

Depth and Trim Control of an AUV

Damon M. Vizl¹

University of New South Wales at the Australian Defence Force Academy

The use of Autonomous Underwater Vehicles in the field of underwater research and maintenance has driven the development of a prototype vessel at the University of New South Wales Australian Defence Force Academy campus. The aim of this design project was to investigate an alternative method of trim and depth control in order to minimize power consumption and address additional recommendations outlined in Nicholas Gover's paper *Optimum Design of a Small Underwater Vehicle: Small White Design and Build*. The design methodology followed an iterative approach in order to enhance the use of feedback in the development of software packages and physical design. The method chosen to control trim and depth was to alter the buoyancy of the vessel, this method allowed for minimal weight calibration, decreased power consumption and internalized the mechanism to increase water tight integrity.

Contents

I. Introduction	2
II. Background	2
III. Aims	3
IV. Design and Development	5
V. Testing and Results	8
VI. Conclusions	9
VII. Recommendations	9
VIII. Acknowledgments	10
IX. References	11

Nomenclature

Terms

AUV :	Autonomous Underwater Vehicle
UNSW:	University of New South Wales
ADFA:	Australian Defence Force Academy
SEIT:	School of Engineering and Information Technology
PID:	Proportional, Integral, Derivative Controller

Variables

m_s	=	mass of water in the syringes
g	=	gravity [9.81 ms^{-1}]
ρ_w	=	density of water [1024 kgm^{-1}]
A	=	cross-sectional area of the vessel [
V_d	=	volume of water displaced by the vessel
C_d	=	drag coefficient of a cylinder [0.7]
F_b	=	buoyancy force [$\rho_w V_d g$]

¹ SBLT, School of Engineering & Information Technology. ZEIT4500

I. Introduction

The earth's surface is dominated by water, around 70%. Although the depths of the ocean are vastly uncharted, there is a strong need for shallow water surveillance as data and power cables are often run under lakes and dams. Using Divers for these tasks can be dangerous and time consuming, this is where Autonomous Underwater Vehicles (AUVs) come in handy. To be able to follow a cable and monitor it for damages is one task that is intended for AUVs. The SEIT department of UNSW at ADFA have been developing an AUV over the last three years. Last year the task to design the AUV fell to Nicholas Gover, from his design and testing, a number of recommendations arose. These wereⁱ;

- **Minimize Water Ingress:** The method chosen for trim and depth control by Gover was to install multiple thrusters that drove the vessel up or down to control depth. This method presented a problem as the hull breaches made the vessel very hard to seal.
 - **How it was addressed:** In order to minimize water ingress these exterior thrusters were removed from the design completely, instead a series of internalized syringes were placed in the vessel, both fore and aft. This allowed only 2 small hull breaches (one for water intake/outtake and one for pressure sensing) that were successfully sealed.
- **Weight Control:** The method of weight control used by Gover was a series of led weights put in the nose and tail sections of the AUV. This was necessary to create enough weight to counter the buoyancy force, but did not allow any dynamic weight adjustment. Additionally the weight of the vessel had to be precisely calibrated to ensure neutral buoyancy.
 - **How it was addressed:** In order to achieve neutral buoyancy, led weights were still used to get an approximate weight. From there the buoyancy was fine tuned using the syringes. This allowed for a more dynamic weight adjustment, for future payloads or minor weight variations.
- **Depth Control:** The method for depth control in Gover's iteration was to use vertically positioned propeller thrusters to move the vessel up or down. This proved to be taxing on mission time (battery life) and the level of control was very coarse.
 - **How it was addressed:** The depth control solution I implemented was ballast control with 4 60 ml syringes. The syringes required small adjustments to alter the depth of the vessel resulting in decreased battery usage and gave a greater level of stability in depth control due to the small increments in the mass of the vessel achievable through the use of linear actuators with position feedback.
- **Maintenance Access:** As the vessel required regular internal access it became prevalent that the design of the vessel should be altered to allow easier access. The thrusters were permanently built into the hull of the vessel and proved difficult to plug/unplug on assembly/disassembly.
 - **How it was addressed:** To increase accessibility of the vessel, the entire internal component of the AUV can be removed as one modular section. Only two hoses need to be unplugged for removal. This allowed quick (<5 min) removal for maintenance.

