AUTOMATED BALLAST TANK CONTROL SYSTEM FOR AUTONOMOUS UNDERWATER VEHICLES

by

Shawn A. Woods

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
March 2012

DALHOUSIE UNIVERSITY

DEPARTMENT OF MECHANICAL ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled “AUTOMATED BALLAST TANK CONTROL SYSTEM FOR AUTONOMOUS UNDERWATER VEHICLES” by Shawn A. Woods in partial fulfilment of the requirements for the degree of Master of Applied Science.

Dated: March 30, 2012

Co-supervisor: ________________________________

Co-supervisor: ________________________________

Readers: ________________________________

____

DALHOUSIE UNIVERSITY

DATE: March 30, 2012

AUTHOR: Shawn A. Woods

TITLE: AUTOMATED BALLAST TANK CONTROL SYSTEM FOR AUTONOMOUS UNDERWATER VEHICLES

DEPARTMENT OR SCHOOL: Department of Mechanical Engineering

DEGREE:	M.A.Sc.	CONVOCATION:	May	YEAR:	2012

Signature of Author

I dedicate this thesis to David Russell for providing the motivation to continue my pursuit of knowledge and for believing that my hard work and effort would be rewarding in the end. I also dedicate this work to my mother Karen Fillmore, step-father Garry Fillmore, and father Ronald Woods for giving me continued encouragement during my studies.

TABLE OF CONTENTS

List of Tables

List of Figures

Abstract

Underwater autonomous vehicles are frequently used for deep-water ocean applications such as surveying and cable-laying, where accurate control of vehicle depth and attitude is needed. The water level in the on-board ballast tanks are typically manually set for neutral buoyancy before each mission, while the vehicle is on the surface. The resulting weight of the water level is not normally adjusted while the unmanned vehicle is in operation to control vehicle depth and orientation. As a result, vehicle trajectory and orientation is exclusively controlled using the vehicle's control surfaces during a mission. The challenges with controlling the depth and trim of an underwater vehicle include nonlinear hydrodynamic forces as well as relatively slow response times and inherent time delays (latencies) associated with water tank level changes and valve adjustments. To meet these challenges, this thesis proposes two unique variable ballast system control approaches. The proposed control approaches may be suitable for large autonomous underwater vehicles with both small (volume = 0.027 m^3, each) and large (volume = 0.216 m^3, each) ballast tanks. The first proposed variable ballast system controller uses the current parameters of the ballast tanks to determine the appropriate action to be implemented. This controller was designed change the weight of the AUV to help control vehicle parameters such as depth and vertical (inertial) velocity. The second proposed variable ballast controller attempts to shift the center of gravity x_G along the body-fixed x-(longitudinal) axis by changing the weight in the ballast tanks. By shifting the center of gravity, the controller attempts to reduce depth and pitch angle error while regaining control authority to the bowplane and sternplane deflection fins. The ballasting system consists of two water tanks positioned aft and forward of amidships. The ballast tanks are then automatically filled or emptied of ocean water as desired.

Setpoint depth control and x_G shifting numerical simulations have been carried out on a two-dimensional underwater vehicle simulator to test and compare the performance of the proposed ballast and deflection control systems. The simulation results show that, for the assumptions and conditions tested, the proposed controllers are versatile and capable of achieving a setpoint depth and pitch angle with minimal error by effectively utilizing the ballast tanks and deflection fins. As a result, the work presented in this thesis helps increase the autonomy of large AUVs on long duration missions.

List of Abbreviations and Symbols Used

Symbol	Definition	Units
$A_{wat\ val}$	ballast tank water valve area	m^2
$A_{wat\ val\ max}$	maximum ballast tank water valve area	m^2
B	buoyancy of AUV	N
B_X	inertial *X*-axis component of AUV buoyancy	N
B_Y	inertial *Y*-axis component of AUV buoyancy	N
B_{AUV}, B_Z	inertial *Z*-axis component of AUV buoyancy	N
ΔB_{AUV}	change in total AUV buoyancy force	N
$\Delta B_{AUV@\max depth}$	change in total AUV buoyancy force at maximum AUV depth	N
c	1st moment of inertial with respect to point *O*	$kg \cdot m$
C_{ab}	inertial to body-fixed transformation matrix	–
ca_{cons}	compressed air constant	–
C_{ba}	body-fixed to inertial transformation matrix	–
db	bowplane angle	*rad*
$db_{setpoint}$	setpoint bowplane angle	*rad*
$depth$	depth of the AUV	*m*
ds	sternplane angle	*rad*
$ds_{setpoint}$	setpoint sternplane angle	*rad*
e_{depth}	depth error of the AUV	*m*
e_{pitch}	pitch error of the AUV	*rad*
f_b	weight and buoyancy forces	*N*
F_{dist}	vertical disturbance force	*N*
$F_{dist\ max}$	maximum vertical disturbance force	*N*
$F_{dist,x}$	x-axis component of vertical disturbance force	*N*
$F_{dist,z}$	z-axis component of vertical disturbance force	*N*
$F_{hyd,x}$	axial hydrodynamic force	*N*
$F_{hyd,z}$	normal hydrodynamic force	*N*

Symbol	Definition	Units
F_{prop}	forward propulsion force of the AUV	*N*
g	acceleration due to gravity	m/s^2
g_{ob}	moment due to forces	$N \cdot m$
$h_{air\ min}$	minimum ballast tank air height	m
$h_{bal\ tank}$	height of ballast tank	*m*
$height$	height of the AUV	*m*
$h_{minor\ loss}$	minor head loss	*m*
h_{wat}	height of water in ballast tank	*m*
$h_{wat\ aft}$	height of water in aft ballast tank	*m*
$h_{wat\ fwd}$	height of water in forward ballast tank	*m*
I_x	*x* component of the 2nd moment of interia	m^4
I_{xx}	*xx* component of the 2nd moment of interia	m^4
I_y	*y* component of the 2nd moment of inertia	m^4
I_{yy}	*yy* component of the 2nd moment of interia	m^4
I_z	*z* component of the 2nd moment of interia	m^4
I_{zz}	*zz* component of the 2nd moment of interia	m^4
J_b	2nd moment of inertia with respect to point O	m^4
k_{AUV}	compressibility of the AUV	Pa^{-1}
K_D	derivative control coefficient	*varies*
$K_{D,VBS}$	derivative control coefficient for VBS controller	*s/m*
K_P	proportional control coefficient	*varies*
$K_{P,bow}$	proportional control coefficient for bowplane control	*rad/m*
$K_{P,ste}$	proportional control coefficient for sternplane control	–
$K_{P,VBS}$	proportional control coefficient for VBS controller	*1/m*
k_{water}	compressibility of water	Pa^{-1}

Symbol	Definition	Units
$l_{bal\ tank}$	length of ballast tank	*m*
$length$	length of the AUV	*m*
$\dot{m}_{air}$	compressed air cylinder mass flow rate	*kg/s*
$\dot{m}_{air\ max}$	maximum compressed air cylinder mass flow rate	*kg/s*
$mass$	entire mass of the AUV	*kg*
M_{db}	pitching moment hydrodynamic component relating to *db*	–
M_{dbdb}	pitching moment hydrodynamic component relating to *dbdb*	–
$M_{db\|db\|}$	pitching moment hydrodynamic component relating to $db\|db\|$	–
M_{ds}	pitching moment hydrodynamic component relating to *ds*	–
M_{dsds}	pitching moment hydrodynamic component relating to *dsds*	–
$M_{ds\|ds\|}$	pitching moment hydrodynamic component relating to $ds\|ds\|$	–
$M_{hyd,x}$	rolling hydrodynamic moment	*N/m*
$M_{hyd,y}$	pitching hydrodynamic moment	*N/m*
$M_{hyd,z}$	yawing hydrodynamic moment	*N/m*
M_{prop}	propulsion moment of the AUV	*N/m*
M_q	pitching moment hydrodynamic component relating to *q*	–
$M_{q\ dot}$	pitching moment hydrodynamic component relating to $\dot{q}$	–
M_{qq}	pitching moment hydrodynamic component relating to q^2	–
$M_{q\|q\|}$	pitching moment hydrodynamic component	–

Symbol	Definition	Units
	relating to $q\|q\|$	
m_{ref}	reference mass of the AUV	kg
M_{star}	pitching moment hydrodynamic component relating to u^2	–
M_w	pitching moment hydrodynamic component relating to w	–
M_{ww}	pitching moment hydrodynamic component relating to w^2	–
$M_{w\|w\|}$	pitching moment hydrodynamic component relating to $w\|w\|$	–
ΔP	change in ocean water pressure	Pa
P_{air}	ballast tank air pressure	Pa
$P_{air,initial}$	initial ballast tank air pressure	Pa
$P_{air\ max}$	maximum allowable air pressure in ballast tank	Pa
$P_{current}$	current ocean water pressure	Pa
pd_{con}	pressure difference constant	$Pa \cdot s/m^3$
P_{diff}	pressure difference between water and air in ballast tank	Pa
P_o	initial ocean water pressure	Pa
pr_{cons}	pump rate constant	–
P_{wat}	ballast tank water pressure (external ocean pressure)	Pa
P_{vent}	venting pressure	Pa
q	angular velocity of the AUV around the body-fixed y-axis	rad/s
q_{dot}	angular acceleration of the AUV around the body-fixed y-axis	rad/s^2
$Q_{wat\ pump}$	volume flow rate of the water pump	m^3/s
$Q_{wat\ pump\ max}$	maximum vlume flow rate of the water pump	m^3/s

Symbol	Definition	Units
p	angular velocity of the AUV around the body-fixed *x*-axis	*rad/s*
$\dot{p}$	angular acceleration of the AUV around the body-fixed *x*-axis	rad/s^2
r	angular velocity of the AUV around the body-fixed *z*-axis	*rad/s*
$\dot{r}$	angular acceleration of the AUV around the body-fixed *z*-axis	rad/s^2
R_{air}	gas constant of air	$J/(kg \cdot °K)$
$rate_{air\ valve\ actual}$	actual air valve flow rate	*kg/s*
$rate_{comp\ air\ actual}$	actual compressed air mass flow rate	*kg/s*
$rate_{water\ pump\ actual}$	actual water pump flow rate	m^3/s
$rate_{water\ valve\ actual}$	actual water valve flow rate	m^3/s
$rate_{air\ valve\ set}$	set air valve flow rate	*kg/s*
$rate_{comp\ air\ set}$	set compressed air flow rate	*kg/s*
$rate_{water\ pump\ set}$	set water pump flow rate	m^3/s
$rate_{water\ valve\ set}$	set water valve flow rate	m^3/s
r_B	body-fixed position of the center of buoyancy of the AUV	*m*
R_x	*x*-axis component of the transformation matrix	–
R_y	*y*-axis component of the transformation matrix	–
R_z	*z*-axis component of the transformation matrix	–
T_{air}	temperature of air in the ballast tank	$°K$
$t_{open/close}$	full water valve open/close time	*s*
u	body-fixed *x*-axis velocity of the AUV	*m/s*
u_{dot}	body-fixed *x*-axis acceleration of the AUV	m/s^2
u_{out}	controller output	–
u_{cap}	capable controller output	–

Symbol	Definition	Units
$u_{cap,e}$	capable controller output for free-emptying	–
$u_{cap,f}$	capable controller output for free-filling	–
u_{dot}	body-fixed *x*-axis acceleration of the AUV	m/s^2
v	body-fixed *y*-axis velocity of the AUV	*m/s*
v_{dot}	body-fixed *y*-axis acceleration of the AUV	m/s^2
V_{air}	ballast tank air volume	m^3
V_{AUV}	AUV volume	m^3
ΔV_{AUV}	AUV volume change	m^3
$V_{AUV\ ref}$	reference volume of AUV	m^3
$V_{bal\ tank}$	total ballast tank volume	m^3
v_{cap}	capable fill or empty velocity	*m/s*
$v_{d,e}$	desired ballast empty velocity	*m/s*
$v_{d,f}$	desired ballast fill velocity	*m/s*
$v_{G,z}$	velocity at the center of gravity – inertial	*m/s*
$V_{large\ bal}$	total volume of the large ballast tank	m^3
v_o	AUV velocity	*m/s*
$\dot{v}_o$	AUV acceleration	m/s^2
$V_{small\ bal}$	total volume of the small ballast tank	m^3
v_{valve}	water valve open/close velocity	*m/s*
$v_{x\ bf}$	*x*-axis body-fixed velocity of the AUV	*m/s*
w	body-fixed *z*-axis velocity of the AUV	*m/s*
w_{bal}	width of the ballast tank	*m*
W_{bal}	weight of ballast tank	*N*
w_{dot}	body-fixed *z*-axis acceleration of the AUV	m/s^2
$width$	width of the AUV	*m*
w_{valve}	width of water valve	*m*
W	weight of AUV	*N*
W_X	inertial *x*-axis component of AUV weight	*N*

Symbol	Definition	Units
W_Y	inertial *y*-axis component of AUV weight	N
W_Z	inertial *z*-axis component of AUV weight	N
x	body-fixed *x*-axis position of the AUV	m
X	inertial *X*-axis position of the AUV	m
x_B	*x*-axis position of the center of buoyancy of the AUV	m
X_{dbdb}	axial force hydrodynamic component relating to *dbdb*	–
X_{dsds}	axial force hydrodynamic component relating to *dsds*	–
x_G	body-fixed *x*-axis center of gravity of the AUV	m
$x_{G\_aft\_bal}$	body-fixed *x*-axis center of gravity of the aft ballast tank	m
$x_{G\_fwd\_bal}$	body-fixed *x*-axis center of gravity of the forward ballast tank	m
$x_{G_{ref}}$	reference body-fixed center of gravity of the AUV	m
$x_{G\ shift}$	body-fixed x-axis center of gravity shift	m
X_q	axial force hydrodynamic component relating to q	–
X_{qq}	axial force hydrodynamic component relating to q^2	–
$X_{q\|q\|}$	axial force hydrodynamic component relating to $q\|q\|$	–
X_{star}	axial force hydrodynamic component relating to u^2	–
$X_{u\ dot}$	axial force hydrodynamic component relating to $\dot{u}$	–
X_{wq}	axial force hydrodynamic component relating to	–

Symbol	Definition	Units
	wq	
X_w	axial force hydrodynamic component relating to *w*	–
X_{ww}	axial force hydrodynamic component relating to w^2	–
$X_{w\|w\|}$	axial force hydrodynamic component relating to $w\|w\|$	–
y_B	body-fixed *y*-axis center of buoyancy of the AUV	*m*
y_G	body-fixed *y*-axis center of gravity of the AUV	*m*
z	body-fixed *z*-axis position of the AUV	*m/s*
Z	inertial *Z*-axis position of the AUV	*m*
z_B	*z*-axis position of the center of buoyancy of the AUV	*m*
Z_{db}	normal force hydrodynamic component relating to *db*	–
Z_{ds}	normal force hydrodynamic component relating to *ds*	–
z_G	*z*-axis position of the center of gravity of the AUV	–
Z_q	normal force hydrodynamic component relating to *q*	–
$Z_{q\ dot}$	normal force hydrodynamic component relating to $\dot{q}$	–
Z_{qq}	normal force hydrodynamic component relating to q^2	–
$Z_{q\|q\|}$	normal force hydrodynamic component relating to $q\|q\|$	–
Z_{star}	normal force hydrodynamic component relating	–

Symbol	Definition	Units
	to u^2	
Z_w	normal force hydrodynamic component relating to w	–
$Z_{w\,dot}$	normal force hydrodynamic component relating to $\dot{w}$	–
Z_{ww}	normal force hydrodynamic component relating to w^2	–
$Z_{w\|w\|}$	normal force hydrodynamic component relating to $w\|w\|$	–
ρ_{wat}	mass density of water	kg/m^3
ω	angular velocity of the AUV	*rad/s*
ψ	roll angle	*rad*
θ	pitch angle	*rad*
ϕ	yaw angle	*rad*

Acknowledgements

The authors would like to thank Defence Research and Development Canada (Atlantic) and the Natural Sciences and Engineering Research Council of Canada (NSERC) for providing financial support for this research.

