Experimental Verification of a Depth Controller using Model Predictive Control with Constraints onboard a Thruster Actuated AUV

Leo V. Steenson * **Alexander B. Phillips** * **Eric Rogers** ** **Maaten E. Furlong** *** **Stephen R. Turnock** *

* *FSI Research Group, University of Southampton, Hampshire, UK (Tel: 023 8059 2488; e-mail: l.v.steenson@soton.ac.uk).*
** *ISIS Research Group, University of Southampton, Hampshire, UK (Tel: 023 8059 2197; e-mail: etar@ecs.soton.ac.uk)*
*** *National Marine Facilities, National Oceanography Centre, Hampshire, UK (Tel: 023 80596376, e-mail: m.furlong@noc.soton.ac.uk)*

Abstract: In this work a depth and pitch controller for an autonomous underwater vehicle (AUV) is developed. This controller uses the model predictive control method to manoeuvre the vehicle whilst operating within the defined constraints of the AUV actuators. Experimental results are given for the AUV performing a step change in depth whilst maintaining zero pitch.

Keywords: Autonomous underwater vehicle, AUV, depth and pitch control, model predictive control, MPC

1. INTRODUCTION

There exists two types of unmanned underwater vehicle (UUV); remotely operated vehicles (ROVs), and autonomous underwater vehicles (AUVs). ROVs 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.

AUVs are typically of torpedo shape, with four control surfaces and a propeller at the stern of the vehicle. These vehicles are used for long range survey type operations where the vehicle essentially acts like a bus for onboard sensors to log data (McPhail [2009]). 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 becomes unstable (Burcher and Rydill [1994]). This results in a vehicle that is incapable of hovering and thus unable to conduct detailed inspection type missions.

The next generation of AUV will have the ability to transit long distances, typical of standard AUVs, but also slow down to a hover in order to conduct inspections on areas of interest, typical of ROVs. In order to do this requires a new approach to designing the vehicle and the on-board actuators. Along with the physical design challenges are implementation challenges such as controlling the vehicle using a multitude of actuators.

In this paper a depth and pitch controller is developed using the model predictive control method. A model of the AUV, that will be used within the controller design, is given. The model predictive control method and implementation is described. Experiments have been conducted in an acoustic tank, measuring 8 x 8 x 5 metres, at the University of Southampton. All the tests are the same, with a step change in depth from the surface to 1 metre whilst maintaining 0^o, and controller parameters varied to tune the controller. The experimental results of the depth and pitch controller are given and evaluated by the depth overshoot, depth settling time and final pitch value.

1.1 *Delphin2 AUV*

The Delphin2 AUV (Fig. 1) (Phillips et al. [2010]) is a prototype vehicle designed for the development of AUV control and navigation methods, a brief specification is given in Appendix A. The AUV is torpedo shaped and over-actuated with four through-body tunnel thrusters, four independently controlled control surfaces, and a rear propeller. The term 'over-actuated' is used as the vehicle has more actuators (nine) than degrees of freedom (six), however 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).

As mentioned earlier, a typical AUV has a minimum velocity below which the vehicle becomes unstable. With the Delphin2 AUV, the through-body tunnel thrusters are used below this critical speed to maintain vehicle

Fig. 1. Delphin2 AUV: Over-actuated hover capable AUV

stability. The thrusters are the dominant actuator set when operating between minus 0.3 ms^{-1} to plus 0.5 ms^{-1} forward speed.

To date the Delphin2 AUV has relied upon gain-scheduled proportional integral derivative (PID) controllers for manoeuvring the vehicle, (Steenson et al. [2011a]) and (Steenson et al. [2011b]). 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 (loss or gain of buoyancy).

2. AUV MODEL

For this work the AUV will be modelled in two degrees of freedom; pitch (θ) and depth (Z), and with two inputs; the front and rear vertical through-body tunnel thrusters. A complete representation of the vehicle physics would result in a highly nonlinear model (Fossen [1994]). As the control method in this work relies on a linear model, first a simplified nonlinear model is development and then a linear approximation of the nonlinear model is created. This linear model is used within the model predictive control design. The nomenclature for all the equations can be found in Appendix B.