II. Background

The background theory that was used to achieve ballast control was based on a relatively simple equation. The equation using Archimedes principal which states that the force acting upwards on an object is equal to the weight of water that it displaces. Originally I considered ignoring drag to model the submarine, the resulting equation is shown in equation (1) below.

$$F = F_b + F_g$$

$$ma = (m_v + m_s)g - V_v \rho_w g \quad (1)$$

Equation (1) eliminates the drag term in an effort to simplify the equation, unfortunately the assumption that drag could be ignored was disqualified by modeling and practical testing. The revised equation, which includes drag is shown below in equation (2).

$$F = F_g - F_b - F_d$$

$$ma = (m_v + m_s)g - V_v \rho_w g - \frac{\rho_w v^2 A C_d}{2} \quad (2)$$

Equation (2) can be broken down into the following terms; the sum of forces (ma) is equal to the buoyancy force acting upwards on the object ($V_v p_w g$) and the force due to the mass of the object acting downwards ($(m_v + m_s)g$). The positive direction was defined to be down for all modeling. This equation is a second order non-linear, non-separable differential equation and due to its complexity was modeled in Matlab.

Equation (2) is a sum of forces equation, for the submarine to be at a steady depth the equation must be equal to 0. To achieve this the mass term must be equivalent to the buoyancy term. The only controllable variable is the m_s term, which is the mass of the water in the syringes (can be varied by ± 0.12 kg).

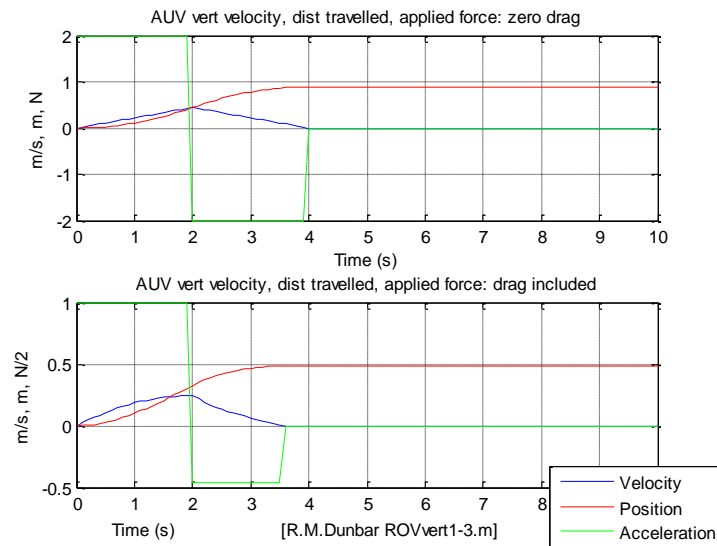


Figure 1-1: Effects of drag on motion of submarineⁱⁱ

We can see from figure above that it cannot be assumed that drag can be excluded from the modeling.

III. Aims

The aims for this project were laid out early in the development process. They are listed below;

- Develop a set of equations that model the forces acting on the Small White submarine,
- Build a prototype model to demonstrate the concept works as expected,
- Design and build the ballast control system, and
- Conduct testing to show the ballast control system working as intended.

These aims dictated the milestones in the project management schedule used and allowed for constant feedback into the iterative design and development process. As this project was focused on proving a concept and less on optimisation of the design, no requirements were set to gauge the success of each implemented system. This could be investigated at a later stage or in another years project.

These aims bled into a set of requirements that defined the project, listed below.