Chapter 1 Introduction

Autonomous underwater vehicles (AUVs) are designed to perform underwater missions with minimal human interaction. The AUV, Theseus, for example, is shown in Figure 1 and was designed in 1992 to lay fiber-optic cable in the Canadian Arctic under-ice at water depths of down to 2,000 m [1]. The Artic Explorer AUV, developed in 2001, is capable of reaching depths of 5,000 m [2].

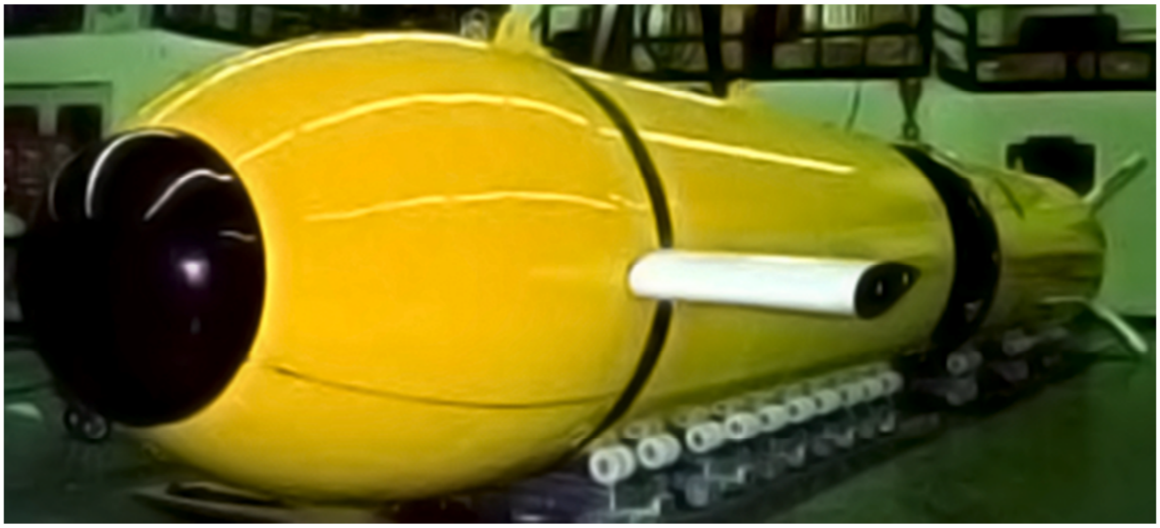

Figure 1 – Theseus AUV used in Laying Cable Under-ice [1]

1.1 Background and Motivation

While most AUVs require that the level of water in any on-board ballast tanks be adjusted manually, there has been relatively little reported in the literature to automate this process and enable the ballast tank water levels to be automatically varied during AUV operation. Xu and Smith [3] carried out detailed simulations of a variable ballast system (VBS) to control the depth of an AUV using an adaptive fuzzy logic control system. While the simulation results exhibited effective VBS depth control, the controller required further development to adaptively update the fuzzy rule base. More recently, Tangirala and Dzielski [4] utilized a VBS to try to control both the depth and trim of an

AUV. Resulting simulations and experiments of their VBS system showed that the proposed system was adequate only in water depths of 10 m or less. Sumantr and Karsiti [5] also utilized a VBS to control the depth of a spherical AUV. Simulated results showed that there was a significant power requirement to change the spherical AUV's weight in deep water due to the hydrostatic forces. A cigar-shaped AUV was studied by Byron and Tyce [6]. They proposed a proportional and derivative (PD) mass-ballast displacement controller for controlling an AUV capable of both horizontal and vertical stability. Their controller experimental results displayed naturally-occurring influences in the ocean that caused deviations in the stability of their AUV.

With AUVs going to much deeper depths and ever-increasing mission durations (AUV size is correlated to mission endurance), automated VBS is a timely and relevant topic that impacts AUV autonomy and their subsequent use. The use of ballast tanks makes sense on-board large AUVs which are generally long mission duration AUVs as well. The AUV's orientation (trim) has to be adjusted when the AUV transits through water regions that have changes in salinity and temperature. For AUVs with very large depth ranges (like Theseus with 2,000 m or the Arctic Explorer with 5,000 m) its buoyancy will also change due to compression of the vehicle at deep depths due to hydrostatic forces. For an AUV that is laying cable, its mass changes as it deploys the cable and this loss of mass will have an impact on the AUV's buoyancy and attitude. As well, for AUVs that altitude-keep, the seabed may have large changes (as seen in littoral waters) in its mean depth which may be better addressed through an automated VBS rather than sternplane and bowplane control. Given the need for further research into the use of a VBS to control the depth of AUVs, the following thesis objectives arose.

1.2 Thesis Objectives

- Derive the two-dimensional equations of motion for a submerged AUV
- Implement these equations of motion in MATLAB®, including the nonlinear hydrodynamic forces and torques experienced by a submerged AUV
- Develop and implement a ballast tank computer model

- Design and implement a viable controller and corresponding logistics required to manage the depth of an AUV by controlling the water contents of dual positioned (forward and aft) ballast tanks
- Design and implement a viable controller and corresponding logistics needed to manage the location of the center of gravity of the AUV to help increase the control authority of existing bowplane (forward-located AUV fins) and sternplane (aft-located AUV fins) proportional depth controller

1.3 Organization of Thesis

The following list shows how this thesis is organized:

- *Chapter 2* overview of large AUVs and corresponding 2D dynamic equations of motion
- *Chapter 3* design and implementation of the ballast tank computer model
- *Chapter 4* design and implementation of the MATLAB® computer simulator
- *Chapter 5* overview of the VBS controllers proposed
- *Chapter 6* analysis of the bowplane and sternplane proportional depth and pitch controller
- *Chapter 7* analysis of the proportional derivative (PD) VBS depth controller
- *Chapter 8* design and analysis of the proposed hybrid proportional derivative condition-based (HPDCB)VBS depth controller
- *Chapter 9* design and analysis of the proposed VBS controller to shift the longitudinal center of gravity, x_G, of the AUV along the body-fixed x-axis (VBS x_G shifting controller)
- *Chapter 10* results and discussion of the proposed controllers
- *Chapter 11* conclusions and recommendations

Chapter 2 Autonomous Underwater Vehicles

The following sections summarize the AUV motion equations and corresponding parameters needed develop a computer simulation of the system.

2.1 AUV Body-fixed Equations of Motion

For the scope of this thesis, the AUV is constrained to move in two dimensions (planar motion). Figure 2 shows the inertial (X, Z) and body-fixed (x, z) reference axes used to derive the equations of motion for an AUV. The center of buoyancy r_B is used as the body-fixed origin when implementing the AUV equations of motion.

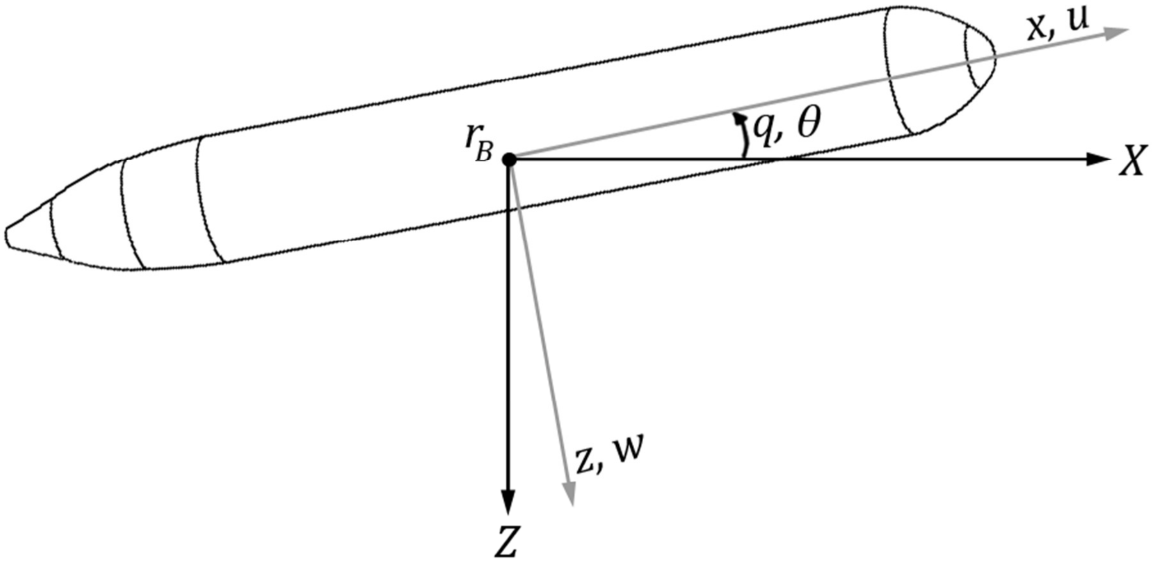

Figure 2 – AUV Reference Axes

The parameter r_B refers to the location of the center of buoyancy and is based on the pressure vessel geometry of the AUV (centroid of displaced volume) and will not change unless the actual volume of the AUV changes (which it could at deep depths). The parameters *x* and *z* refer to the body-fixed axes and the parameters *X* and *Z* refer to the inertial axes. The parameters *q* and *θ* refer to the angular velocity and pitch angle of the AUV, respectively. The parameters *u* and *w* refer to the body-fixed AUV velocities for the x-axis and z-axis, respectively. The body-fixed equations of motion and corresponding hydrodynamic components were presented by Feldman [8]. While Feldman showed these dynamic equations in 3D, this thesis focuses on 2D simulations where the axial force, normal force and pitching moment equations are, respectively, as follows:

$$\begin{aligned} mass \cdot [u_{dot} + w \cdot q - x_G \cdot q^2 + z_G \cdot q_{dot}] \\ = F_{hyd,x} - (W_Z + B_Z) \cdot sin(\theta) + F_{prop} \end{aligned} \tag{1}$$

$$mass \cdot [w_{dot} - u \cdot q - z_G \cdot q^2 - x_G \cdot q_{dot}] = F_{hyd,z} + (W_Z + B_Z) \cdot cos(\theta) \tag{2}$$

$$\begin{aligned} I_y \cdot q_{dot} + mass \cdot [z_G \cdot (u_{dot} + w \cdot q) - x_G \cdot (w_{dot} - u \cdot q)] \\ = M_{hyd,y} - (x_G \cdot W_Z + x_B \cdot B_Z) \cdot \cos(\theta) - (z_G \cdot W_Z + z_B \cdot B_Z) \\ \cdot \sin(\theta) \end{aligned} \tag{3}$$

where u_{dot}, u_{dot}, and q_{dot} refer to the body-fixed x-axis acceleration, z-axis body-fixed acceleration, and angular acceleration of the AUV, respectively. The parameters x_G, z_G, x_B, and z_B refer to the x-axis and z-axis positions of the center of gravity, and the x-axis and z-axis positions of the center of buoyancy, respectively. The parameters W_Z and B_Z refer to the z-axis weights and buoyancy of the AUV, respectively. The parameter F_{prop} refers to the propulsion force of the AUV. The parameters $F_{hyd,x}$, $F_{hyd,z}$, and $M_{hyd,y}$ refer to the axial forces, normal forces, and pitching moments of the AUV, respectively. The following are the hydrodynamic axial forces $F_{hyd,x}$, normal forces $F_{hyd,z}$ and pitching moments, $M_{hyd,y}$ of the AUV, modified from Feldman's [8] equations of motion, as provided by the Defence Research and Development Atlantic (DRDC Atlantic, written communication):

$$\begin{aligned} F_{hyd,x} = RL4 \cdot (X_{qq} \cdot q^2) + RL3 \cdot (X_{u\,dot} \cdot u_{dot} + X_{wq} \cdot w \cdot q + X_q \cdot u \cdot q) + RL2 \\ \cdot (X_{star} \cdot u^2 + X_{ww} \cdot w^2 + X_w \cdot u \cdot w) + RL2 \\ \cdot (X_{dbdb} \cdot u^2 \cdot db^2 + X_{dsds} \cdot u^2 \cdot ds^2) \end{aligned} \tag{4}$$

$$F_{hyd,z} = RL4 \cdot \left(Z_{q\,dot} \cdot q_{dot} + Z_{q|q|} \cdot q|q| + Z_{qq} \cdot q^2\right) + RL3 \cdot \left(Z_q \cdot u \cdot q + Z_{w\,dot} \cdot w_{dot}\right) + RL2 \cdot \left(Z_{star} \cdot u^2 + Z_w \cdot u \cdot w + Z_{w|w|} \cdot w^2 + Z_{ww} \cdot w^2\right) + RL2 \cdot (Z_{db} \cdot u^2 \cdot db + Z_{ds} \cdot u^2 \cdot ds) \tag{5}$$

$$M_{hyd,y} = RL5 \cdot \left(M_{q\,dot} \cdot q_{dot} + M_{qq} \cdot q^2 + M_{q|q|} \cdot q|q|\right) + RL4 \cdot \left(M_q \cdot u \cdot q\right) + RL3 \cdot \left(M_{star} \cdot u^2 + M_w \cdot u \cdot w + M_{ww} \cdot w^2 + M_{w|w|} \cdot w^2\right) + RL3 \cdot u^2 \cdot \left(M_{db} \cdot db + M_{dbdb} \cdot db^2 + M_{db|db|} \cdot db|db|\right) + RL3 \cdot u^2 \cdot \left(M_{ds} \cdot ds + M_{dsds} \cdot ds^2 + M_{ds|ds|} \cdot ds|ds|\right) + RL3 \cdot \left(M_{prop} \cdot u^2\right) \tag{6}$$

where X_{qq}, $X_{q|q|}$, $X_{u\,dot}$, X_{wq}, X_q, X_{star}, $X_{w|w|}$, X_{ww}, X_w, X_{dbdb}, and X_{dsds} are axial force hydrodynamic coefficients, $Z_{q\,dot}$, $Z_{q|q|}$, Z_{qq}, Z_q, $Z_{w\,dot}$, Z_{star}, Z_w, $Z_{w|w|}$, Z_{ww}, Z_{db}, and Z_{ds} are normal force hydrodynamic coefficients, and $M_{q\,dot}$, M_{qq}, $M_{q|q|}$, M_q, M_{star}, M_w, M_{ww}, $M_{w|w|}$, M_{db}, M_{dbdb}, $M_{db|db|}$, M_{ds}, M_{dsds}, and $M_{ds|ds|}$ are pitching moment hydrodynamic coefficients. The parameter M_{prop} represents the propulsion moment of the AUV. The parameters $RL2$, $RL3$, $RL4$, and $RL5$ are defined as follows:

$$RL5 = 1/2 \cdot \rho_{wat} \cdot length^5 \tag{7}$$

$$RL4 = 1/2 \cdot \rho_{wat} \cdot length^4 \tag{8}$$

$$RL3 = 1/2 \cdot \rho_{wat} \cdot length^3 \tag{9}$$

$$RL2 = 1/2 \cdot \rho_{wat} \cdot length^2 \tag{10}$$

where *length* represents the length of the AUV. The resulting 2D velocity along the absolute (inertial frame of reference) *X*- and *Z*-axes are given by:

$$\dot{X} = u \cdot cos(\theta) + w \cdot sin(\theta) \tag{11}$$

$$\dot{Z} = -u \cdot sin(\theta) + w \cdot cos(\theta) \tag{12}$$

where $\dot{X}$ represents the *X*-axis inertial velocity of the AUV and $\dot{Z}$ represents the *Z*-axis inertial velocity of the AUV. The angular velocity about the absolute *Y*-axis is simply:

$$\dot{\theta} = q \tag{13}$$

where $\dot{\theta}$ represents the angular velocity of the AUV. The full derivation of these equations of motion can be found in Appendix A. The AUV simulator incorporates these hydrodynamic forces into the AUV dynamic equations of motion.

2.2 Chosen AUV Parameters

Table 1 lists the parameters for a hypothetical AUV simulated in this thesis. Several of these parameters were modelled after those from the Theseus and Arctic Explorer AUVs. The parameters that were modelled after one of those AUVs will be denoted by [T] (Theseus) or [A] (Arctic Explorer). As shown in this table, two ballast tank sizes ("small" and "large") were implemented to help illustrate the effect that these different design choices would have on the performance of the AUV.

