invading Lough Erne [28] eco-system, juvenile mussels (1-3mm long) attached to vessels are easily transported to new locations. This alien non-native species forms large colonies that attach to any hard surface. The presence of zebra mussels has had a significant impact on the ecology of Lough Erne, resulting in a decrease in chlorophyll and an increase in water clarity [29], yet their distribution over the lake bed has not been widely studied. Three transects: 1) bearing 320° for 230 metres, 2) bearing 0° for 730 m and 3) bearing 30° for 300 m were performed, see Figure 10. The AUV was programmed to switch on its downwards and forwards facing cameras then track the Lough bottom at 0.75 m altitude at 0.5 m forward speed, using a downwards facing echo-sounder. Due to the high turbidity of the water in the lake the vehicle needed to be within one metre of the lake bottom to record usable video footage. The visibility in the lake was an unknown prior to deployment of the vehicle, with hindlight alternative optical technologies and enhanced lighting may have provided more detailed images.

In the first transect the edge of the bank, approximately 30 metres out, from Carrickreagh jetty was followed for 230 m on a bearing of 320° . Figure 11 gives the depth track on the vehicle about the Lough bottom, and on the bottom subplot temperature against distance. Note that as the vehicle dives from the surface at the start of the mission the temperature drops quickly from 16.1°C to about 13.8°C . At approximately 85 metres into the transect the AUV hit the bottom, this was a slow speed crash so not of concern. For these transects the vehicle had no measurements of the lake bed topology in front of the vehicle. Hence, the vehicle had insufficient warning of the upcoming feature. Transect 2 is the longest of the three and involved the vehicle driving north up to a water depth of 7.7 metres. Figure 12 shows the vehicle position and temperature during this transect, again the temperature drops with increasing water depth. The final of the three transects. This appears to have quite a narrow deep section in the middle of the transect. Unfortunately, see Figure 13 the vehicle did not dive quick enough to track this valley fully and then struggled on the upwards phase of the mission.

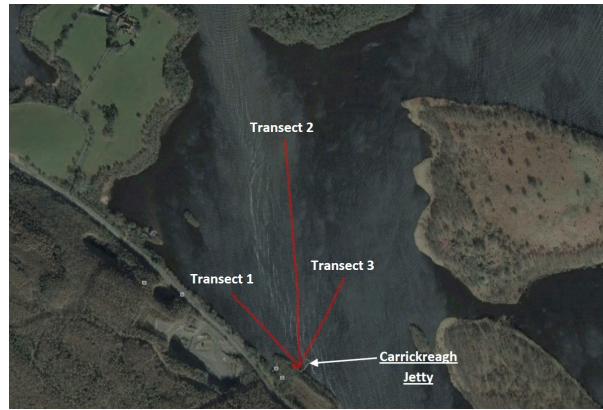
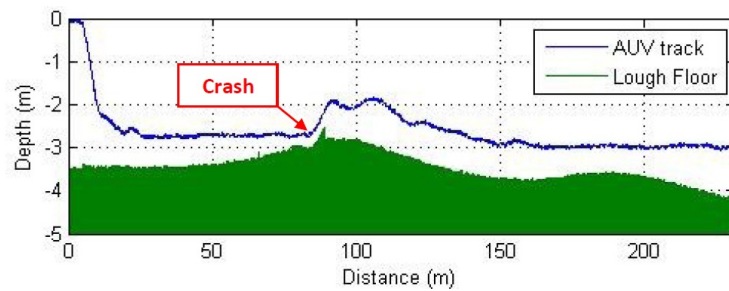
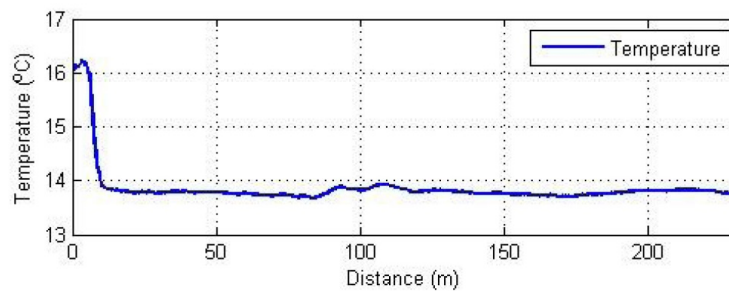


Figure 10: Carrickreagh region overlaid with the vehicles track coordinates for the three transects (image source : Google Earth [30])

By examining the video footage the extent of the zebra mussel population can be assessed. Zebra mussels colonise hard surfaces which they attach themselves to using an organ called a bysall. This is reflected in the transect footage. Each transect was initiated from close proximity to the Carrickreagh jetty, the underwater structure of the jetty was heavily colonised below a water depth of 1m, see Figure 14. In addition live mussels formed smaller colonies on the scattered shells of dead individuals which had fallen from the jetty structure. The centre of Carrickreagh bay is characterised by a mainly silty lake bed, unsuitable for Zebra mussels. However, small colonies were observed on small rocks or other hard detritus, see Figure 15.