- **A Calibrated Pressure Sensor** – The pressure sensor feeds analog data to the Arduino via its analog port, which the Arduino converts to 1024 digital steps. These numbers themselves do not tell us any relevant pressure information, and depth cannot be inferred by them. For the pressure sensor data to be of any use an initial calibration experiment must be conducted to map the analog readings to a specific pressure, and in turn a depth in water.
- **A Depth Sensor Interface for the Arduino** – There is no standing documentation for the MPX4250A pressure sensor with regard to the Arduino microcontroller. In order to use the pressure sensor as a peripheral to the Arduino board an interface must be written for the pressure sensor.

- **A L12 Actuator Interface for the Arduino** – There is also no implemented support for the L12 linear actuator for the Arduino board. To use this actuator as a peripheral to the Arduino board an interface must be written for the linear actuator.
- **System for Depth Control** – A depth control system must be written to the Arduino. This system must use the inputs of the depth sensor and predetermined user inputs and drive the linear actuators and in turn change the depth of the vessel.
- **System for Trim Control** – Similarly to the depth control system, code must be written to the Arduino to allow trim control. The system must allow static trim control in addition to dynamic control to accommodate for changes in trim due to hydrodynamic forces. Dynamic control is not a necessary feature for the operation of the AUV, and will be investigated should time and resources permit.
- **PID Control System for Stabilised Depth** – To satisfy the depth stabilisation objective a control system must be written to the Arduino. This control system must be able to hold the vessel at a determined depth to within a maximum deviation. A PID controller will be used for this system to achieve an optimised response.
- **Wireless Transmission** – In order to give a large variety of commands to the AUV without removing the microcontroller (a relatively lengthy process) wireless communication would make the AUV much more dynamic.

IV. *Design and Development*

Design Philosophy

The design philosophy used for this project was the iterative design process. This approach was chosen to allow continual input from the physical testing aspect of the project into the mathematical modelling, and vice versa. This was integral to the project as some of the models were based on assumptions that were not reflected entirely in the real world, this allowed for minor adjustments to feedback into the theoretical work. This will become apparent throughout this chapter.

Design Stages

The project went through two major phases of design; the proof of concept model and the final prototype.

Proof of Concept Design

The proof of concept model was intended to do a few things. Firstly the model would familiarise me with the concepts of the final design and how the mathematical models behave in the real world. Secondly the model would flag any issues that may arise early in the design and development process so they could be either removed or worked around. This proved to be integral in the project as issues with small actuator originally used, weight calibration and watertight integrity became prevalent.

The proof of concept model was designed around the following criteria;

- Quick and simple assembly,
- Include all major components of final design,
- Be operational as quickly as possible.

To this end the prototype was made from commercially available components; the exterior case was made from a 1 L peach jar, the microcontroller an Arduino UNO board, the actuator a firgelli PQ12 linear and the pressure sensor a freescale semiconductor MPX4250A pressure sensor.

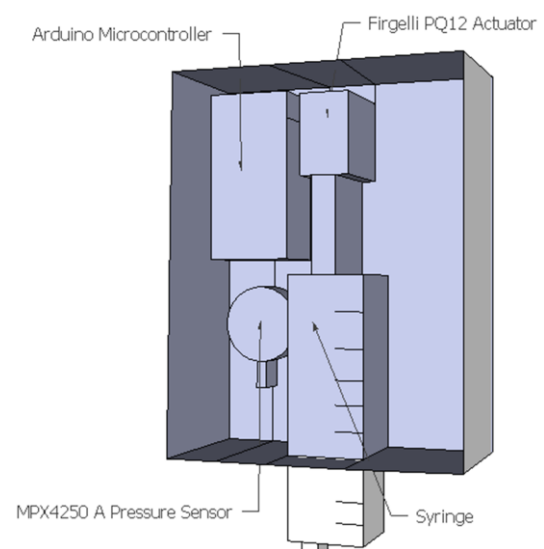


Figure 1-2: Proof of Concept Model

The prototype allowed the functionality of reading the pressure from the pressure sensor and converting that to an equivalent depth and driving the actuator for a set amount of time to fill or empty the syringe.