Table 1 – Chosen AUV Parameters

Parameter	Value
AUV Length [1][T]	10.7 m
Ballast Tank Sizes	0.30 m x 0.30 m x 0.30 m ("small") 0.60 m x 0.60 m x 0.60 m ("large")
Basic Control Type [1][T]	bowplane and sternplane proportional controller
Bowplane and Sternplane Fin Ranges	±25°
Compressed Air Cylinder Volume	0.1 m^3
Forward Velocity Range [1][T]	0 to 2.06 m/s (0 to 4 knots)
Initial Compressed Air Cylinder Pressure	100 MPa
Mass / Displacement [1][T]	8,600 kg (84,366 N)
Operating Depth Range [2][A]	0 to 5,000 m

Parameter	Value
Reference Center of Buoyancy	x_B = 0 m, y_B = 0 m, z_B = 0 m
Reference Center of Mass	x_G = 0 m, y_G = 0 m, z_G = 0.25 m†

† The AUV's vertical center of mass z_G along the body-fixed z-axis will be modified to 0.075 m within chapter 9 to help enhance the effect of shifting the body-fixed x-axis center of gravity. For all other chapters, the value of z_G will remain 0.25 m.

2.3 Hydrodynamic Coefficients

The following hydrodynamic coefficients have been provided by Defence Research and Development Atlantic (DRDC Atlantic, written communication). The axial hydrodynamic coefficients used for all AUV-based simulations in this thesis are as follows:

- X_{qq} = 0.0093828
- $X_{u\ dot}$ =–0.0005729
- X_{wq} = 0
- X_q = 0
- X_{star} =–0.0032870
- X_{ww} = 0.0898433
- X_w = 0
- X_{dbdb} =–0.0035418
- X_{dsds} =–0.0064658.

The normal hydrodynamic coefficients used for all AUV-based simulations in this thesis are as follows:

- $Z_{q\ dot}$ =–0.00005726
- $Z_{q|q|}$ =–0.0021804
- Z_{qq} = 0
- $Z_{w\ dot}$ =–0.01998
- Z_{star} = 0
- Z_w =–0.0770456
- Z_{ww} = 0
- Z_{db} =–0.01605397.
- Z_{ds} =–0.01998134
- Z_q = 0.0006267
- $Z_{w|w|}$ =–0.1203970

The pitching moment hydrodynamic coefficients used for all AUV-based simulations in this thesis are as follows:

- $M_{q\ dot}$ =–0.001196
- M_{qq} = 0
- $M_{q|q|}$ =–0.0015860
- M_w = 0.0180739
- M_{ww} = 0
- $M_{w|w|}$ = 0.0031395
- M_{db} = 0.0062778
- $M_{db|db|}$ = 0.0000000
- M_{ds} =–0.008312668
- M_{dsds} = 0.
- $M_{ds|ds|}$ = 0
- M_q =–0.0086821

- $M_{star} = 0$
- $M_{dbdb} = 0$

One of the most important components of the simulated AUV is its dual ballast tanks. Their purpose is to change the weight of the AUV in order to reach a specific setpoint depth. Coupled with a method of control, the tanks present an alternative to augment the traditional bowplane and sternplane method of control. An overview of the ballast tank setup is covered in Chapter 3.

Chapter 3 Ballast Tank Design

The presence of both a forward and an aft ballast tank is essential to the proposed AUV control solution to drive the AUV to a setpoint depth, or shift the center of gravity (weight) along the body-fixed x-axis. The ballast tank incorporates water valves, air valves, a compressed air cylinder assembly, and a water pump assembly. With each of these components, the variable ballast system is able to change its weight by modifying its water and air content to comply with the objectives of the ballast tank controller.

In situations where the AUV is to ascend or descend to a specific depth, the ballast tank will attempt to decrease or increase the water content (weight) of the two tanks at the same rate. Situations requiring depth control with the ballast tanks will be explored in detail in Chapter 7 and Chapter 8. When attempting to shift the center of gravity (weight) of the AUV either forward or aft, the ballast tank water content (weight) will either be increased or decreased in each tank as required by the controller. In these situations, it is important to know the location of the center of gravity for each of the ballast tanks. The center of gravity x_G along the body-fixed x-axis of the AUV can be determined using the following equation:

$$x_G = \left(\frac{1}{mass}\right)\left[\left(m_{fwd} * x_{G\_fwd\_bal}\right) + \left(m_{aft} * x_{G\_aft\_bal}\right) + \left(m_{ref} * x_{G_{ref}}\right)\right] \qquad (14)$$

where m_{fwd} represents the mass of the forward ballast tank, m_{aft} represents the mass of the aft ballast tank, m_{ref} represents the reference mass of the AUV, $x_{G\_fwd\_bal}$ represents the x-axis center of gravity of the forward ballast tank, $x_{G\_aft\_bal}$ represents the x-axis center of gravity of the aft ballast tank, and $x_{G_{ref}}$ represents the reference center of gravity of the AUV. Situations requiring the ballast tanks to shift the center of gravity along the x-axis will be explored in detail in Chapter 9.

Figure 3 shows a schematic of the “small” cubic ballast tank where all sides of the tank have a measurement of 0.30 m (300 mm). Each of the two small ballast tanks on the AUV, therefore, has a volume of 0.027 m^3 resulting in a total combined volume of 0.054 m^3. For the simulations carried out in this thesis, both ballast tanks on the AUV are assumed to be of identical size and to have identical neutral buoyancy conditions. As shown in Figure 3, each ballast tank is equipped with: a water pump, a compressed air source, an air vent to help control the pressure inside the ballast tank, and valves for free-filling and free-emptying of the tank.

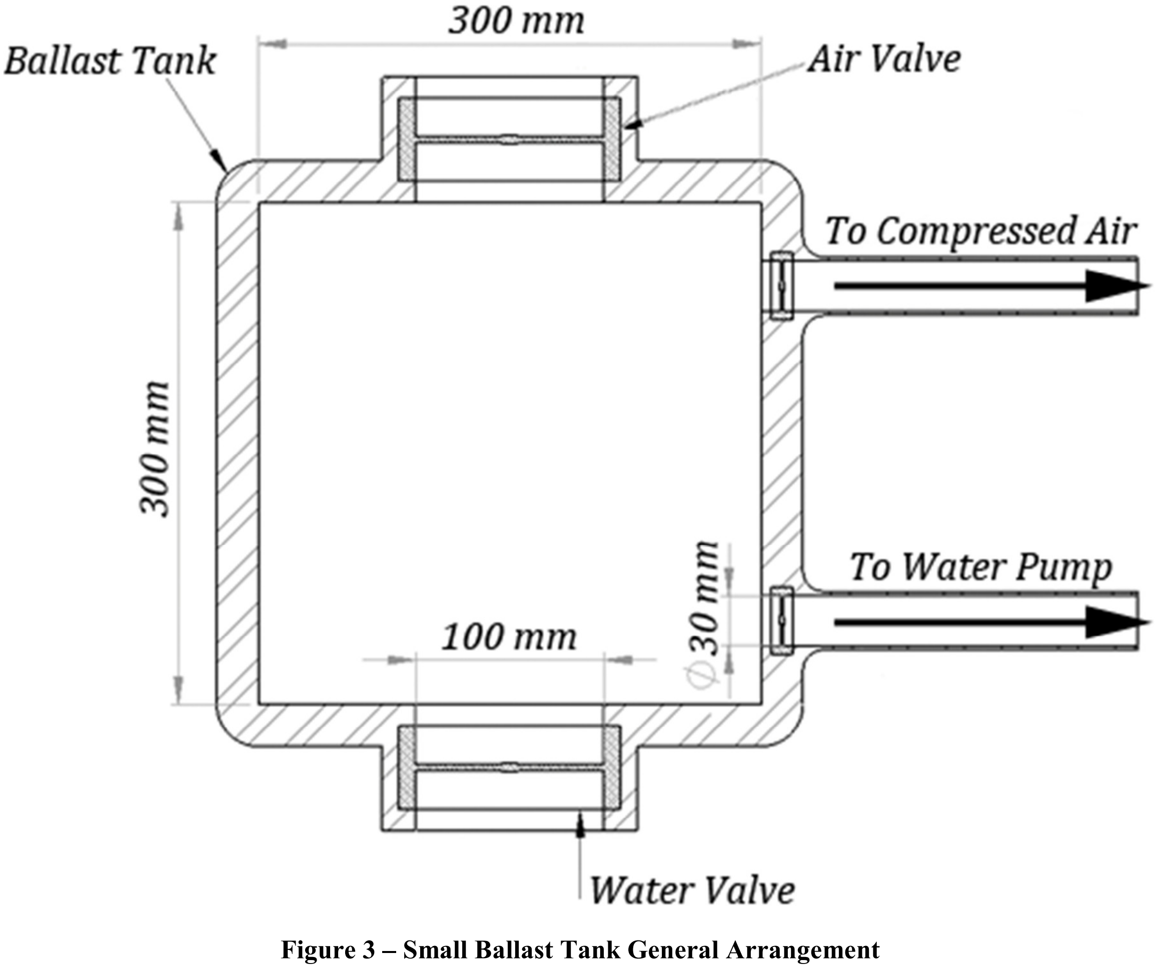

Figure 3 – Small Ballast Tank General Arrangement

3.1 Free-Fill and Free-Empty Rate

Free-filling and free-emptying may occur when the ballast tank water or air valves are opened and a pressure difference P_{diff} exists between the external environment water pressure P_{wat} and the air pressure within the ballast tank P_{air}. This water pressure difference is calculated as follows:

$$P_{diff} = P_{wat} - P_{air}. \tag{15}$$

In the scenarios covered in this thesis, the external environment is assumed to be ocean water (salt water; density, ρ_{wat} = 1,025 m^3/s) with the corresponding water pressure calculated as follows:

$$P_{wat} = \rho_{wat} \cdot g \cdot depth \tag{16}$$

where the *g* represents acceleration due to gravity and *depth* represents the depth of the AUV. Having a positive pressure difference results in potential free-filling, whereas having a negative pressure difference results in potential free-emptying.

The rate of change of the water height in the ballast tank v_{wat} is a function of several changing parameters including the pressure difference between the interior and exterior of the tank as well as the water valve orifice area. Figure 4 plots ballast tank water velocity as it varies as a function of pressure difference P_{diff} for both a "small" and "large" ballast tank (see Section 3.6 for further detail regarding ballast tank size). An estimated pressure difference constant pd_{con} of 334,888,890 $(Pa \cdot s)/m^3$ is used in calculating the ballast tank water velocity. The rate of change of height in the ballast tank v_{wat} (also referred to as *ballast tank water velocity*) can then be calculated based on conservation of mass as follows:

$$v_{wat} = \left(\frac{P_{diff}}{pd_{con}}\right)\left(\frac{A_{wat\,val}}{A_{wat\,val\,max}}\right)\left(\frac{1}{{w_{bal}}^2}\right) \tag{17}$$

where ${w_{bal}}^2$ represents the cross-sectional area of the ballast tank and $A_{wat\,val}/A_{wat\,val\,max}$ represents the fraction that the water valve is opened.

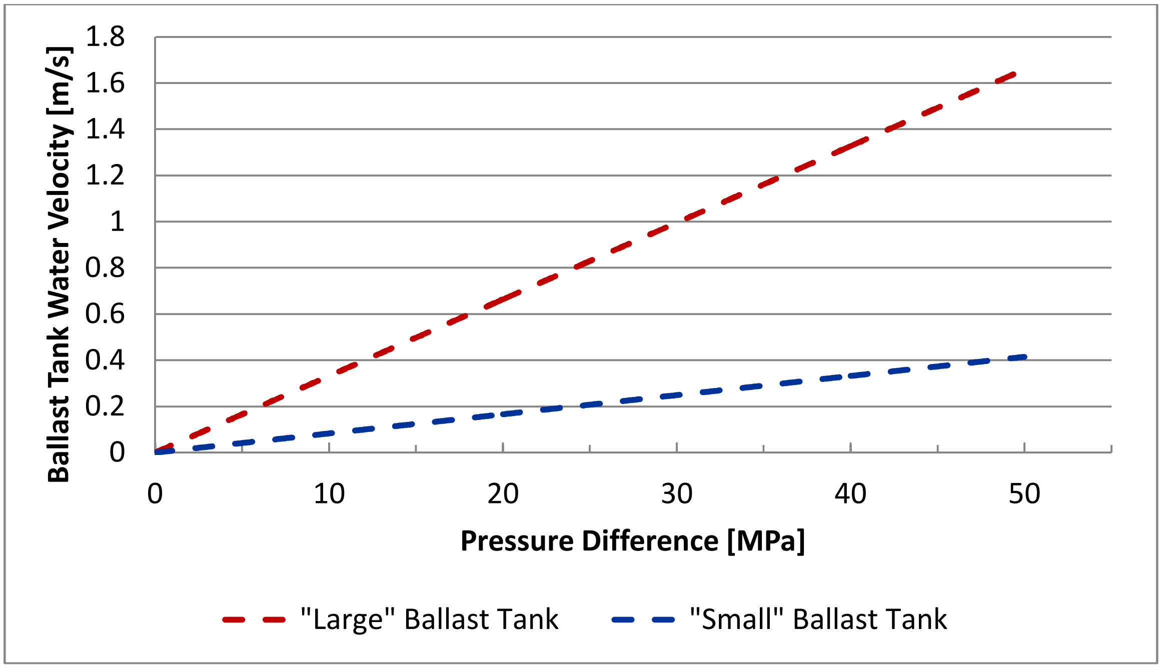

Figure 4 – Free-filling and Free-emptying Velocity Approximation

The flow in and out of the ballast tank is unsteady. However, if assumed to be steady, Bernoulii's equation can be used to calculate a flow rate that varies with the square of the pressure difference $\left(flow\ rate \propto \sqrt{P_{diff}}\right)$. For further accuracy, the Navier-Stokes equations and finite element analysis (FEA) can be used to determine the flow rate in the ballast tank.

3.2 Water Pump Support

It is assumed that there are two water pumps within the AUV: one for each of the ballast tanks. The water pumps are used to compensate for an inadequate pressure difference between the interior and exterior of the ballast tank to achieve a desired free-fill velocity. During water pump support, the water and air valves need to be closed. If the water or air valve is opened, and the air pressure in the ballast tank is higher than the ocean water, there may be undesirable free-emptying due to the pressure difference. It is also assumed that the water pump does *not* operate at the same time as the compressed air system so that they do not conflict with one another given their opposing purposes. To ensure that the pressure within the ballast tank does not rise above a critical limit, the air pressure is compared against a maximum-allowable value P_{vent} beyond which potential damage to the ballast tank, valves and other structural components may occur. As the pressure in the tank approaches this pressure limit, the water pumps are shut off. Another assumption is that there are no back pressures experienced by the water pump. As a result, the water pump outputs a volume flow rate will not change as a result of the air pressure in the ballast tank.