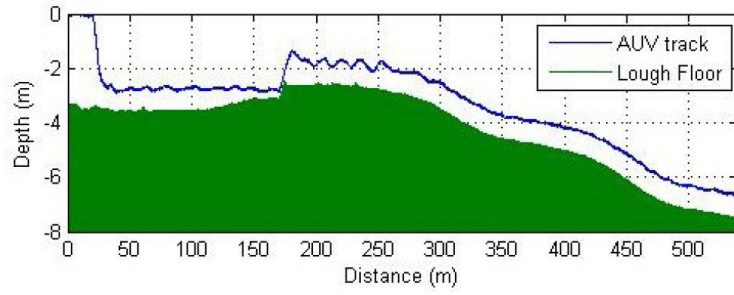


(a) Position

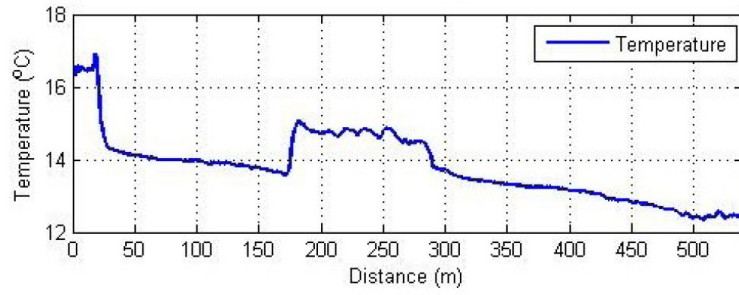


(b) Temperature

Figure 11: Vehicle position and temperature track for transect 1.

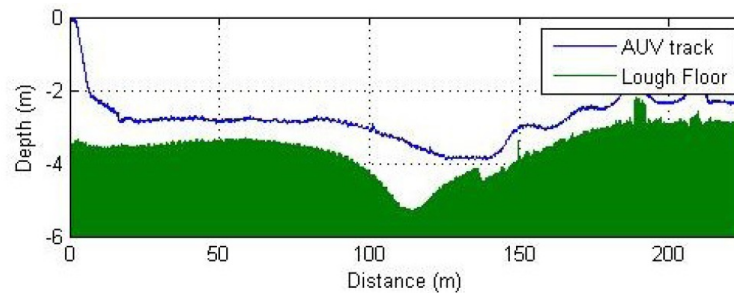


(a) Position

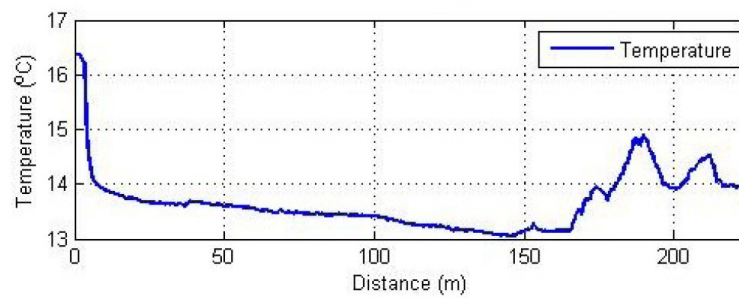


(b) Temperature

Figure 12: Vehicle position and temperature track for transect 2.



(a) Position



(b) Temperature

Figure 13: Vehicle position and temperature track for transect 3.



Figure 14: Zebra mussel colony on the Carrickreagh jetty, image taken while operating *Delphin2* as an ROV.