The first task was to find the buoyancy force exerted due to the volume of the plastic jar. The volume was found to be $1.2 \times 10^{-3} m^3$.

The buoyancy force then was given by,

$$F_b = g \times \rho_{water} \times V_{jar}$$

$$F_b = 11.76 \text{ N}$$

Where;

$$\rho_{water} = 1000 \text{ kgm}^{-3}$$

$$V_{jar} = 1.2e^{-3} m^3$$

In order to achieve neutral buoyancy, weights were added to give a force due to gravity as close to 11.76 N as possible. It is not necessary to be entirely precise, as the initial water in the syringe can be varied.

$$F_g = 11.76 = m \times g$$

$$m = 1.17 \text{ kg}$$

The total mass of the apparatus without ballast weight, m_{jar} , was 242 grams leaving a necessary weight of 928 grams.

As a proof of concept and fault finding exercise, I have considered the testing conducted to be a success. One of the objectives of this project was to have minimal weight calibration required, the PQ12 linear actuators had a such a small stroke length (20 mm) and the syringes were of a small volume (60 ml) that the weight of the test rig had to be more precise. To allow for this, a larger actuator was used in the final build (of length 100 mm). Additionally the actuator used in the proof of concept model had no position feedback, this meant that the actuators were driven for an amount of time instead of to a position. This problem was also addressed by the new actuator used, which had a position feedback option. This is one example of the iterative design used throughout this project. Another design consideration flagged in the prototype testing was the need for wireless communication to tell the AUV when to dive, this saves rushing to get the vessel in the pool for its initial intake of water.

Final Design

From the considerations flagged by the proof of concept model, the final design was conducted. There are a number of design aspects that were undertaken in the process, these are; positioning of the actuators, positioning of the electronics, number and size of the inlet and outlets, the choice for wireless communications, the implementation of the control system. Figure 1-3 is an exploded view of the vessel.

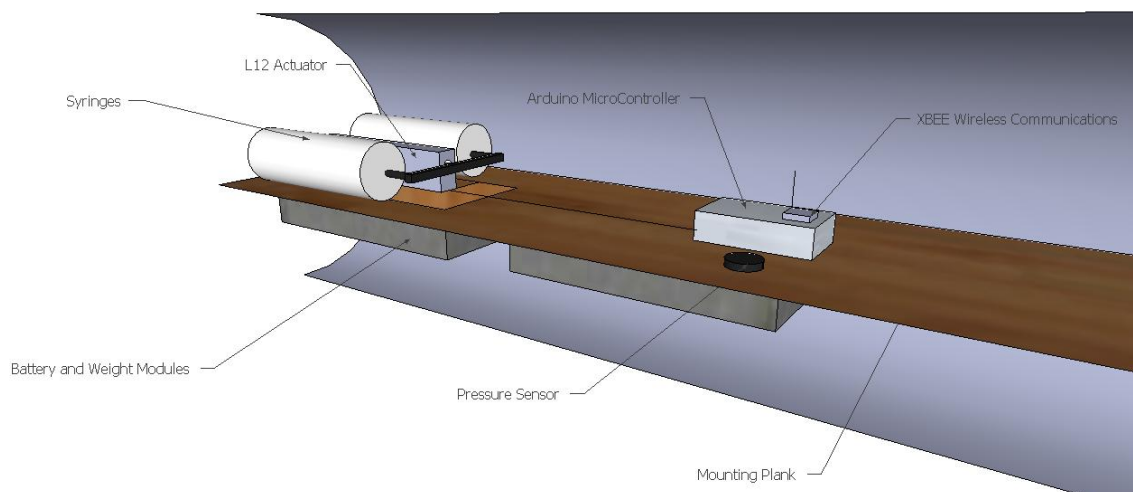


Figure 1-3: AUV Final Design Layout

Microcontrollerⁱⁱⁱ

The microcontroller used in the AUV was an Arduino UNO board shown in figure 1-4 . The Arduino was chosen as it has a wide range of libraries written and uses the ATMEGA CPU chip, which provides a fast enough clock for our purposes.