3.3 Compressed Air Support

Within the AUV, it is assumed that there are two air compressors present; one for each of the ballast tanks. When the air pressure inside the ballast tank is not adequate to provide a desired free-empty flowrate, compressed air can be used to increase the mass of the air and, assuming an ideal gas, the pressure of the air in the ballast tank. It is assumed that compressed air support is only available when the air valve is closed to prevent unwanted loss of air. Furthermore, it is assumed that compressed air support can also only occur if the compressed air output pressure is larger than the air pressure within the ballast tank to ensure that air flows into the tank. Finally, it is assumed that there is no back pressure to hinder air flow from the compressed air supply. Depending on the pressure difference in the ballast tank, the water valve itself may be opened or closed during compressed air support. For example, if the water pressure outside the ballast tank is higher than the air pressure in the tank, the water valve will remain closed to prevent unwanted free-filling from occurring. Free-emptying of water from the ballast tank will occur when the water valve is open if the pressure difference P_{diff} between the water outside the ballast tank P_{wat} and the pressure inside the ballast tank P_{air} satisfies the following relationship:

$$P_{diff} \leq 0. \tag{18}$$

The maximum air mass flow rate utilized during compressed air support will vary depending on setpoint depth. The reason for having a variable air mass flow rate is that, depending on depth, the air mass required to remove the water from the ballast tank will vary greatly. At large depths, such as 5,000 m, a low air mass flow rate will take much longer to increase the air mass within the ballast tank as opposed to a higher air mass flow rate. When the mass increases, this will increase the air pressure and eventually allow for free-emptying to remove the water from the ballast tank. The maximum air mass flow rate can be found by the following equation involving the MATLAB® function, *round*:

$$Air\ Mass\ Flow\ Rate = round(setpoint\ depth/500)/100. \qquad (19)$$

The *round* function, in MATLAB®, rounds a value to the nearest integer. The value of the air mass flow rate is saturated between 0.01 kg/s and 0.10 kg/s, where a setpoint depth of 0 m will result in a value of 0.01 kg/s and a setpoint depth of 5,000 m will result in a value of 0.10 kg/s. Exact ranges for the air mass flow rate can be seen in Table 2:

Table 2 – Maximum Air Mass Flow Rate

Setpoint Depth Range m	Maximum Air Mass Flow Rate kg/s
0 – 750	0.01
750 – 1,250	0.02
1,250 – 1,750	0.03
1,750 – 2,250	0.04
2,250 – 2,750	0.05
2,750 – 3,250	0.06
3,250 – 3,750	0.07
3,750 – 4,250	0.08
4,250 – 4,750	0.09
4,750 +	0.10

3.4 Air Venting

The purpose of venting the air in the ballast tank is to ensure that the pressure inside the tank does not become too high. If the air mass and corresponding air pressure increases inside the ballast tank, the ability for effective free-filling will diminish and, eventually, the ability to fill the ballast tank using the water pump will also disappear. It is assumed that, when the air is compressed, the air pressure rises according to the idea gas law:

$$P_{air} = m_{air} \cdot R_{air}T_{air}/V_{air} \qquad (20)$$

where m_{air} represents air mass, R_{air} represents the gas constant of air, T_{air} represents the temperature of air, and V_{air} represents the volume of air in the ballast tank.

When the air mass increases and the air volume decreases due to compression, the resulting air pressure will rise relatively quickly. The ballast tank needs to begin venting air when the air pressure P_{air} exceeds the maximum preset pressure limit for the tank $P_{air\,max}$. The parameter P_{vent} is used to check whether or not air venting is required:

$$P_{vent} = P_{air} - P_{air\,max} \tag{21}$$

where $P_{air\,max}$ represents the maximum allowable air or water pressure (i.e. 10 MPa) for the ballast tank used in this thesis. When P_{vent} is equal to or greater than zero, the maximum pressure in the ballast tank has been met or exceeded and the air valve is opened to allow the air to vent, thus reducing its mass and corresponding pressure inside. The higher-pressure air (relative to the outside water pressure) will flow out of the ballast tank and into the ocean until the pressures of air and water are equal. For simulation purposes, it is assumed that air venting can only occur when the mass of the air in the ballast tank is greater than a minimum value of 0.005 kg. This minimum air mass value is used to keep the air from needlessly being vented when its mass is negligible.

3.5 Ballast Tank Logic Overview

Figure 5 summarizes the logic associated with the use VBS, while Table 3 gives a general overview of the controller status and corresponding outputs for VBS-based controllers.

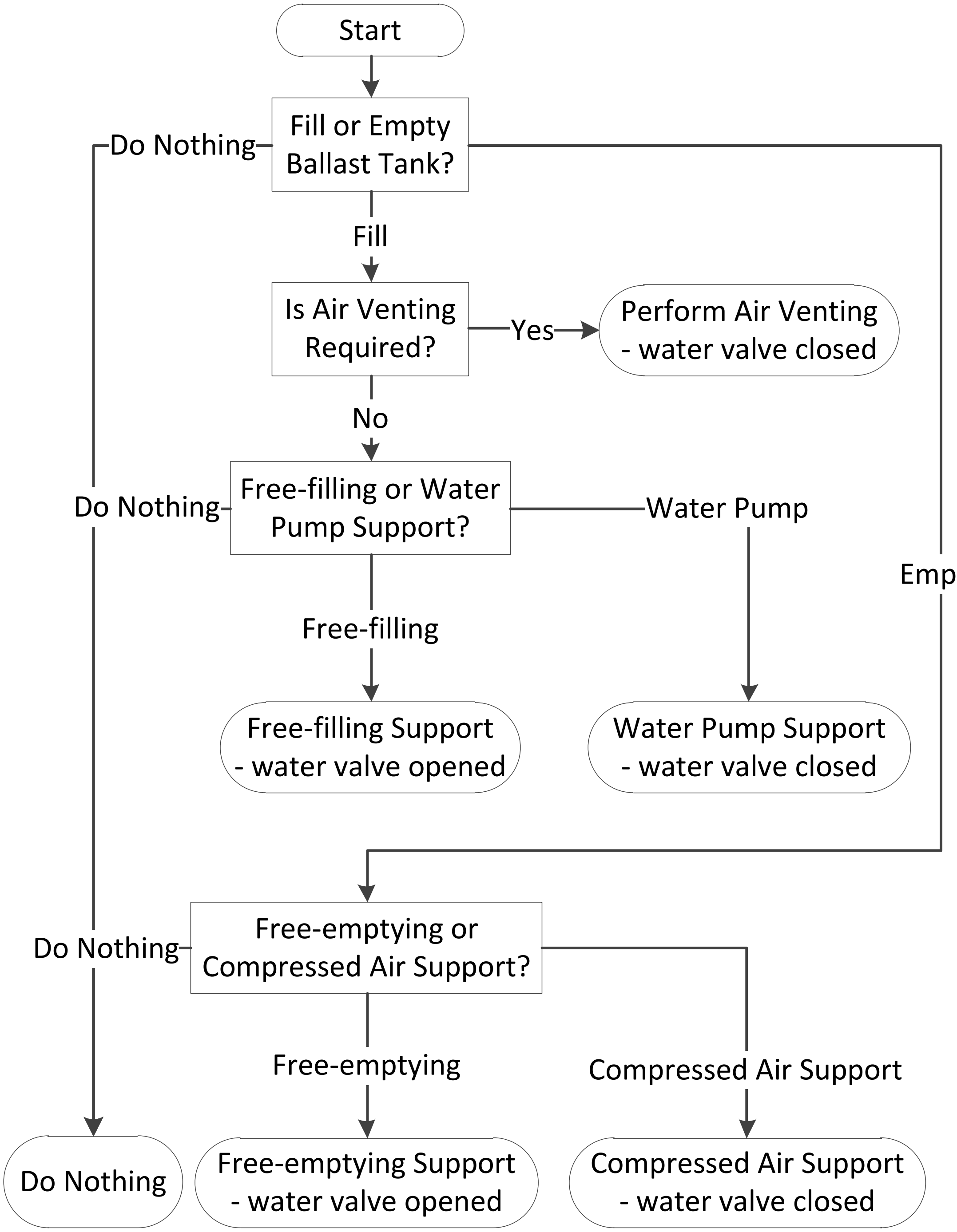
Start
Fill or Empty Ballast Tank?
Do Nothing
Fill
Is Air Venting Required?
Yes
Perform Air Venting - water valve closed
No
Free-filling or Water Pump Support?
Do Nothing
Water Pump
Empty
Free-filling
Free-filling Support - water valve opened
Water Pump Support - water valve closed
Free-emptying or Compressed Air Support?
Do Nothing
Free-emptying
Compressed Air Support
Do Nothing
Free-emptying Support - water valve opened
Compressed Air Support - water valve closed

Figure 5 – Ballast Tank Logic Overview

Table 3 gives a general overview of the controller outputs and can be applied for all VBS-based controllers, with exceptions (i.e. no free-filling for the PD VBS depth controller, as stated).

Table 3 – General Controller Status and Corresponding Outputs

Control Status	Water Pump Rate m^3/s	Air Mass Rate kg/s	Water Valve Area m^2
0 – no action	0	0	0
1 – free-filling	0	0	variable
2 – water pump support	+ variable	0	0
3 – air venting	0	– constant	0
4 – free-emptying	0	0	variable
5 – compressed air support	0	+ variable	0

3.6 Ballast Tank Size

With a maximum diameter of 1.27 m, the Theseus AUV is able to accommodate ballast tanks of various sizes. While the Theseus AUV incorporates spherical ballast tanks, the ballast tank shape used in this thesis is assumed to be cubic for simplification purposes. Depending on the volume of the ballast tank, the ballast control capabilities will vary greatly. The ballast control capability represents the magnitude of weight difference that can be either increased or decreased due to changing the amount of water within the ballast tanks. Figure 6 displays the ballast tank volumes and corresponding control capabilities based on ballast tank size. The ballast tank size is given as the measurement of one side of the cube-shaped ballast tank. The volumes shown in the figure are that of the combined volume of both ballast tanks ($V_{bal\,tank}$) in the AUV where the volume of an individual tank is given by:

$$V_{bal\,tank} = h_{bal\,tank} \cdot w_{bal} \cdot l_{bal\,tank} \tag{22}$$

where $h_{bal\,tank}$ represents the height of the ballast tank, w_{bal} represents the width of the ballast tank, and $l_{bal\,tank}$ represents the length of the ballast tank. The control capability of the AUV is defined as follows:

$$control\ capability = \rho_{wat} \cdot g \cdot V_{bal\,tank}. \tag{23}$$

The control capabilities shown on the right-hand axes of Figure 6 are calculated assuming that the ballast tanks are neutrally buoyant when the ballast tanks are half-filled. Based on Figure 6, a ballast tanks with a combined volume of 2 m^3 will have a combined control capability of approximately 10,055 N. From this neutral buoyancy condition, when both ballast tanks on the AUV are 100% filled, the weight of the AUV will increase by 10,055 N. Likewise, when both ballast tanks on the AUV are 100% emptied, the weight of the AUV will decrease by 10,055 N. Figure 6 shows that the relationship between volume and control capability is linear.

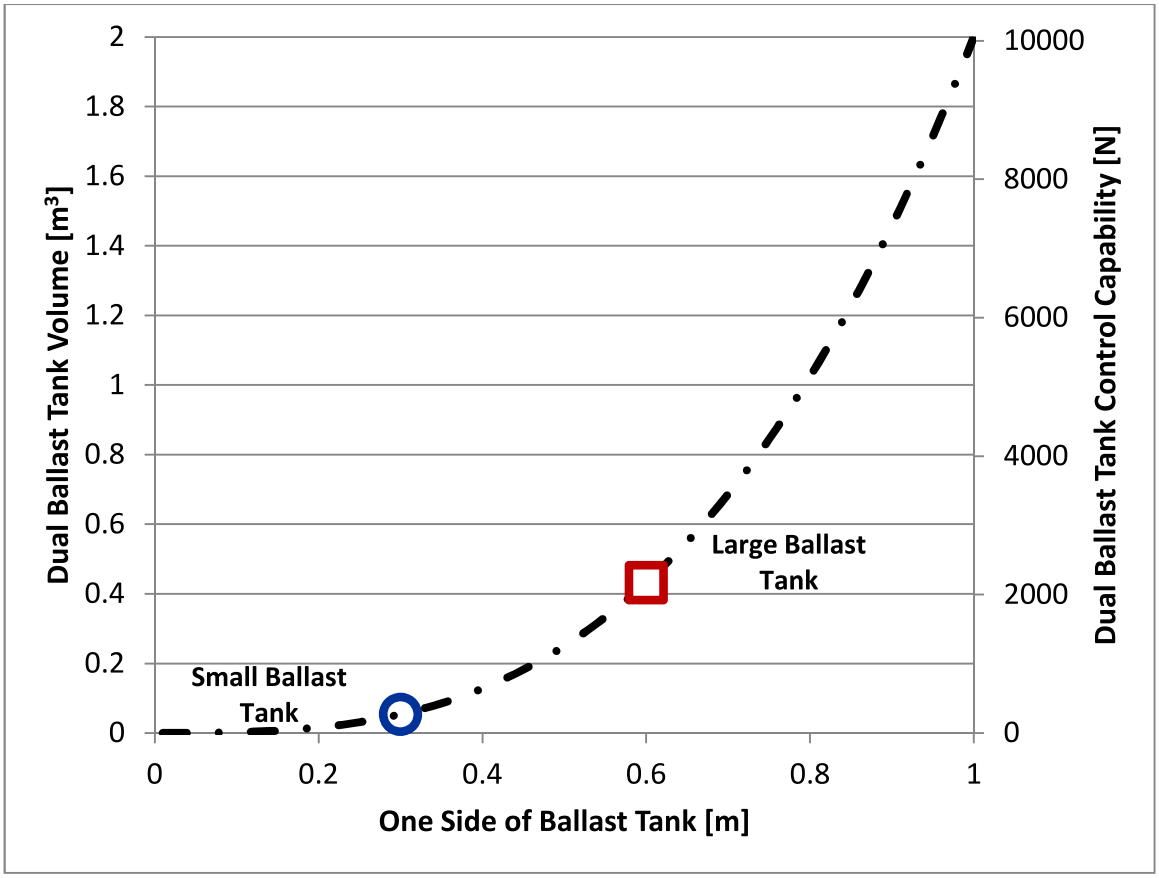

Figure 6 – Dual Ballast Tank Volume [m^3] and Control Capability [N]

As shown by the navy blue circle marker in Figure 6, the "small" ballast tank studied in this thesis exhibits a dual ballast tank volume of approximately 0.054 m^3 with a dual ballast tank control capability of approximately 271 N. Table 4 summarizes additional tank specifications of the small ballast tank used in this research.

Table 4 – Small Ballast Tank Specifications

Parameter	Single Tank	Both Tanks
Total Volume	0.027 m^3	0.054 m^3
Total Mass, 100% Filled	27.68 kg	55.35 kg
Total Weight, 100% Filled (both tanks)	271.49 N	542.98 N
Maximum Air Mass (@ $P_{air\,max} = 50\ MPa$)	16.06 kg	32.12 kg

While conservation of resources is important for long-term sustained AUV performance, there are many factors that can affect the control capability of the ballast tanks including water density changes, temperature changes, and pressure increases due to depth For example, as the water pressure increases due to an increase in depth, so will the amount of air mass necessary for re-establishing neutral buoyancy. To help understand the importance of ballast tank size on VBS control, a second ballast tank, referred to as the "large" ballast tank, was studied in this thesis. With a single ballast tank volume of approximately 0.432 m^3 this large ballast tank has a volume approximately eight times more than that of the small ballast tank. The resulting dual ballast tank control capability is approximately 2,171.93 N as identified by a red square in Figure 6. Table 5 summarizes additional tank specifications of the large ballast tank used in this research.

Table 5 – Large Ballast Tank Specifications

Parameter	Single Tank	Both Tanks
Total Volume	0.216 m^3	0.432 m^3
Total Mass, 100% Filled	110.70 kg	221.40 kg
Total Weight, 100% Filled (both tanks)	1,085.97 N	2,171.93 N
Maximum Air Mass (@ $P_{air\,max} = 50\ MPa$)	128.48 kg	256.95 kg

3.7 Achieving Neutral Buoyancy at Depth

When determining the necessary size of the ballast tank, it is important to know the required air mass to bring the AUV to a neutrally buoyant state for different depths and ballast tank water levels. The following equation shows the mass of air required to achieve neutral buoyancy as a function of depth.

$$mass = \frac{P_{wat@5000m} \cdot V_{air}}{R_{air}T_{air}} = \frac{\left(P_{atm} + (\rho_{wat} \cdot g \cdot depth)\right) \cdot ({w_{bal}}^{2} \cdot h_{wat})}{R_{air}T_{air}} \qquad (24)$$

where $P_{wat@5000m}$ represents the pressure of ocean water at a depth of 5,000 m, P_{atm} represents atmospheric pressure (101,325 Pa), and h_{wat} represents the ballast tank water height. The calculation for the mass of air assumes that there is initially no air in the ballast tanks and that they are 100% filled with ocean water. Figure 7 plots the necessary air mass within each ballast tank that is required to bring the AUV to a neutrally buoyant (when both ballast tanks are *half-filled*) state as a function of AUV depth. As there are two ballast tanks in the AUV, the calculated air mass shown in Figure 7 represents the air mass for both ballast tanks. The dashed navy blue curve represents the *half-filled* case with the "small" ballast tanks ($V_{small\ bal}$ = 0.027 m^3). As can be seen, the total required air mass at an AUV depth of 5,000 m, for the "small" ballast tank, is 17.36 kg. The dashed red curve represents the *half-filled* case with the "large" ballast tanks ($V_{small\ bal}$ = 0.216 m^3). As can be seen, the total required air mass, at an AUV depth of 5,000 m is 130.40 kg. Using a ballast tank with equal sides that are only two times larger (0.60 m vs. 0.30 m) will require an air mass that is eight times greater (130.40 kg vs. 17.36 kg *and* 260.79 kg vs. 34.71 kg) in order to achieve identical neutral buoyancy changes.