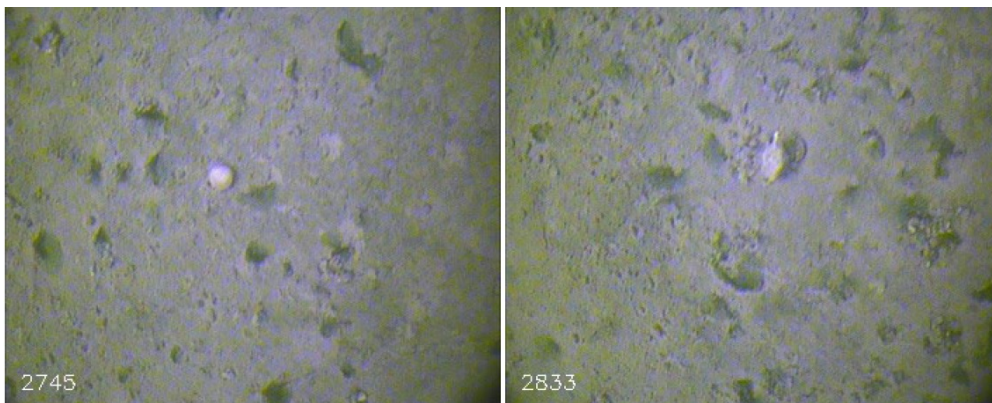


Figure 15: Example images from the downwards facing camera from transect 2.

7 Conclusions

This paper presents a model predictive control strategy for low speed depth control of an over actuated hover capable AUV. Results of the MPC controller are presented for both laboratory tests in an acoustic tank at zero forward speed and field experiments at slow forward speed performing bottom following. The performance of this controller showed significant promise in controlling the pitch, depth (or altitude) and speed of the vehicle. The performance of the MPC control strategy was more than sufficient to allow a high quality video survey of a lake bed, maintaining an altitude of 0.75m from the bottom (7.7 m from the surface), which was critical due to high turbidity in the lake. From these results it is believed that the MPC algorithm could be successfully implemented, with minimal effort, on most AUVs that operate at low speeds using thrusters.

This work, with the supporting lake test results, confirms that MPC is applicable AUV control and there is much further development possible. Issues requiring further research from an MPC standpoint include: robustness and the domain of attraction and hence estimation of the safe operating region, as well as comparisons with classical control strategies.

References

- [1] P. Stevenson, M. Furlong, and D. Dormer. Auv shapes - combining the practical and hydrodynamic considerations. In *OCEANS 2007 - Europe*, pages 1–6, 2007.
- [2] S. McPhail. Autosub6000: a deep diving long range AUV. *Journal of Bionic Engineering*, 6:55–62, 2009.
- [3] B. Burcher and L. Rydill. *Concepts in Submarine Design*. Cambridge Oceans Technology Series, 1994.
- [4] G.E. Packard, R. Stokey, R. Christenson, F. Jaffre, M. Purcell, and R. Littlefield. Hull inspection and confined area search capabilities of remus autonomous underwater vehicle. In *OCEANS 2010*, pages 1–4, 2010.
- [5] S. Desset, R. Damus, F. Hover, J. Morash, and V. Polidoro. Closer to deep underwater science with *odysseyiv* class hovering autonomous underwater vehicle (hauv). In *Oceans 2005 - Europe*, volume 2, pages 758–762 Vol. 2, 2005.
- [6] T.L. Curtis, D. Perrault, C. Williams, and N. Bose. C-scout: a general-purpose *auv* for systems research. In *Underwater Technology, 2000. UT 00. Proceedings of the 2000 International Symposium on*, pages 73–77, 2000.
- [7] B. Allen, R. Stokey, T. Austin, N. Forrester, R. Goldsborough, M. Purcell, and C. von Alt. Remus: a small, low cost auv; system description, field trials and performance results. In *OCEANS '97. MTS/IEEE Conference Proceedings*, volume 2, pages 994–1000 vol.2, 1997.
- [8] S.D. McPhail and M. Pebody. Autosub-1. a distributed approach to navigation and control of an autonomous underwater vehicle. In *Electronic Engineering in Oceanography, 1997. Technology Transfer from Research to Industry., Seventh International Conference on*, pages 16–22, 1997.
- [9] M. Santhakumar and T. Asokan. A self-tuning proportional-integral-derivative controller for an autonomous underwater vehicle, based on taguchi method. *Journal of Computer Science*, 6(8):862–871, 2010.
- [10] I. Boiko, L. Fridman, A. Pisano, and E. Usai. Analysis of chattering in systems with second-order sliding modes. *Automatic Control, IEEE Transactions on*, 52(11):2085–2102, 2007.
- [11] A.J. Healey and D. Lienard. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 18(3):327–339, 1993.
- [12] W Naeem. Model predictive control of an autonomous underwater vehicle. In *Proceedings of UKACC'2002 Postgraduate Symposium*, pages 19–23, 2002.
- [13] 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*, pages 1–8, 2012.
- [14] Q. Truong, L. Wang, and P. Gawthrop. Intermittent model predictive control of an autonomous underwater vehicle. In *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*, pages 1–6, 2006.
- [15] G.J. Sutton and R.R. Bitmead. Performance and computational implementation of nonlinear model predictive control on a submarine. In Frank Allgwer and Alex Zheng, editors, *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*, pages 461–472. Birkhuser Basel, 2000.