Positioning of Syringes

The syringes were positioned as far fore and aft of the vessel as possible. This was done to give maximum lever arm for trim control. I created a threaded bar to join the syringes to the actuator and fix them in place. The two syringes shown in figure 1-3 are mirrored on the other end of the AUV.

XBEE Wireless Communication

The XBEE wireless device was chosen for its simplicity in sending serial data between the Arduino microcontroller and the computer. Using the XBEE allowed me to communicate with the AUV, to give it commands and to receive depth data recorded during its missions. The XBEE transmits at 2.4 GHz, meaning underwater communications cannot be maintained, but this was countered by transmitting and receiving before and after each mission while the AUV was surfaced.



Figure 1-4: Arduino UNO microcontroller



Figure 1-5: XBEE Transmitter

Component Layout

The positioning of the battery and weight modules and electronics packages was laid out to ensure that the vessel is always upright, this eliminates the roll degree of freedom and ensures that the electronics are above any residue water from leaks to prevent short circuiting or damage.

All internal components were fixed to the mounting plank for quick removal of the internal mechanisms from the vessel. There is an outlet and an inlet cable that must be detached for removal, but fittings were installed to keep this modular.

Control System

A PID depth controller was implemented on the Arduino microcontroller. The considerations for this controller were; the sample time of the controller had to be limited to allow the actuators to extend/retract, the output signal had to be saturated to accurately model the small amount of variation in the mass (± 0.12 kg), and the control system had to allow for the case when the vessel overshoot the desired depth by inverting the output signal. A model of the control system is shown in appendix A.

Calibration of Subsystems

In order to keep the AUV open to upgrades or changes, the design was kept modular, with all peripherals interfacing with the onboard microcontroller. This has proven to be an effective way of managing sensors and actuators.

Pressure Sensor

The freescale semiconductor MPX4250A pressure sensor was used to determine the depth of the vessel.

This model is ideal for use in this project as it is designed to work with micro controllers, its small size and weight, and its relevant operating characteristic, which can be found in table 1-1



Figure 1-6: MPX4250A Pressure Sensor

Characteristic	Min	Typical	Max
Supply Voltage (V _s)	4.85	5.1	5.35
Pressure Range (kPa)	20	—	250
Operating Temperature (°C)	0	—	85
Accuracy (percent of full scale voltage)	-1.5%		+1.5%

Table 1-1 Operating Characteristics of the MPX4250A Pressure Sensor^{iv}

Table 1-1 illustrates the specifications of the pressure sensor used. The supply voltage range encompasses the output voltage of the Arduino board being used, the pressure range allows for depths up to 15 m (which is greater than the intended operating depth of 10 m), the temperature range is satisfactory for all predicted operating environments, the accuracy (1024 unique values) allows for approximately 0.22 kPa precision and the response time of 1 ms is fast enough to be used for the depth control.

In order to calibrate the pressure sensor a simple experiment was set up using an elevated hose filled with water. To emulate underwater pressure, the water level was raised to an equivalent depth above the pressure sensor. As barometric pressure varies from day to day and with location, instead of matching the depth of water to some predetermined pressure, the depth was matched relative to the initial pressure measured before each mission. These results can be seen in figure v-2.