2.1 Nonlinear Depth and Pitch Model

Pitch Model

$$\ddot{\theta} = [T_{0act}xT_0 + T_{1act}xT_1 - BGmg\sin\theta - \frac{1}{2}\rho C_{D\theta}A_\theta\dot{\theta}^2]/I \quad (1)$$

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

Depth Model

$$\ddot{Z} = [T_{0act}\cos\theta + T_{1act}\cos\theta + B - \frac{1}{2}\rho C_{DZ}A_Z\dot{Z}^2]/I \quad (3)$$

$$\dot{Z} = \int_0^t \ddot{Z}\, dt,\ Z = \int_0^t \dot{Z}\, dt \quad (4)$$

2.2 Thruster Dynamics

It has been shown that if the actuator dynamics are not taken into account when designing a controller for an UUV then performance of the controller can be dominated by the actuator dynamics (Yoerger et al. [1990]). For this work the thruster dynamics have been simplified as a first order system from thrust demand to actual thrust. Although the thrusters are not exactly a first order system, the approximation should be sufficient to enable the controller to handle the inherent lag between demand and generated thrust (Steenson et al. [2011a]). The controller output is therefore a thrust demand (N). As the motor controller used by the thrusters accepts a speed demand, the inverse of the quasi-steady thrust equation (7) is used to define the speed set-point that will produce the thrust demand (Steenson et al. [2011c]).

$$n = [T_{act}/\rho K_T D^4]^{0.5} \quad (5)$$

2.3 Linear Depth and Pitch Model

Both the pitch and depth models, (1) and (4), have nonlinear terms due to hydrodynamic damping when the vehicle is moving. These terms will be approximated by a linear term about zero. There are also cosine functions in the nonlinear model, these have been approximated to 1. This should be adequate for vehicle operation $\pm 20^o$ pitch angle, with a maximum error of $\pm 6\%$.

Combining (1) - (4), linear hydrodynamic damping terms and the thruster dynamics, we can represent the AUV with the state space model:

$$\begin{bmatrix} \dot{Z} \\ \ddot{Z} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{T}_{0act} \\ \dot{T}_{1act} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{-k_d}{M} & 0 & 0 & \frac{1}{M} & \frac{1}{M} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-G}{I} & \frac{-k_p}{I} & \frac{xT_0}{I} & \frac{xT_1}{I} \\ 0 & 0 & 0 & 0 & \frac{-1}{T_T} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{T_T} \end{bmatrix} \begin{bmatrix} Z \\ \dot{Z} \\ \theta \\ \dot{\theta} \\ T_{0act} \\ T_{1act} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_T} & 0 \\ 0 & \frac{1}{T_T} \end{bmatrix} \begin{bmatrix} T_{0demand} \\ T_{1demand} \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} Z \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Z \\ \dot{Z} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (7)$$

3. MODEL PREDICTIVE CONTROL

Model predictive control (MPC) has been chosen for this AUV due its ability to handle multiple degrees of freedom using a model of the system with constraints. MPC 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.
In this work the constraints are set on the minimum and maximum thruster forces. Due to the vehicle being positively buoyancy (it floats), the thruster limits are set such that the thrusters only rotate in one direction and the motors do not stop. The reasoning for avoiding zero speed is due to the nonlinear dead-band associated with a motor starting stationary.
In this study two controller parameters are adjusted in order to tune the controller. The performance of the controller will be evaluated by the overshoot (percentage of step change), settling time (time to within 10% of set-point) and the pitch angle at the end of the test.

3.1 MPC Algorithm

A full derivation of the algorithm can be found in (Wang [2010]) and the relevant references therein. Here the steps required to create and run the model predictive controller will be described.