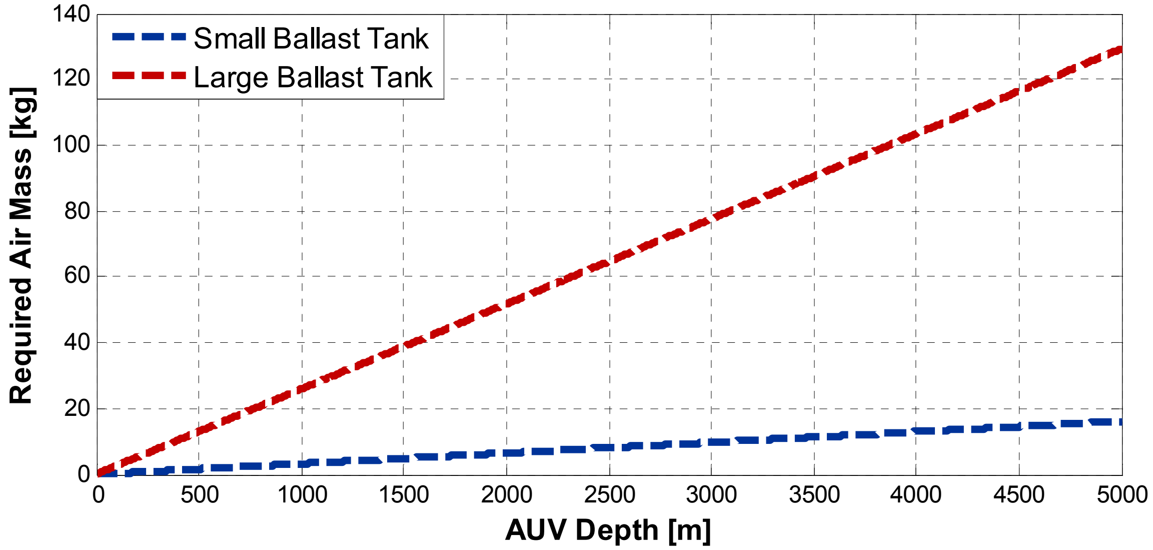

Figure 7 – Required Air Mass for Neutral Buoyancy

Figure 8 shows the mass of air contained inside a compressed air cylinder given a range of internal pressures and cylinder volumes. Figure 8 shows compressed air cylinders with volumes ranging from 0.1 to 1.0 m^3 and cylinder pressures ranging from approximately 1.5 × 10^4 kPa to 5 × 10^4 kPa (15 MPa to 50 MPa) have air masses between approximately 18 kg and 603 kg.

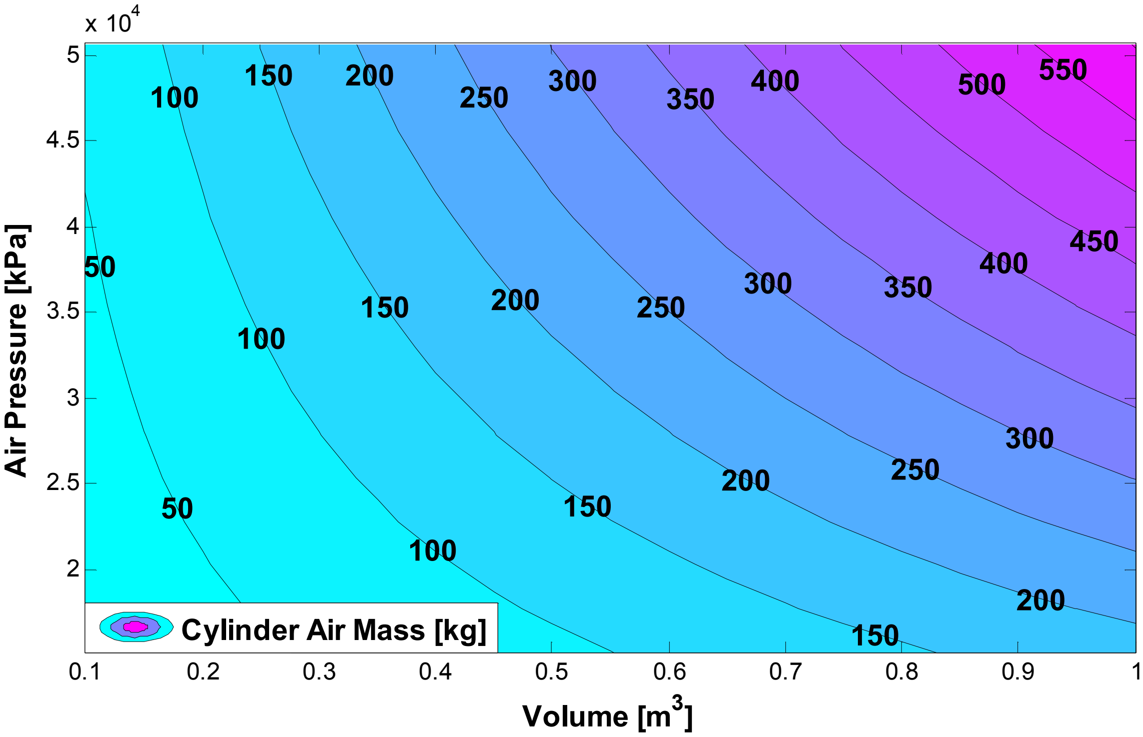

Figure 8 – Air Cylinder Mass

Having knowledge of the size of the AUV and the desired testing parameters (such as maximum depth), Figure 6 and Figure 7 can be used to help select an appropriate compressed air cylinder. For operations dealing with greater depths, for example, Figure 6 and Figure 7 show that large or high pressure compressed air cylinders should be chosen. Also, for situations where a large ballast tank is desired (such as in shifting the center of gravity along the body-fixed *x*-axis), a large or highly pressurized compressed air cylinder is also recommended. Other operations requiring a smaller ballast tank or lower maximum depths can utilize a small volume or lower pressurized compressed air cylinder. The selected compressed air cylinder will be an important design choice for the AUV as it will determine the boundaries of ballast tank controller performance which will be detailed in the next section.

Chapter 4 Simulator Design and Implementation

GENERAL SYSTEM LAYOUT

SIMULATOR DESIGN

AUV AND COMPONENT VALIDATION

Designed using MATLAB® script files, the AUV computer simulator incorporates various controllers, ballast tank systems, dynamics and hydrodynamics, as well as several methods for analyzing the simulated data. This chapter will cover key aspects of the simulator including controller design, simulator validation, system stabilization, and integration.

4.1 General System Layout

Figure 9 illustrates the overall simulator layout developed and used in this thesis.

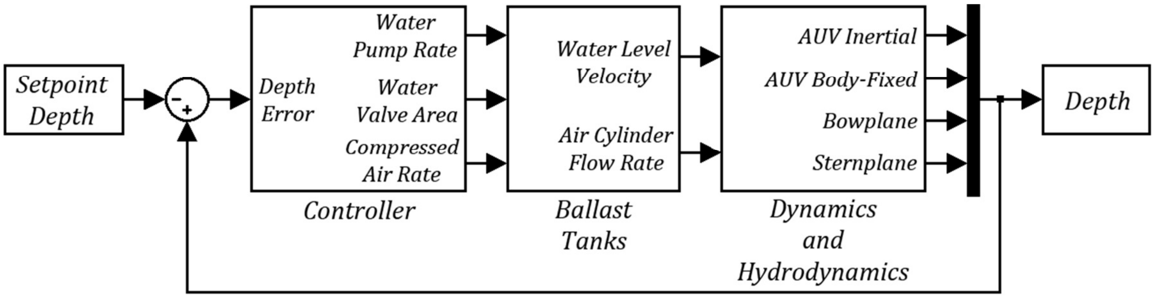

Figure 9 – Overall Computer Simulator Layout

The components shown in Figure 9 are summarized as follows:

- *Water Pump Rate* represents the volume flow rate of the water pump
- *Compressed Air Rate* represents the mass flow rate of the compressed air cylinder
- *Water Valve Area* represents the area of the ballast tank water valve
- *Water Level Velocity* represents the overall ballast tank water height rate of change which includes free-filling, free-emptying, and the water pump
- *Air Mass Velocity* represents the overall ballast tank mass flow rate including the compressed air cylinder and air venting
- *AUV Inertial* represents the *X* and *Z* inertial positions and the pitch angle *(θ)* of the AUV

- *AUV Body-fixed* represents the *u* and *w* body-fixed velocities and the angular velocity *(q)* of the AUV
- *Bowplane* represents the bowplane fin angle of the AUV (both bowplane fins angles are set to be identical for the simulations performed in this thesis)
- *Sternplane* represents the sternplane fin angle of the AUV (both sternplane fins angles are set to be identical for the simulations performed in this thesis)

The following are a list of setpoint parameters that can be controlled either directly or indirectly within the simulator:

- Depth (direct control)
- Ballast Tank Water Height (indirect control)
- Ballast Tank Air Mass (indirect control)
- Ballast Tank Body-fixed x-axis Center of Mass (indirect control).

Direct control refers to setpoint parameters that are chosen *a priori* by the user – these parameters are not determined within the simulator itself. Indirect control parameters are determined within the simulator during simulation by the controller algorithm. The ballast tank water height and air mass, for example, are manipulated based on the depth error of the AUV or the ballast tank water height required, achieving a desired center of gravity along the body-fixed *x*-axis.

4.2 Simulator Design

The simulator was designed using features available in MATLAB® including nested structures, variable-step integrators, event functions and soft saturations, mass matrix functions, and maximum time-steps. Knowledge of these simulator design features will enable a better understanding of the results found within the subsequent sections.

4.2.1 Application Structure

The MATLAB® simulator utilizes a nested structure which results in the following general layout of the simulator code:

```
function auv_simulator
        code (parameter definitions and pre-integration setup code)
        integrator initialization
        code (post-integration and post-event code)
        function integrator
                code
        end
        function mass_matrix
                code
        end
        function events
                code
        end
end
```

The full MATLAB® application code can be found in the Appendix B.

4.2.2 Chosen Integrator

The AUV simulator uses MATLAB®'s variable-step ode45 (Dormand-Prince) integration solver. Tolerances were set at their default values of 1E-3 and 1E-6 for the relative and absolute tolerances, respectively. This integrator incorporates a specific maximum time-step, the variable mass effect and event functions. The control frequency of the AUV, in this simulation, is controlled and determined by the variable time-step of the ode45 integrator.

4.2.3 Event Functions and Soft Saturations

The integrator in this simulator utilizes the MATLAB® build-in event function in order to control over- or under-saturation conditions. While saturation conditions can be accommodated within the Simulink simulation environment using saturation blocks, such an equivalent method does not exist within the MATLAB® script domain and event functions were, therefore, used. The event function checks for values that cross from positive to negative and then stops the ode45 integrator. Once the integrator is stopped, additional function code is then executed. This event code has been developed to reset the simulator parameters to appropriate values in order to correct the over- or under-saturation condition that caused the integration stoppage. The following parameters and corresponding saturation conditions are checked by the event function within this simulator:

▪ Ballast water tank height	< 0	> maximum possible water height
▪ Ballast tank mass of air	< 0	
▪ Bowplane angle	< –25°	> +25°
▪ Sternplane angle	< –25°	> +25°
▪ Ballast tank water valve area	< 0	> maximum water valve area
▪ Ballast tank water pump flow rate	< 0	> maximum water pump flow rate

Looking at the ballast tank water valve area parameter, for example, the following illustrates the need for, and utilization of, saturation through event functions. As the water valve is opening or closing, the integration of the valve's rate-of-change of orifice area (valve velocity) results in a corresponding valve opening area (valve position). Due to the nature of MATLAB®'s variable timestep integration algorithm (via ode45), the exact change in valve area from one specific timestep to another cannot be determined as the timestep is variable. Therefore, it is possible that the change in valve size can depend on the specific time step change that MATLAB® determines. The change in valve area size may be large enough to cause the water valve area to become larger than the maximum allowable area (representing the valve being fully opened). When closing, the valve area change may be large enough to cause the water valve area to become smaller than zero (representing a closed valve). Having the water valve area larger than the maximum allowable area ($A_{wat\,val\,max}$ = 0.01 m^2) or smaller than the minimum allowable area ($A_{wat\,val}$ = 0 m^2) will initiate the *events* function of the simulator to force the integrator to stop and reinitialize with a new water valve area. One solution to reducing the occurrence of these integrator stoppages might be to disallow further integration of the water valve when it is close to being fully opened or closed (i.e. at a valve open position of approximately 10% away from the valve being fully opened or closed). However, unlike the ballast tank water height, which can be left at these pre-determined dead zones, the water valve must be able to completely close in order to fully function properly (e.g. during water pump support). Therefore, to allow actions (such as water pump support) to occur when the water valve is opened, a soft-saturation was implemented in order to allow the water valve to be fully closed when the valve area is within a 10% threshold ($A_{wat\ val\ max}$ = 0.001 m^2). This soft-saturation would also reduce excessive integration stoppage events by disallowing further integrations when the valve width is beyond the 90% threshold of being fully-opened ($A_{wat\ val\ max}$ = 0.009 m^2). For this condition, the soft saturation will also communicate to the simulator that the valve is closed or fully-opened for instances when further valve opening or closing may be desired. The major purpose for these soft saturations is to speed up integration which can be slowed by an over-abundance of integration stoppages. Stoppages consist of stopping the MATLAB® integrator, performing several calculations, and then restarting the integrator at the same point in time where it was stopped. There will be a minor error due to a loss of accuracy due to these soft saturations. Artificially stating that the valve is closed, when it is not, will end up reducing the time for the valve to close. However, with a maximum valve opening time of 2 seconds, this error will be at most 0.2 seconds. When opening, there will not be any perceived loss of accuracy, just a reduction in potential. When attempting to fully-open, the valve area will end up between the threshold ($A_{wat\ val\ max}$ = 0.009 m^2) and maximum ($A_{wat\ val\ max}$ = 0.010 m^2). With the water valve not being able to open to its area will pose a minor limitation on the free-filling or free-emptying output. While not critical, since free-filling and free-emptying are based on pressure differences, as mentioned in section 3.1, this limitation will only cause the valve to require being opened longer in order to compensate fully for this loss of potential.

Each of the other *event* function parameters have their own soft saturations in order to avoid having the integrator stop for insignificant values. As a result, this soft saturation solution prevents the simulation from slowing down due to excessive integration stoppages.

4.2.4 Mass Matrix Function

MATLAB®'s mass matrix function is also being utilized within the ode45 integrator. This function is external to the ode45 integrator function and is used to calculate the AUV mass matrix at each integration time step. This function is needed because the mass of the ballast tanks will change at each step due to changes in the water levels within.

4.2.5 Maximum Sample Time

When choosing to perform a simulation, the maximum sample time must be taken into consideration. For the ode45 integrator, which utilizes a variable time step, the maximum sample time represents the largest possible time between one integration time step and another. To investigate the effect of changes to the maximum sample time, a sample AUV proportional derivative (PD) depth control simulation (K_P = 1, K_D = 40) was carried out for three different sample times: 0.01 sec, 0.10 sec, and 1.00 sec. The AUV was commanded a step change in depth from 100 m to 130 m and the resulting water and air resources used over 100 seconds were predicted.