- [16] A. Budiyo. Model predictive control for autonomous underwater vehicle. *Indian Journal of Geo-Marine Sciences (IJMS)*, 40(2):191–199, 2011.
- [17] L. V. Steenson. *Experimentally Verified Model Predictive Control of a Hover-Capable AUV*. PhD thesis, University of Southampton, 2013.
- [18] A B Phillips, L. Steenson, C. Harris, E. Rogers, S. R. Turnock, and M. Furlong. Delphin2: An over actuated autonomous underwater vehicle for manoeuvring research. *The Transactions of the Royal Institution of Naval Architects, Part A International Journal of Maritime Engineering*, in Press, 2013.
- [19] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and Andrew Y. Ng. ROS: an open-source robot operating system. In *Proceedings of the ICRA Workshop on Open Source Software*, 2009.
- [20] A.B. Phillips, M. Haroutunian, S.K. Man, A.J. Murphy, S.W. Boyd, J.I.R. Blake, and G. Griffiths. Nature in engineering for monitoring the oceans: comparison of the energetic costs of marine animals and auvs. In G.N. Roberts and R. Sutton, editors, *Further advances in unmanned marine vehicles*, pages 373–405. The Institution of Engineering and Technology, 2012.
- [21] L. V. Steenson, A. B. Phillips, E. Rogers, M. E. Furlong, and S. R. Turnock. Preliminary results of a hover capable AUV attempting transitional flight. In *Unmanned Untethered Submersible Technology (UUST)*, 2011.
- [22] L. V. Steenson, A. B. Phillips, E. Rogers, M. E. Furlong, and S. R. Turnock. Control of an AUV from thruster actuated hover to control surface actuated flight. In *Specialists Meeting AVT-189/RSM-028 Assessment of Stability and Control Prediction Methods ofr NATO Air & Sea Vehicles*, 2011.
- [23] D.P. Eickstedt and S.R. Sideleau. The backseat control architecture for autonomous robotic vehicles: A case study with the iver2 auv. In *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pages 1–8, 2009.
- [24] L. V. Steenson, A. B. Phillips, E. Rogers, M. E. Furlong, and S. R. Turnock. Control of an AUV from thruster actuated hover to control surface actuated flight. In *Second International Symposium on Marine Propulsors*, 2011.
- [25] T.I. Fossen. *Guidance and Control of Ocean Vehicles*. Wiley-Blackwell, 1994.
- [26] A. Palmer, G.E. Hearn, and P Stevenson. Modelling tunnel thrusters for autonomous underwater vehicles. In *Navigation, Guidance and Control of Underwater Vehicles (NGCUV'08)*, 2008.
- [27] L. Wang. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer Verlag, 2010.
- [28] R. S. Rosell, C. M. Maguire, and T. K. McCarthy. First reported settlement of Zebra Mussels *Dreissena polymorpha* in the Erne System, Co. Fermanagh, Northern Ireland. *Biology and Environment: Proceedings of the Royal Irish Academy*, 98B(3):191–193, 1998.
- [29] R. S. Rosell, C. M. Maguire, and T. K. McCarthy. Ecological change in Lough Erne: influence of catchment changes and species invasions. *Freshwater Forum*, 24(1):38–58, 2005.
- [30] Google Earth. *Carrickreagh, Lower Lough Erne*. [Online]. Available from: <http://earth.google.com/>. [Accessed 3rd July 2012].

Appendix

B	Buoyancy (N)
C_{Dw}	Coefficient due to heave velocity
C_{Dq}	Coefficient due to pitch velocity
I_y	Inertia term for the y axis ($kg.m^2$)
\dot{q}_v	Angular acceleration about the y axis (pitch) (rad/s^2)
T_{vf}	Thrust from the front vertical thruster (N)
T_{vr}	Thrust from the rear vertical thruster (N)
V	Volume (m^3)
W	Weight of the vehicle (kg)
w_v	Heave velocity (m/s)
q_v	Pitch angular velocity (rad/s)
\dot{w}_v	Heave acceleration (m/s^2)
x_{Tvf}	Distance along the x axis between the centre of gravity and the front vertical thruster (m)
x_{Tvr}	Distance along the x axis between the centre of gravity and the rear vertical thruster (m)
z_g	Vertical distance between the vertical centre of gravity and vertical centre of buoyancy (m)
θ	Pitch angle (rad)