First the continuous state-space model of the system is converted into a discrete state-space model using the time step chosen for the controller giving the dynamics, input and output matrices A_d, B_d and C_d respectively. This discrete state-space model is then embedded with an integrator to create an augmented model of the system giving:

$$A = \begin{bmatrix} A_d & o_d^T \\ C_d A_d & I \end{bmatrix}; B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix}; C = [o_d I] \qquad (8)$$

Using the prediction horizon, N_p, and the control horizon, N_c, we create the matrices;

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{Np} \end{bmatrix} \qquad (9)$$

$$\phi = \begin{bmatrix} CB & 0 & 0 & ... & 0 \\ CAB & CB & 0 & ... & 0 \\ CA^2B & CAB & CB & ... & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & ... & CA^{N_p-N_c}B \end{bmatrix} \qquad (10)$$

The optimal change in control signal, ΔU, for the future N_c control inputs can be calculated by solving the quadratic programming problem;

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}(\Phi^T\bar{R}_s r(k_i) - \Phi^T F x(k_i)) \qquad (11)$$

In this work, $\phi^T\bar{R}_s$ is taken as the last n_{in} columns of F, where n_{in} is the number of system inputs. $\bar{R}$ is used as a tuning parameter and for this work is an identity matrix multiplied by a scaler, and $r(k_i)$ is the depth and pitch set-points at sample k_i.

The optimal solution is then evaluated to check whether any of the control inputs have violated the specified constraints. If a constraint has been violated then the Hildreth programming procedure (HPP) is used to find a correction term, λ^*, such that an optimal solution can be found within the constraints. Therefore the optimal within constraints is:

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}(\Phi^T\bar{R}_s r(k_i) - \Phi^T F x(k_i))... \\ ... - (\Phi^T\Phi + \bar{R})^{-1}M^T\lambda^* \qquad (12)$$

The first n_{in} values in the optimal ΔU set are taken and added to the previous control input U (initialised at zero), such that the control signal is an integral of the optimal solution ΔU.

3.2 Implementation

The MPC controller has been integrated into the Delphin2 control software as a separate node. The code is written in the computer programming language Python within the robotics framework ROS (Quigley et al. [2009]). Processing efficiency of this code is not optimised however has proven adequate for this work. For better computational efficiency a low level programming language such as C or C++ could be used. Depth and pitch are measured using a pressure transducer and digital compass sensor respectively. Linux is used as the computers operating system on-board a dual core mini-itx board.

4. RESULTS

The results presented here are for the Delphin2 AUV diving, using the vertical tunnel thrusters only, from the surface to 1m depth whilst maintaining 0^o pitch. The system is a multi-input multi-output system where the inputs are the front and rear vertical thrusters and the outputs are depth and pitch. The system has been discretized using a Δt value of 0.1 seconds. Two parameters in the controller have been varied in order to tune the controller; Np and $\bar{R}$. Twenty-two tests were conducted at four different Np values. The tests were conducted in an acoustic tank measuring 8 x 8 x 5 metres deep.

4.1 Overview

Fig. 2 presents the results from a test with Np and $\bar{R}$ equal to 50 and $1.25I$ respectively. As can be seen in the top plot, the depth does not start from 0 m, this is because the offset of the transducer is not correctly set. This does not effect the performance of the controller as the offset remains the same for all the tests.

The performance of the depth controller is good, with a smooth transition from the surface to 1 m depth. Maximum measured overshoot for the test is measured at 7.6% however it is expected that if the depth signal was post-processed with a filter, to remove the noise, then the real overshoot would be much less. The pitch control in this case resembles most of the test cases in that there is a strong disturbance to the pitch whilst diving but the