Table 6 summarizes the total simulation time and number of integration time steps for each of the three maximum sample times tested. It should be noted that the total simulation time shown in Table 6 also includes post processing of the simulation results, so that an increase in the amount of time steps will have a significant impact on the total simulation time. Table 6 shows that decreasing the maximum sample time from 0.10 sec to 0.01 sec increases the total simulation time and number of time steps by a factor of 10

Table 6 – Maximum Sample Time – Total Simulation Time

Maximum Sample Time sec	Total Simulation Time sec	Number of Time Steps
0.01	1,673	400,153
0.10	166	40,249
1.00	28	5,561

(Notebook setup: Intel® Core™2 Quad Processor Q9550 @ 2.83 GHz, 4.00 GB RAM, dual nVidia 9800m GT video cards in SLI, and Windows Vista x64) Figure 10 plots the resulting water volume usage and air mass usage for an AUV utilizing a PD depth controller (K_P = 1 and K_D = 40) in response to a 30 m step change in depth for the three sample times listed in Table 6. As can be seen from Figure 10 the maximum sample time can have a significant impact on the accuracy of the predicted water volume and air mass resource usage within the simulator. Using a maximum sample time of either 0.01 sec or 0.10 sec enables the simulator to predict less than 0.1 m^3 total water volume usage and approximately 0.3 kg air mass usage, while an increase to 1 second results in an erroneous prediction in the total water volume usage to approximately 0.7 m^3 and total air mass usage of approximately 0.5 kg. Given the significant time savings, observed in Table 6, and the fact that Figure 10 shows virtually identical simulation results for maximum sample times of 0.10 sec and 0.01 sec, a maximum sample time of 0.10 sec was selected for all the simulations carried out in this thesis.

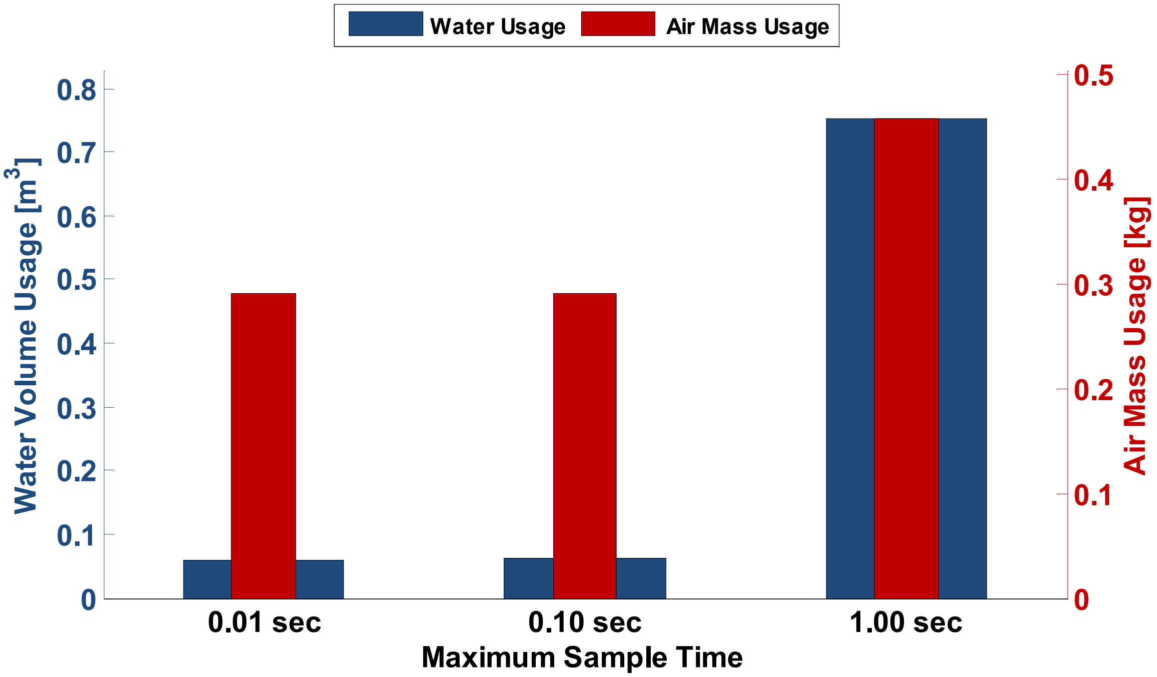

Figure 10 – Simulation Time Comparison (PD VBS Controller, K_P = 1, K_D = 40, t = 1000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

4.3 AUV and Component Validation

In order to determine whether or not the AUV simulator and its components performed as intended, tests on the following component validation tests were carried out:

- AUV Dynamics & Hydrodynamics
  - Descent and Ascent
  - Orientation (Pitch Angle)
  - Bowplane and Sternplane Fins
  - Hydrodynamics
- Ballast Tank
  - Free-emptying and Free-filling
  - Water Pump Support
  - Compressed Air Support.

Details of intended performance and corresponding results of the successful component validation tests can be found in the Appendix C. The following chapter describes the various controllers that were developed and tested using the AUV simulator including a bowplane and sternplane proportional depth and pitch controller, a proportional derivative (PD) VBS (variable ballast system) depth controller, a hybrid proportional derivative condition-based (HPDCB) VBS depth controller, and an x_G shifting (trim) controller.

Chapter 5 Overview of Controllers

In this thesis, four different controllers were developed and tested: a bowplane and sternplane proportional controller, a proportional derivative (PD) VBS depth controller, a hybrid PD condition-based (HPDCB) VBS depth controller, and a VBS x_G shifting controller. Of these four controllers, the HPDCB VBS depth controller and VBS x_G shifting controller are unique contributions. Table 7 summarizes these four controllers along with the simulation variables they attempt to control:

Table 7 – Controllers and Control Variable

Controller Name	Control Variable
Bowplane & Sternplane Proportional Depth and Pitch Controller	Depth and Pitch
Proportional Derivative (PD) VBS Depth Controller	Depth
Hybrid PD Condition-based VBS Depth Controller	Depth
VBS x_G Shifting Controller	x_G Shift

For the 2D simulations presented in this thesis, the desired depth of the AUV is termed the "setpoint depth". In a typical mission scenario, the setpoint depth may vary during the operation of the AUV. The actual depth of the AUV is termed the "feedback depth". The difference between the setpoint and feedback depths represents the depth error e_{depth} of the AUV:

$$e_{depth} = Setpoint\ Depth - Feedback\ Depth. \tag{25}$$

The feedback depth of the AUV, at any given time, is the Z-axis (inertial) coordinate of the AUV at its origin r_B, as shown in Figure 11.

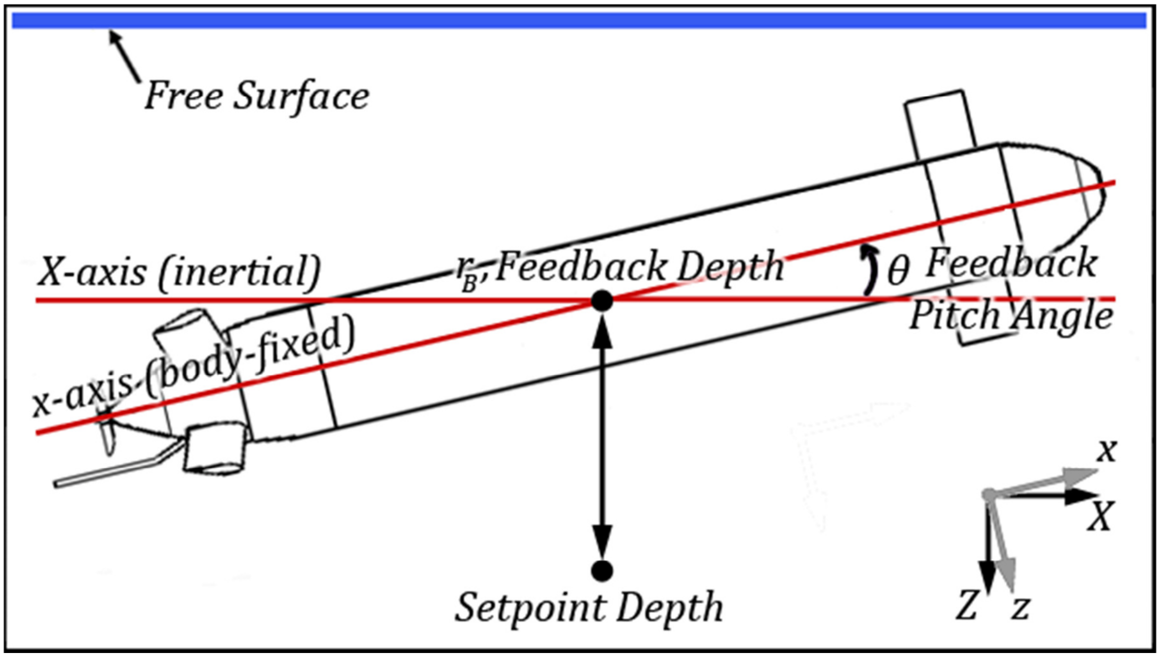

Figure 11 – Depth Error and AUV Layout [7]

The "feedback pitch angle" represents the actual pitch angle of the AUV. The difference between the feedback pitch angle and the setpoint pitch angle represents the pitch error e_{pitch} of the system, given as:

$$e_{pitch} = Setpoint\ Pitch\ Angle - Feedback\ Pitch\ Angle. \tag{26}$$

In many situations, the pitch angle will change as a direct result of the nonlinear hydrodynamic forces and moments acting on the AUV as the vehicle changes depth.

5.1 Bowplane and Sternplane Proportional Controller

The first control system discussed in this thesis is the bowplane and sternplane proportional controller which uses separate proportional control gains for the bow- and sternplanes. This type of controller is the current method used to control depth and pitch on the Theseus AUV. Located in the middle (roughly midship) of the AUV, the bowplane fins are used to control the depth of the AUV given an adequate relative forward velocity in the fluid. The higher the relative AUV velocity, the larger the effect (or control authority) that the deflections of the bowplane fins will have on controlling the depth of the AUV. The sternplane fins are located at the aft end of the AUV. They are best suited for controlling the pitch angle of the AUV. The control capability of the sternplane increases as the forward velocity of the AUV increases. The bowplane and sternplane proportional controller does not attempt to vary the contents of the ballast tanks while adjusting the AUV depth and pitch, and this proportional controller will be covered in-depth in Chapter 6.

5.2 Proportional Derivative (PD) VBS Depth Controller

One approach to incorporating ballast tanks into the control system is to employ a PD VBS depth control system. The proposed PD VBS depth control system monitors the PD controller output to determine whether to fill or empty the ballast tanks. For example, if the controller output is positive, then the water pump will be used to fill the ballast tank and cause the AUV to sink. Similarly, if the controller output is negative, then the compressed air support will begin to raise the mass of air (and corresponding air pressure) in the ballast tank. Once the mass of air is increased to the point where the air pressure is at least equal to the pressure of the external ocean water, the ballast tank water valve will open so that the water ballast can be emptied causing the AUV to rise. Due to the relatively simplistic nature of this proposed control approach, free-filling could not be implemented for this depth controller. The proposed PD VBS controller will be covered in-depth in Chapter 7.

5.3 Hybrid PD Condition-Based VBS Depth Controller

To enable free-filling, an alternative VBS controller is proposed which examines the current state of the ballast tank to determine the appropriate control action. This new hybrid proportional derivative condition-based (HPDCB) VBS depth controller calculates two values to determine which methods of ballasting are feasible: the "capable controller output for free-filling" $u_{cap,f}$ and the "capable controller output for free-emptying" $u_{cap,e}$. Using knowledge of the current ballast tank system state and the controller output, the ballast system is instructed to perform free-filling, free-emptying, water pump, and compressed air support as needed. Unlike the PD VBS depth controller that relies solely on the controller output, the proposed additional control logic enables free-filling and attempts to minimize resource usage by, for example, adding controller dead zones. The proposed HPDCB VBS depth controller will be covered in-depth in Chapter 8.

5.4 VBS x_G Shifting Controller

One situation where the center of gravity (x_G) could shift is when cable is being laid. The loss of mass, in the portion of the AUV where the cable is being unloaded, will cause x_G to shift along the body-fixed x-axis to a new location. When the center of gravity (x_G) of the AUV shifts forward or aft, there is change in pitch angle, likely away from the desired setpoint pitch angle. Typically when this situation occurs, the bow- and sternplanes act to try to correct the change in pitch angle to reachieve the desired setpoint pitch angle. Such a control approach can result in the fins being unable to provide their full control authority towards controlling the AUV's depth and pitch angle. The fins may not be able to provide their full control authority when depth or pitch angle offset is too large that equilibrium is not possible. To help overcome this reduction in AUV planes' control authority, a VBS x_G shifting controller is proposed in this thesis to try to shift the center of gravity along the body-fixed x-axis back toward the center of buoyancy in an attempt to enable the sternplane to perserve its control authority for small changes in depth and pitch. There is drag associated with the bow- and sternplane fin deflections. By shifting the center of gravity, the requirement for using the bow- and sternplane fins to continuously achieve the preferred trim will decrease. Reducing bow- and sternplane fin usage, where possible, will result in increased AUV and planes endurance. The proposed VBS x_G shifting controller will be covered in-depth in Chapter 9.

Chapter 6 Bowplane and Sternplane Proportional Depth and Pitch Controller

Autonomous underwater vehicles typically use a bow- and sternplane controller while in operation, without adjusting the water level in any internal ballast tanks. The effectiveness of these deflection control surfaces is simulated in this chapter to provide a baseline for further comparison. In this thesis, the bowplanes and sternplanes are adjusted using a proportional controller to control depth and pitch.

6.1 Design of Bowplane Proportional Controller

When designing a bowplane proportional controller, it is important to understand the effect the fins have on the AUV for different propulsion forces and forward velocities. It is also important to choose an appropriate proportional control parameter to achieve a balance between small steady-steady depth error and small depth error overshoot and oscillation. Lastly, it is important to determine whether or not the AUV can sustain a setpoint depth. It should be noted that all simulations of the bow- and sternplane controller in this thesis have the fin angle quantized to ±1°. This quantization has the effect of both improving simulator speed and incorporating sensor resolution into the simulations. For these tests, the depth quantization is within 1 m. The units for the proportional (K_P) control parameters for the bowplane $\left(K_{P,bow}\right)$ are $\frac{depth\ degree\ deflection}{meter\ depth\ error} = \frac{rad}{m}$ and for the sternplane $\left(K_{P,ste}\right)$, $\frac{pitch\ degree\ deflection}{pitch\ angle\ error} = \frac{rad}{rad}$ (unitless). It should be noted that these units have *not* be displayed throughout this thesis. Setpoint bow- (*db*) and sternplane (*ds*) angles are one of the most important controller aspects of the bow- and sternplane depth and pitch controller. The setpoint bow- and sternplane angles are defined as follows:

$$db_{setpoint} = -K_{P,bow} \cdot e_d(t) \tag{27}$$

$$ds_{setpoint} = -K_{P,ste} \cdot e_p(t) \tag{28}$$

where e_d represents the depth error of the AUV, $e_p(t)$ reprents the pitch angle error of the AUV, $K_{P,bow}$ represents the proportional parameter of the bowplane portion of the controller, and $K_{P,ste}$ represents the proportional parameter of the sternplane portion of the controller.

Figure 12 plots AUV simulated depth as a function of time using a bowplane proportional controller (K_P = 0.1) in response to a 30 m step increase in depth for various body-fixed AUV *mean* forward velocities $\overline{u}$. For the simulation plotted in Figure 12, the AUV is neutrally buoyant. These velocities are calculated as the mean velocity of the AUV over the entire 1000 sec simulation and correspond to propulsion forces of 50 N, 250 N, 500 N, and 750 N for mean forward velocities $\overline{u}$ of 0.48 m/s, 1.11 m/s, 1.58 m/s, and 1.95 m/s, respectively. As shown in Figure 12, the higher the AUV velocity, the faster the AUV can (starting at a depth of 100 m) reach the desired 130 m setpoint depth using the fins. It can also be seen that as the AUV velocity decrease, the maximum overshoot and number of oscillations in response to a 30 m step change in depth decreases.