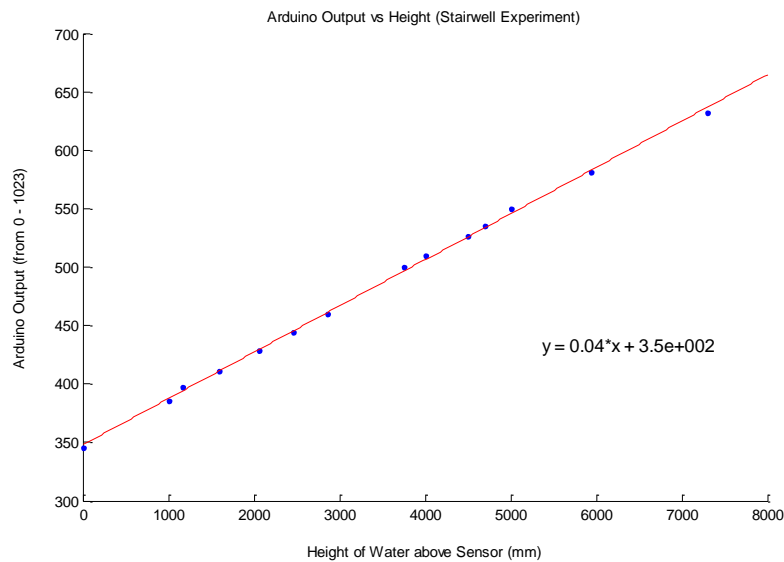


Figure 1-7: Calibration of pressure sensor.

Figure 1-7 show the results of the initial calibration of the MPX4250A pressure sensor. It can be seen from the results that each point was highly matched to the linear regression line. From these results a relationship was found relating depth to the Arduino output.

$$ArduinoOut = 0.04 * depth (mm) + initialOut \quad (3)$$

The initial Arduino output measured on the day of the experiment was 345 (corresponding to a barometric pressure of 102.27 kPa). Using both the initial output and the gradient found during this experiment, any depth can be represented by a specific Arduino output. This is highly beneficial in determining relative depth.

L12 Actuator

The L12 actuator is the 100 mm variant of the PQ12 actuator. Along with a larger stroke length, the L12 also has positional feedback, which is needed in the control system. The actuator was tested against its rated specifications found in the technical data sheet^v.

The results of the testing are shown in table 1-2 below.

Parameter	Rated	Actual
Extension Speed (no load)	12 mm/s	9 mm/s
Time for Full Extension	8.3 sec	11.3 sec
Positional Accuracy	0.3 mm	0.3 mm
Stroke Length	100 mm	102 mm
Analog Range (0-100 mm)	0-1023	9-1014

Table 1-2 L12 Actuator Specifications

V. Testing and Results

Testing for the final design was broken into two phases, no-control and control testing (that is with and without the PID controller implemented).

No Control

The test for the no control implementation of the final design was as follows. The AUV would initially take in 30 ml of water at each syringe to achieve neutral buoyancy. From there the AUV would take in the full amount of 60 ml in order to sink to the desired depth of 1 m before expelling water back to neutral buoyancy and then purging the ballast syringes to resurface. The test results can be seen in figure 1-8 below.

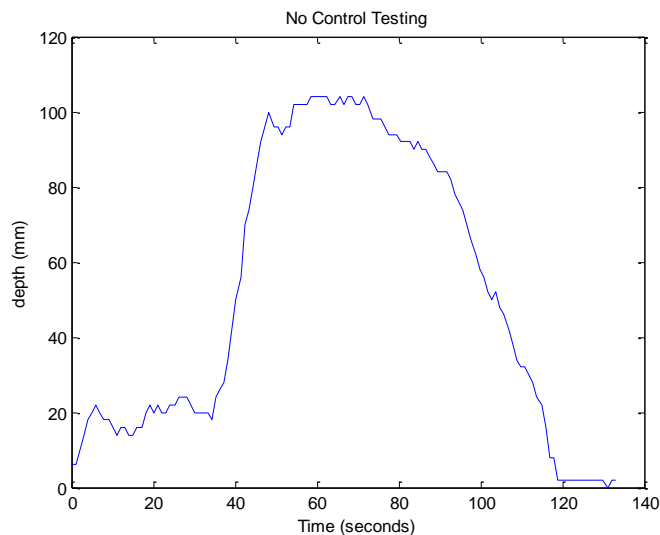


Figure 1-8: Depth response with no control

The noise in the first 40 seconds is due to the AUV being held under water slightly to allow the water intake to be submerged. At the peak of the graph the vessel has hit the bottom of the pool, hence the abrupt deceleration. We can see that without the control system implemented the acceleration is very rapid almost all the movement is done in around 5 seconds.