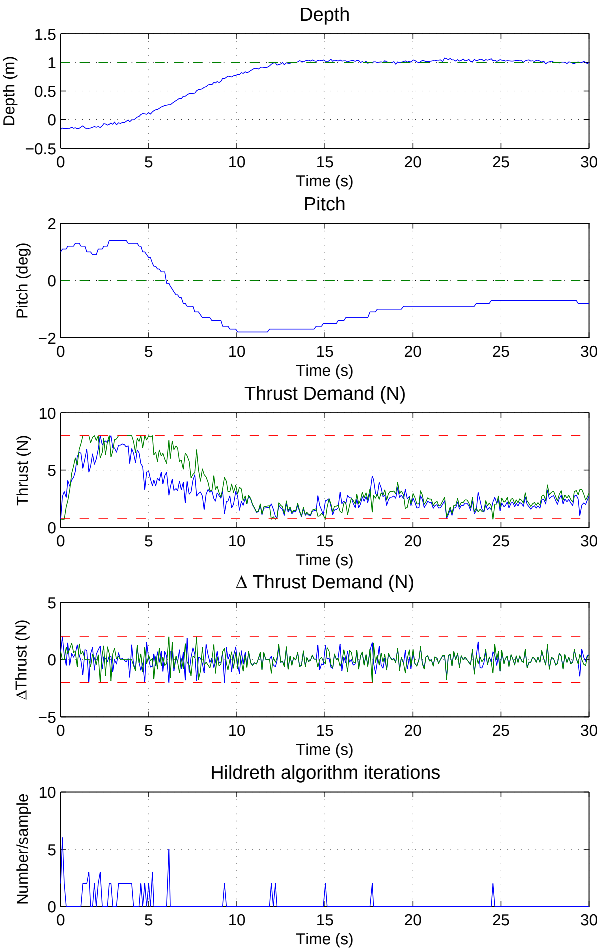

Fig. 2. Step change in depth with pitch demand fixed at 0^o. From top, 1st plot; Dashed line = depth set-point, Solid line = measured depth. 2nd plot; Dashed line = pitch set-point, solid line = measured set-point. 3rd plot; Dash line = maximum and minimum thrust constraints, solid lines = thrust demands. 4th plot; Dash line = maximum and minimum change in thrust constraints, solid lines = change in thrust demands, 5th plot; Number of iterations of the HPP on each control loop.

controller recovers to near the desired set-point. The pitch did not converge to the desired 0^o in any of the tests.

The third and forth plots of Fig. 2 are of the thrust demands and the change in thrust (Δthrust) demands from the controller respectively. Both plots include the constraints imposed on the controller for the maximum and minimum thrust and Δthrust values. The fifth plot shows the iterations of the HPP on each sample. Whenever the value is greater than zero then the constraints have been violated and the HPP will iterate towards a correction term to find an optimal solution within the constraints.

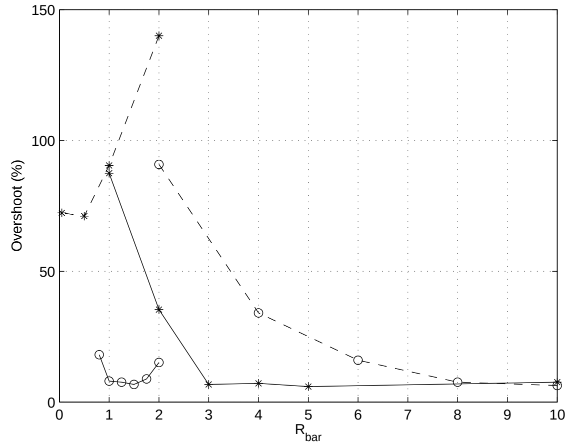

Fig. 3. Percentage overshoot of the set-point for four Np values; '*' dashed line = Np of 25, 'o' solid line = Np of 50, '*' solid line = Np of 75, and 'o' dashed line = Np of 100

4.2 Overshoot

Overshoot is of great importance for a depth controller as the AUV could collide with the seabed resulting in a potentially catastrophic failure. In Fig. 3 the percentage overshoot is plotted against $\bar{R}$ for the four Np values; 25, 50, 75 and 100. It should be known that none of the $\bar{R}$ values with the Np value of 25 provided a stable controller, instead it oscillated about the set-point with a large amplitude. For the higher Np values stable $\bar{R}$ values were found resulting in overshoots of below 10%. For Np of 50 the range of $\bar{R}$ values providing a stable system is narrow in comparison to the higher Np values of 75 and 100. All of the stable results have an overshoot value of approximately 5%, it is expected that if the depth signal was filtered to remove noise then the overshoot values would be lower.