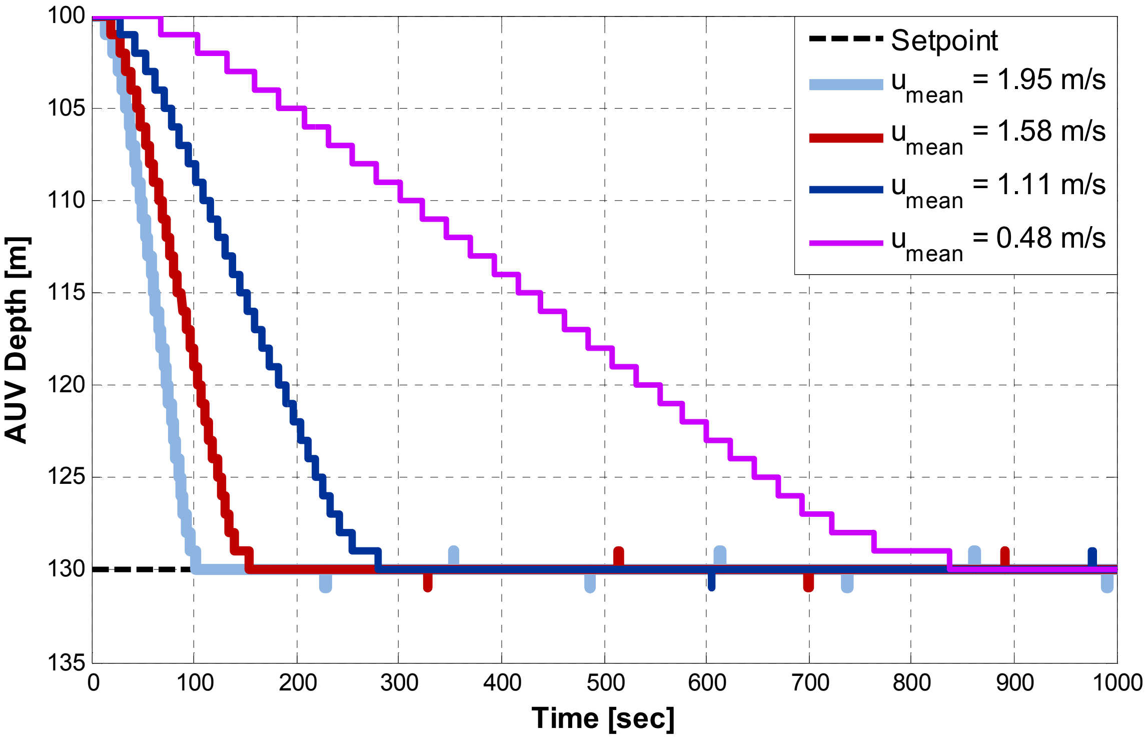

Figure 12 – Bowplane (only) Proportional Depth Control (K_P = 0.1)

The proportional controller output is a function of the depth error and proportional constant K_P. In order to determine an appropriate value for K_P, several simulations were carried out for different values of K_P using a forward propulsion force of 250 N (corresponding to 1.11 m/s, in Figure 12). In the simulation the AUV has an initial depth of 100 m and a setpoint depth of 130 m. The resulting depth error curves are plotted in Figure 13 as a function of time for K_P values of 0.01, 0.10 and 1.00. As can be seen in this figure, values for K_P of 0.01 and 0.10, represented by the dashed red and navy blue curves respectively, appear to provide reasonable depth control response time and negligible pitch angle oscillation. From the values for K_P of 0.01 and 0.10, the value of 0.10 was chosen as the proportional control parameter for the bowplanes results in this thesis due to the extra damping it provides as shown in Figure 13 by the fact that it took slightly longer to achieve setpoint depth. Since a forward AUV velocity of approximately 2 m/s will require more damping than 1.11 m/s (for the simulation in Figure 12 and Figure 13), the extra damping provided by having a K_P of 0.10 is desirable.

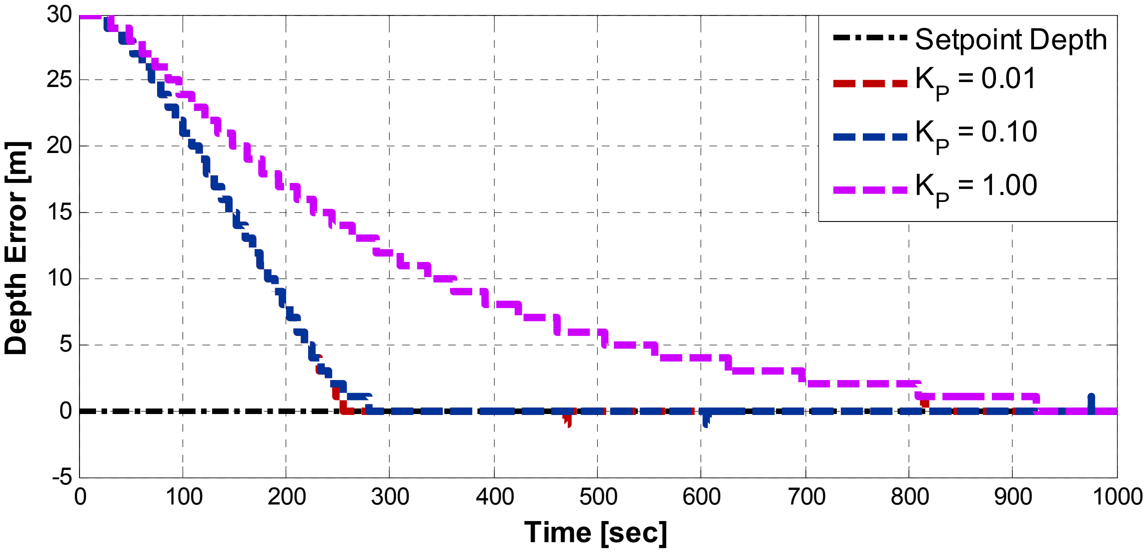

Figure 13 – Bowplane (only) Depth Control for a Range of K_P Values (F_{prop} = 250 N)

Figure 14 plots the resulting AUV depth error as a function of time over a 2000 sec period for an AUV travelling at a mean forward velocity of 1.12 m/s. This figure clearly shows that the bowplane proportional controller, in the absence of disturbance, has the ability to effectively sustain a setpoint depth without the need for a derivative and/or integral controller component.

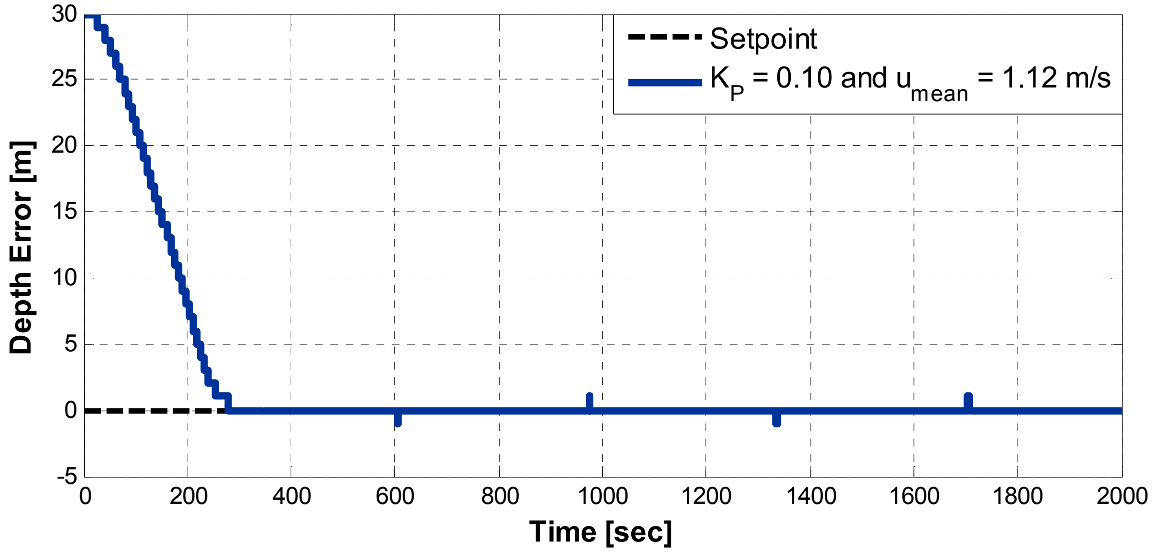

Figure 14 – Bowplane (only) Control – Sustained Setpoint Depth

6.2 Design of Sternplane Proportional Controller

Figure 15 plots the AUV pitch angle response as a function of time in response to a 10° step change in pitch angle as a function of time for different proportional controller gains K_P. To generate these simulation results, a propulsion force of 750 N was applied to the AUV and the pitch angle and sternplane angle sensors were quantized to 1°. As seen, with a K_P value of 8, represented by the solid navy blue curve, the corresponding pitch angle error is lower than when K_P is set to either 5 or 1 (represented by the solid red and magenta curves, respectively). It should be noted that, with K_P values larger than 8, integration tolerance errors within MATLAB® may occur and cause the simulation to terminate. From these results, a K_P value of 8 has been chosen as the proportional controller parameter for the sternplane results in this thesis, without the need for a derivative and/or integral controller component.

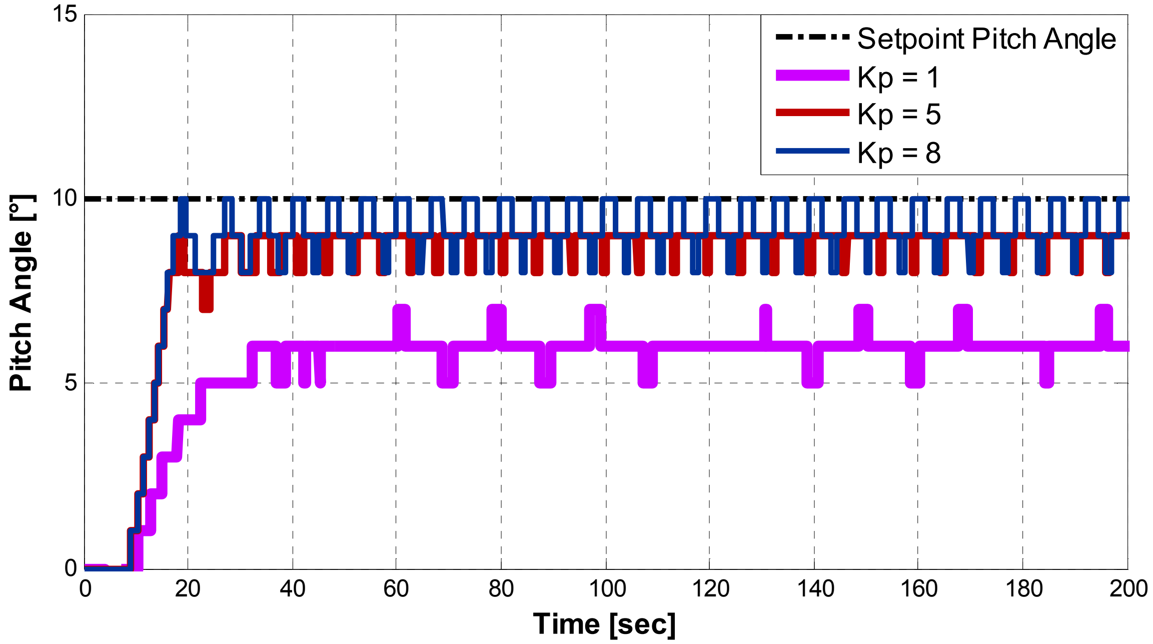

Figure 15 – Sternplane (only) Control – Pitch Angle Error for a Range of K_P Values

6.3 Combined Bowplane and Sternplane Proportional Controller

Figure 16 plots the AUV depth and pitch angle response as a function of time in response to a 30 m step change in AUV depth (from 100 m to 130 m) as a function of time various forward AUV ($K_{P,bow}$ = 0.1 and $K_{P,ste}$ = 8) velocities. The setpoint pitch angle is zero in order to create a baseline at which any changes in pitch angle control response can be observed in later simulations. It can be seen from Figure 16 that, as the AUV velocity increases, the effectiveness of the control surfaces to control the depth increases as expected due to the relationship between forward AUV velocity u and bow- and sternplane angles as can be seen in section 2.1. Figure 16 also shows that the pitch angle begins to exhibit oscillatory motion as the AUV increases its velocity. It is possible that this oscillation could be reduced with the addition of a derivative control term for the sternplane portion of the controller. However, as the bow- and sternplane controller was mainly used as a baseline for comparison of the VBS-based controllers and the oscillations were not comparatively large (approximately 1°), a derivative term was not added.

Referring to Figure 16, at lower forward AUV velocities (such as 0.48 m/s), the AUV takes approximately 780 sec to achieve the desired setpoint depth of 130 m from an initial depth of 100 m. At higher velocities, the AUV is able to achieve the desired setpoint depth faster and, at 1.95 m/s, it takes approximately 170 sec for the AUV to achieve the desired 30 m change in depth. Figure 16 shows that, for the conditions used in the simulation, the bowplane and sternplane controller can sustain a setpoint depth with no steady-state depth error (the steady-state depth error is approximately ±0.5 m without depth or pitch quantization).

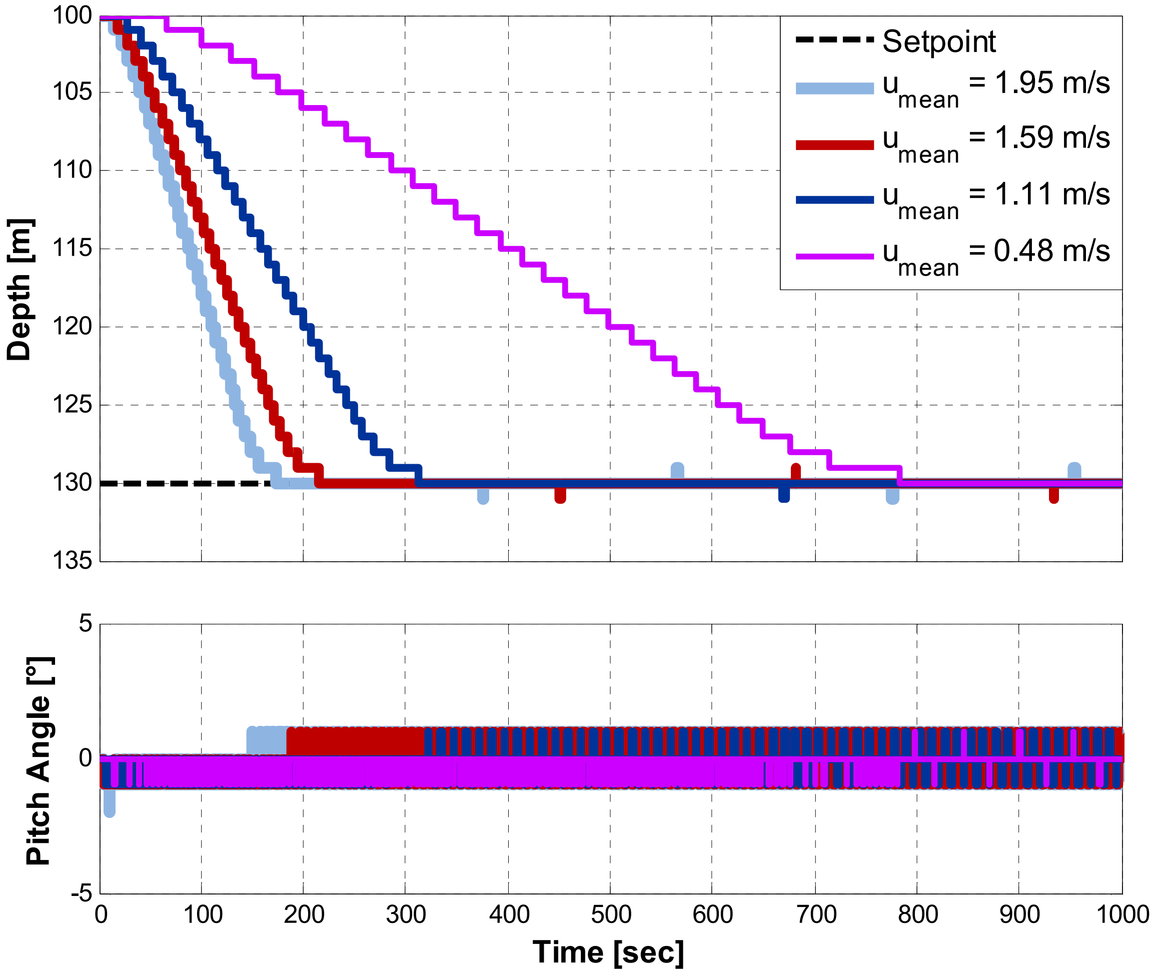

Figure 16 – Bowplane & Sternplane Proportional Controller at Various Forward AUV Velocities ($K_{P,bow}$ = 0.1, $K_{P,ste}$ = 8)

To study the effect of shifting the center of gravity of the AUV, Figure 17 plots the AUV depth and pitch angle as a function of time in response to a step change of 30 m. Figure 17 shows both a –0.0128 and 0.0128 m shift in center of gravity x_G along the body-fixed *x*-axis from the reference center of the AUV, as represented by the solid red and navy blue curves, respectively. Simulation results shown in Figure 17 have the body-fixed z-axis z_G center of gravity located at 0.075 m below the reference centerline of the AUV. As described in section 2.2, this z_G center of gravity location was selected in order to enhance the effect of shifting x_G along the body-fixed x-axis (versus the initial z_G of 0.250 m) since the further z_G is below the z-axis center of buoyancy z_B, the more stable the AUV pitch will be (the passive BG recovery to a pitch perturbation). Figure 17 shows that a positive shift in center of gravity x_G causes the AUV to maintain a steady state depth that is approximately 3 to 4 m lower than the setpoint depth. It can also be observed from Figure 17 that, with a negative shift in x_G, the AUV does not achieve steady state depth (130 m) within the 1000 second simulation. As shown in Figure 17, when positively shifted, the steady state pitch angle is approximately –1° to –2° (setpoint pitch = 0°); when negatively shifted, the steady state pitch angle is approximately 1° to 2°.