Control

Due to water tight integrity issues, the AUV was not able to sustain long periods at depth as the water intake overwhelmed the ballast system. Further efforts to seal the vessel are currently under revision in order to obtain results using the control system. Using a Matlab Simulink model, the theoretical behaviour of the vessel is shown in figure 1-9 below.

The figure shows the critically damped response of the control system for the desired depth of 3 meters. Due to the slow extension time of the actuators, only small pulses of acceleration can be used, or the system becomes rapidly unstable. This leads to a relatively long rise time of around 2 minutes to reach the desired depth to within 10%. There are benefits to this small acceleration boost however, as the battery consumption required is dramatically reduced in comparison to Gover's solution to depth control which led to operational time of around 35 minutes.^{vi}

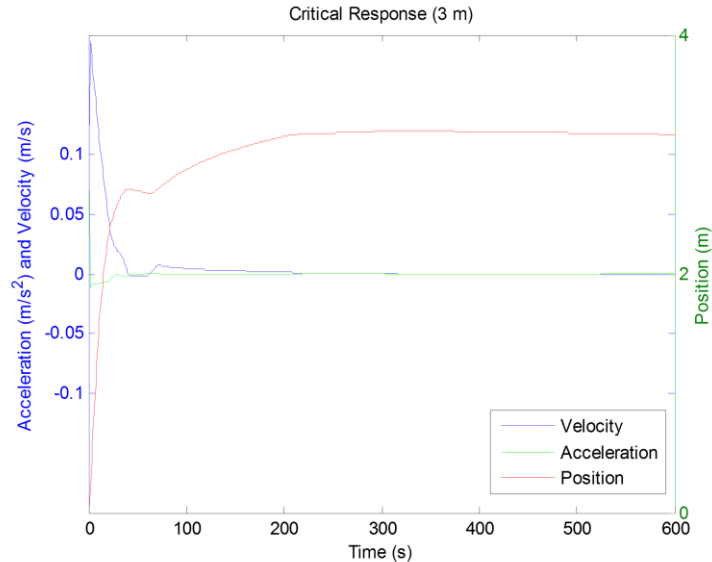


Figure 1-9: Critically damped response with PID implemented

VI. Conclusions

The purpose of the design project was to investigate alternative options for depth control in the pre-existing AUV under development within the SEIT at ADFA. The method for depth control chosen was ballast control, this method allowed the internal mechanisms of the system to be internalised within the vessel, which decreased water ingress, however due to issues with sealing the nose and tail section some undesired water ingress remains. This is addressed in the following chapter. The method chosen also greatly increased the capability of the AUV by decreasing the battery consumption of the depth control system, allowing more power to the propulsion and sensing systems. The aims of the project outlined in a previous chapter were met almost completely, due to aforementioned sealing problems the PID control system has not as yet been verified practically, though the theoretical models show the system can successfully control depth. Implementing the XBEE allowed for wireless communication between a laptop and the AUV to receive commands and to receive recorded data from the AUV's missions.

VII. Recommendations

If this project is to be picked up in further years, there are a number of recommendations that I would put forward to enhance the AUV.

Thread and O-Ring Seal

One of the largest problems encountered throughout the testing phase of this project was water tight integrity. While undesired water intake was minimal just below the surface, as soon as the vessel dove to depths greater than half a meter the pressure overcame any seals and the AUV took on large amounts of water that disabled it from resurfacing. In order to address this problem I propose that the PVCU center and machined nose and tail sections be threaded to enable plumbers thread tape, additionally a groove and O-Ring configuration shown in figure 1-10 should be implemented to ensure the vessel is sealed.