4.3 Settling time

Second to overshoot as regards importance for any controller is the ability to quickly reach the set-point that the user, or high level controller, demands. If a system is designed to be too stable, so as to minimise the risk of overshoot, then the system may never reach the desired set-point within a suitable time. In Fig. 4 the settling time (the time taken from the start until the AUV reaches and stays within ±10% of the set-point) is plotted against $\bar{R}$ for three Np values; 50, 75 and 100. Data from tests that never converged to a stable value within the ±10% of the set-point have been emitted, thus all of the data from the tests with Np of 25 are not included.
From Fig. 4 it can be stated that the shorter the length of the prediction horizon, Np, the faster the settling time. However, as the data from the tests of Np equal to 25 have been emitted due to the AUV being unstable, it can also be said that the desire for a fast settling time must involve consideration for both stability and overshoot.

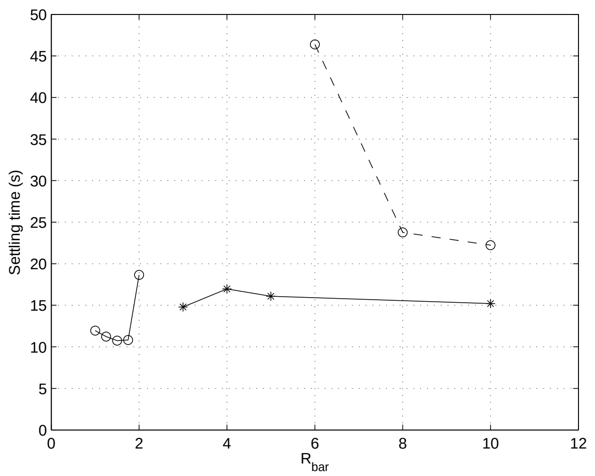

Fig. 4. Settling time to within ±10% of set-point for three Np values; 'o' solid line = Np of 50, '*' solid line = Np of 75, and 'o' dashed line = Np of 100

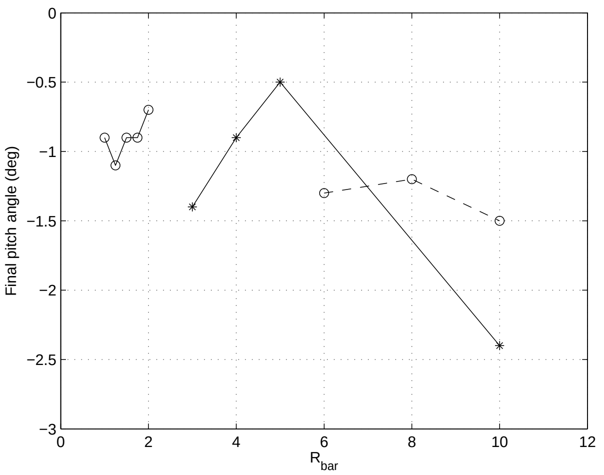

Fig. 5. Final pitch value at the end of the test for three Np values; 'o' solid line = Np of $50I$, '*' solid line = Np of $75I$, and 'o' dashed line = Np of $100I$

4.4 Pitch control

In this work the pitch controller has been used to maintain zero degrees of pitch. All of the tests, that are stable in depth control, have resulted in a slow convergence towards zero pitch. Fig. 5 plots the final pitch value at the end of each test against $\bar{R}$ for three Np values. The data presented in Fig. 5 are for stable test data. In general the lower the Np value the better convergence towards zero pitch angle. This corresponds with the settling time shown in Fig. 4.

5. CONCLUSIONS

A depth and pitch controller has been developed for the Delphin2 AUV using the model predictive control method. The controller has been implemented on the AUV and experimental results given. The controller has been tuned using two controller variables. The depth controller has been shown to work very well with minimal overshoot and a fast settling time. It has been shown that a higher Np value is more likely to provide a stable controller as opposed to a low Np value. For a faster response the opposite is true, with the fastest stable responses being at the Np value of 50 before becoming unstable at Np of 25. For the tuning of a new controller it is expected that a high value of Np should be used to increase the likelihood of stability, then the Np value can be reduced incrementally to improve response time.

The pitch controller has only been tested by maintaining zero degrees pitch. Although the pitch does show convergence to zero, none of the tests converge to the set-point. The settling time for pitch is therefore significantly slower than that of the depth controller. The separate tuning of the depth and pitch response needs to be investigated.

For these tests the linear model has proven sufficiently accurate for the MPC method to work. It would be expected that if MPC was to be used on the vehicle whilst moving forward that a linear model would not be accurate enough due to the dominance of the velocity squared terms in the nonlinear model.

REFERENCES

R. Burcher and L. Rydill. *Concepts in submarine design.* Cambridge Ocean Technology Series, 1994.

Thor I. Fossen. *Guidance and Control of Ocean Vehicles.* Wiley, 1994.

Stephen McPhail. Autosub6000: A deep diving long range auv. *Journal of Bionic Engineering*, 6:55–62, 2009.

A. B. Phillips, L. V. Steenson, J. Liu, C. A. Harris, S. M. Sharkh, E. Rogers, S. R. Turnock, and M. E. Furlong. Delphin2: An over actuated autonomous underwater vehicle for manoeuvring research. *International Journal of Maritime Engineering*, 1:In Press, 2010.

Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

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)*, 2011a.

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 for NATO Air & Sea Vehicles*, October 2011b.

L.V. Steenson, A.B. Phillips, E. Rogers, M.E. Furlong, and S.R. Turnock. The performance of vertical tunnel thrusters on an autonmous underwater vehicle operating near the free surface in waves. In *Second International Symposium on Marine Propulsors*, Humburg, Germany, June 2011c. SMP11.

Liuping Wang. *Model Predictive Control System Design and Implementation Using MATLAB.* Springer, 2010.

D.R. Yoerger, J.G. Cooke, and J.-J.E. Slotine. The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design. *Oceanic Engineering, IEEE Journal of*, 15(3):167 –178, jul 1990. ISSN 0364-9059. doi: 10.1109/48.107145.

Appendix A. DELPHIN2

Depth	= 50 m (maximum)
Speed	= 2 ms^{-1}
Range	= 20 km (estimated)
Endurance	= 8 hours

Appendix B. NOMENCLATURE

ΔU	= Array of Δthrust demands
θ	= Pitch angle (radians)
ρ	= Density of water ($kg.m^{-3}$)
A_θ	= Area of vehicle about pitch axis (m^2)
A_Z	= Area of vehicle about Z axis (m^2)
B	= Vehicle buoyancy (N)
BG	= Distance between centre of buoyancy and centre of gravity (m)
C_{DZ}	= Nonlinear drag coefficient for pitch axis
$C_{D\theta}$	= Nonlinear drag coefficient for pitch axis
D	= Thruster diameter (m)
g	= Acceleration due to gravity (ms^{-2})
G	= Derivative of restoring moment with respect to pitch ($\frac{Nm}{rad}$),
I	= Vehicle inertia about pitch including added inertia (($kg.m^2$))
k_d	= Linear drag coefficient (dimensional) for Z axis ($\frac{N}{ms^{-1}}$),
k_p	= Linear drag coefficient (dimensional) for pitch axis ($\frac{Nm}{rad.s^{-1}}$),
K_T	= Thrust coefficient
m	= Vehicle mass excluding added mass (kg)
M	= Vehicle mass including added mass (kg)
n	= Thruster speed (rps)
n_{in}	= Number of system inputs
Nc	= Control horizon (number of samples
Np	= Prediction horizon (number of samples)
$\bar{R}$	= Controller gain matrix
T_{0act}	= Produced thrust from front thruster (N)
T_{1act}	= Produced thrust from rear thruster (N)
$T_{0demand}$	= Thrust demand for front thruster (N)
$T_{1demand}$	= Thrust demand for rear thruster (N)
T_T	= Thruster time constant (s)
xT_0	= Distance from centre of gravity to front thruster (m)
xT_1	= Distance from centre of gravity to rear thruster (m)
Z	= Depth (m)