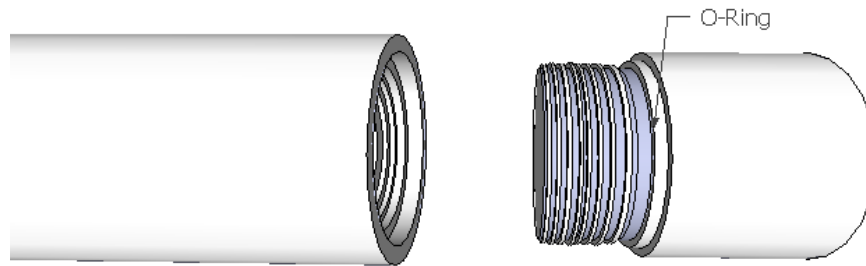


Figure 1-10: Thread and O-Ring Sealing suggestion

Larger Volume Syringes

The current implementation uses four 60 ml syringes, If the AUV is set up to be neutrally buoyant at 30 ml in each syringe this allows only +/- 120 grams of mass, this has an effect on the acceleration achievable by the AUV. Larger syringes would allow the vessel to move vertically more quickly in the water, this however will depend on the requirements of the vessel.

Faster Actuators

As shown in the initial calibration of the L12 actuator, the extension speed differed quite dramatically (from 8.3 sec to 11.3 sec) to the tech data sheet provided. A slower actuator speed meant that the vessel took a longer time to respond to the control system. To react to this I decreased sampling time of the PID controller to allow the actuators time to drive to the required length. Faster actuators would greatly improve the response time and stability of the control system.

Integrate Accelerometer for Trim Control

While trim control was achievable statically, no angular feedback was used, making the trim control primitive. If the microcontroller had an accelerometer, pitch could be determined and controlled using a separate PID controller.

VIII. Acknowledgements

I would like to personally thank my thesis supervisors Craig Benson and Robin Dunbar for their insight and guidance throughout the course of this project. Their ability to look through a problem that I had been buried in and point out something I had missed was beyond helpful. I would like to thank Brett Beauregard for his permission to use his Arduino PID library and for answering any questions I had for him. I would also like to thank my friends and family for all their support and interest, specifically Gerard Martin and James Edge-Williams who showed insight and suggestions into any problems I bounced off them. Finally I would like to thank my partner, Liana for her support and understanding throughout the long nights of little sleep.

IX. References

Papers

2LT Nicholas Gover *Optimum Design of a Small Underwater Vehicle: Small White Design and Build*

Software

Matlab and Simulink software 1984-2008, The MathWorks, Inc.

Arduino sketch-up program Massimo Banzi, David Cuartielles, Tom Igoe, Gianluco Martino and David Mellis

Data Sheets

PQ12 and L12 Actuator by firgelli. Available at www.firgelli.com

Freescalo Semiconductor MPX4250A pressure sensor, data sheet – revision 7, Freescalo Semiconductor, Inc, 2006

Code

Arduino PID library courtesy of Brett Beauregard

ROVvert1_3.m courtesy of Robin Dunbar

ⁱ 2LT Nicholas Gover *Optimum Design of a Small Underwater Vehicle: Small White Design and Build*

ⁱⁱ Courtesy of Robin Dunbar. From ROVvert1_3.m

ⁱⁱⁱ <http://arduino.cc/en/>

^{iv} Freescalo Semiconductor MPX4250A Pressure sensor data sheet – revision 7. Freescalo Semiconductor, Inc., 2006

^v Data sheet for L12 Actuator available at <http://www.firgelli.com/>

^{vi} *Optimum Design of a Small Underwater Vehicle: Small White Design and Build.*