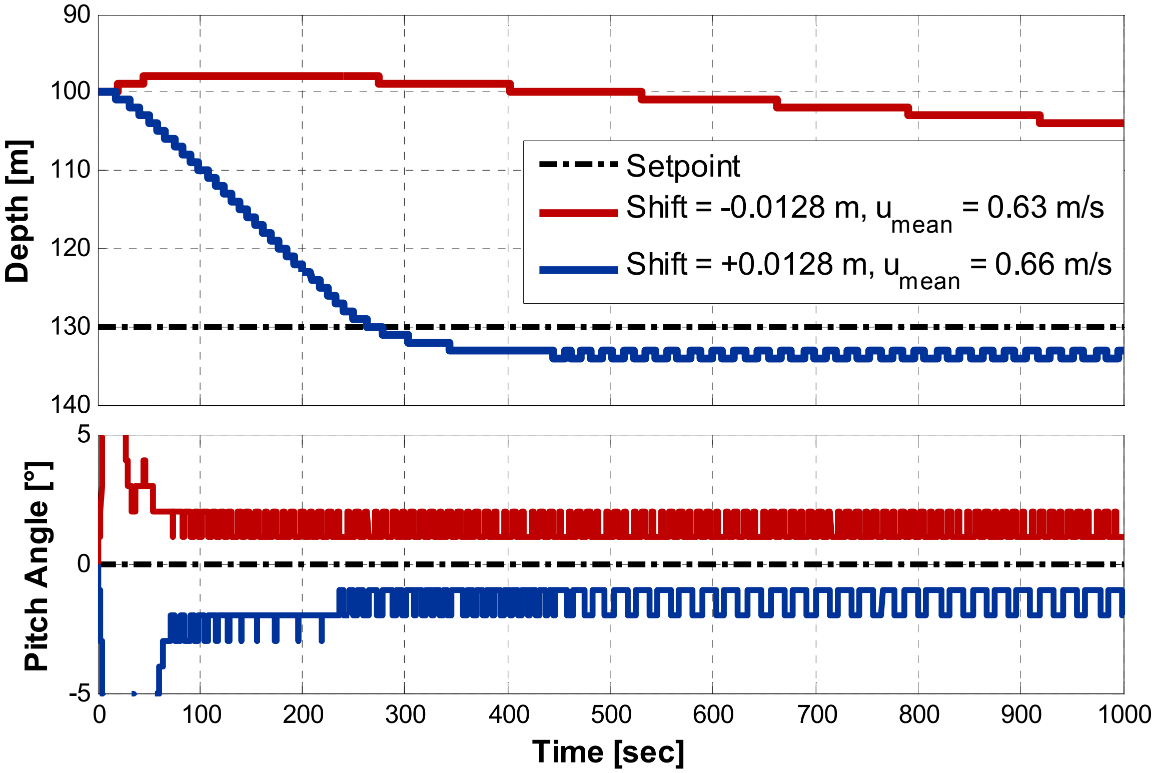

Figure 17 – Bowplane & Sternplane Controller with x_G Shifting ($K_{P,bow}$ = 0.1, $K_{P,ste}$ = 8)

While 3 to 4 m is not a large depth error offset compared with a maximum depth of 5,000 m, these results quantify a reduction in the bow- and sternplane proportional controllers' ability to reduce the depth error of the AUV when a positive or negative shift in x_G exists (compared to when there is no shift in x_G). Another detrimental effect of shifting x_G is that the setpoint pitch angle of the AUV cannot be sustained.

According to Butler & Hertog [13], an the Theseus travelling at 4.1 knots (2.11 m/s) requires 3,100 Watt (W) of power for propulsion and requires 4,400 W of power for VBS at a depth of below 200 m. It takes approximately 100 sec to achieve a setpoint depth change of 30 m when the AUV is moving at a mean forward velocity of 1.95 m/s. For a “small” ballast tank, it takes approximately 1.5 (water pump support @ $Q_{wat\,pump}$ = 0.01 m^3/s) to 30 sec (free-filling @ 1,000 m with $P_{air,initial}$ = 0.1 MPa) to fill the tank from approximately 0.15 to 0.30 m. More details about these values can be found in Appendix C. Since one Watt is defined as one Joule per second, therefore the total energy consumption for using the bowplane and sternplane controller will be approximately 310 kJ and using the VBS will require between 7 and 132 kJ. It is likely that the power requirement for different aspects of the VBS varies. The water pump and compressed air support requires that the water valve be closed where as free-filling and free-emptying requires only motion of the water valve. Therefore, it is assumed that the power requirement of the water pump and compressed air support will be the highest of the VBS.

One major benefit of the bowplane and sternplane proportional controller is that it does not require compressed air or water pump resources. When the center of gravity is not shifted, the controller is able to provide adequate depth and pitch angle control; however, the bowplane and sternplane controller requires an adequate forward AUV velocity (or propulsion force, F_{prop}) in order be able to effectively control both the depth and pitch of the AUV. In situations where the AUV is moving slow (i.e. less than about 1.00 m/s, which will be demonstrated in Chapter 8, Figure 33 is the bowplane and sternplane control equivalent to a VBS over the same distance), or it is desired to achieve setpoint depth of the AUV while conserving (or reducing the usage of) the AUVs battery, a VBS control method may be an option. One such VBS control system is proposed in the following chapter that employs a PD VBS depth controller in conjunction with dual ballast tanks.

Chapter 7 Controlling the AUV's Depth with a PD VBS Controller

7.1 Developing the *"Vanilla"* PD VBS Depth Controller

In developing the proportional derivative (PD) VBS depth controller, it is important to first develop the baseline *"vanilla"* controller that will be used as a comparison for all other depth controllers proposed in this thesis.

Controller output (u_{out}) is one of the most important controller aspects used to develop this "Vanilla" controller where the PD VBS depth controller output is defined as follows:

$$u_{out} = \left(K_P \cdot e_d(t)\right) + \left(K_D \cdot de_d(t)/dt\right) \tag{29}$$

where e_d represents the depth error of the AUV, $de_d(t)/dt$ reprents the rate of change in depth, K_P (units $= 1/m$) represents the proportional parameter of the controller, and K_D (units $= s/m$) represents the derivative parameter of the controller. The parameter u_{out} is used in calculations involving setpoint water valve opening width, setpoint water pump volume flow rate, and setpoint compressed air mass flow rate.

7.1.1 <u>Proportional Control Parameter</u>

For simulations involving the PD VBS depth controller, it should also be noted that the air and water contents of the forward and aft ballast tanks are assumed to be symmetric at all times during the simulation. Also, the initial ballast tank water height is not taken into consideration in any resource usage-related results.

Figure 18 provides a visual representation of the water (volume, m^3) and air (mass, kg) resource usage for five K_P values tested over a 1000 sec simulation. The different colored bars in the figure represent the total water and air resources required and subdivide these resources into those arising from the water pump, free-emptying, free-filling, compressed air cylinder, and air-venting.

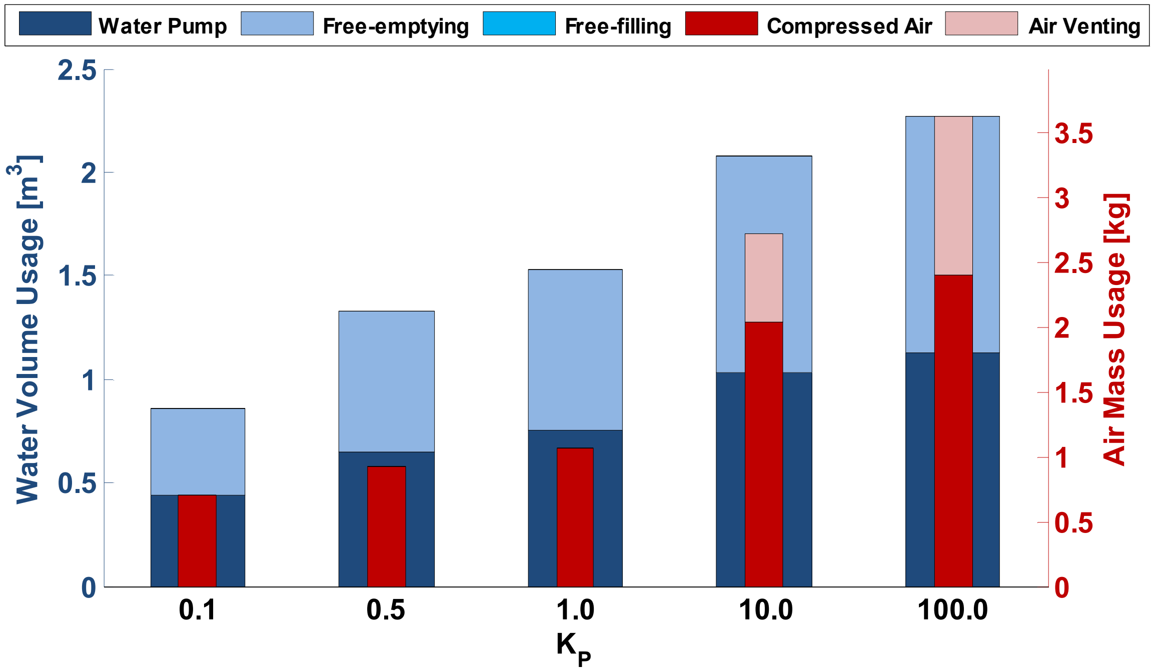

Figure 18 – Proportional VBS Depth Controller – Resource Usage for Various K_P Values (K_D = 0, t = 1000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

The water pump amount shown in Figure 18 describes the total volume of water that was added to the ballast tanks from the ocean via the water pump over the course of the simulation. The free-emptying amount represents the total volume of water leaving the ballast tank through the water valve due to a positive pressure difference ($P_{wat} > P_{air}$) between the internal ballast tank and ocean pressure. Likewise, the free-filling amount, which in Figure 18 is zero, represents the total water volume that entered the ballast tank through the ballast tank's water valve due to a negative pressure difference ($P_{wat} < P_{air}$) between the internal ballast tank and ocean pressure. (Recall that the proposed PD VBS depth controller does not have free-filling capability.) The compressed air amount indicates the total air mass that was added to the ballast tank from the compressed air cylinder, while the air venting amount represents the mass of air that has exited the ballast tank into the ocean over the course of the simulation. As shown in Figure 18, a K_P value of 0.1 provides the lowest resource usage with a water pump usage of 0.44 m^3, a free-emptying water usage of 0.42 m^3, and a compressed air mass usage of 0.70 kg. The largest resource usage is for a K_P value of 100.0 with a water pump usage of 1.13 m^3, a free-emptying usage of 1.15 m^3, and a compressed air mass usage of 2.39 kg. As shown in Figure 18, the proportional controller with K_P values of 0.10, 0.50, and 1.00 utilized the least amount of water and air mass usage over the course of the 1000 second simulation. Figure 19 plots the AUV depth as a function of time in response to a step change of 30 m in setpoint depth. Simulations of setpoint depth control with K_P values of 0.10, 0.50, and 1.00 are shown in Figure 19 representing the K_P values which utilized the lowest water and air mass usage from Figure 18. As shown in Figure 19, a K_P of 1.0, represented by the dotted red curve, has the lowest initial overshoot and the lowest magnitude for oscillations and was chosen as a suitable value for K_P. The next section tests different K_D values for the baseline "Vanilla" PD VBS depth controller.

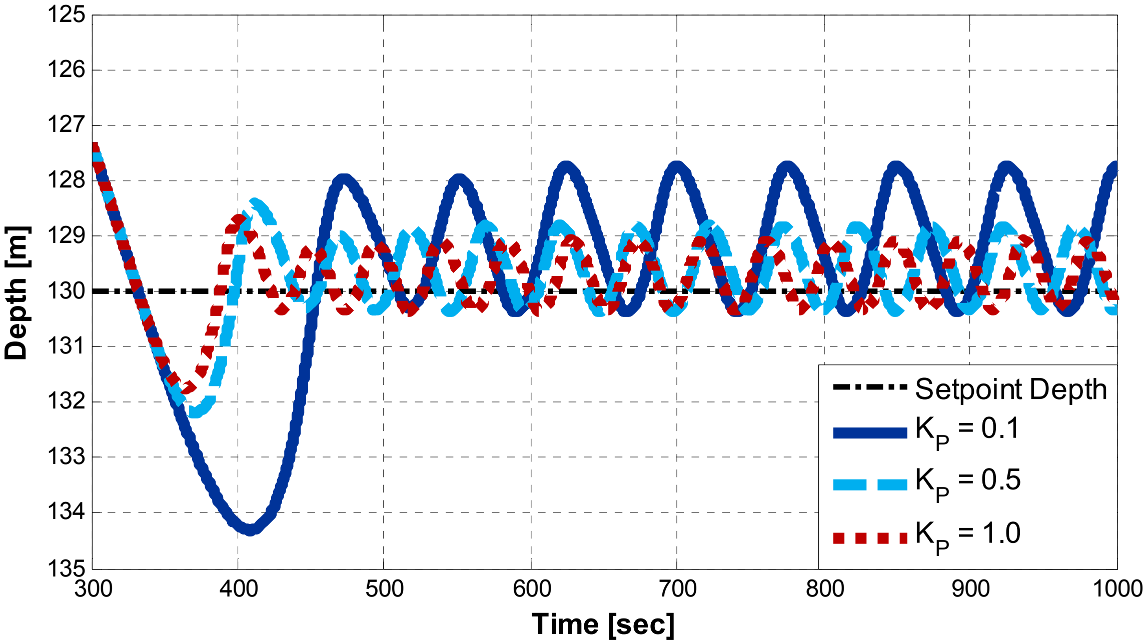

Figure 19 - Proportional (P) VBS Depth Control (K_D = 0, step change in depth from 100 m to 130 m, "small" ballast tanks)

7.1.2 Derivative Control Parameter

Derivative control was added to the "*Vanilla*" PD VBS Depth Controller in order to try to reduce the continuous oscillations that occur in each of the proportional-only simulations of Figure 19, while having a minimal effect on the time it takes to initially reach the setpoint depth. The AUV was again commanded a step change in depth from 100 m to 130 m and the resources used over 1000 sec were predicted. Figure 20 plots the resulting water and air resource usage for a K_P value of 1.0 and a range of K_D values. It can be seen in Figure 20 that K_D values of 40 and 100 show a significant reduction in resource usage when compared with smaller values of K_D (0.1, 1.0, and 10.0). As can be observed in Figure 20, with a K_P value of 1.0 and a K_D value of 40.0 and 100.0, there is relatively low and nearly identical resource usage: 0.03 m^3 for both the water pump and free-emptying, reduced by approximately 96 to 97% for K_D values of 0.1, 1.0, and 10.0. There is a usage of 0.29 kg of compressed air, reduced by approximatey 72% for K_D values of 0.1, 1.0, and 10.0.

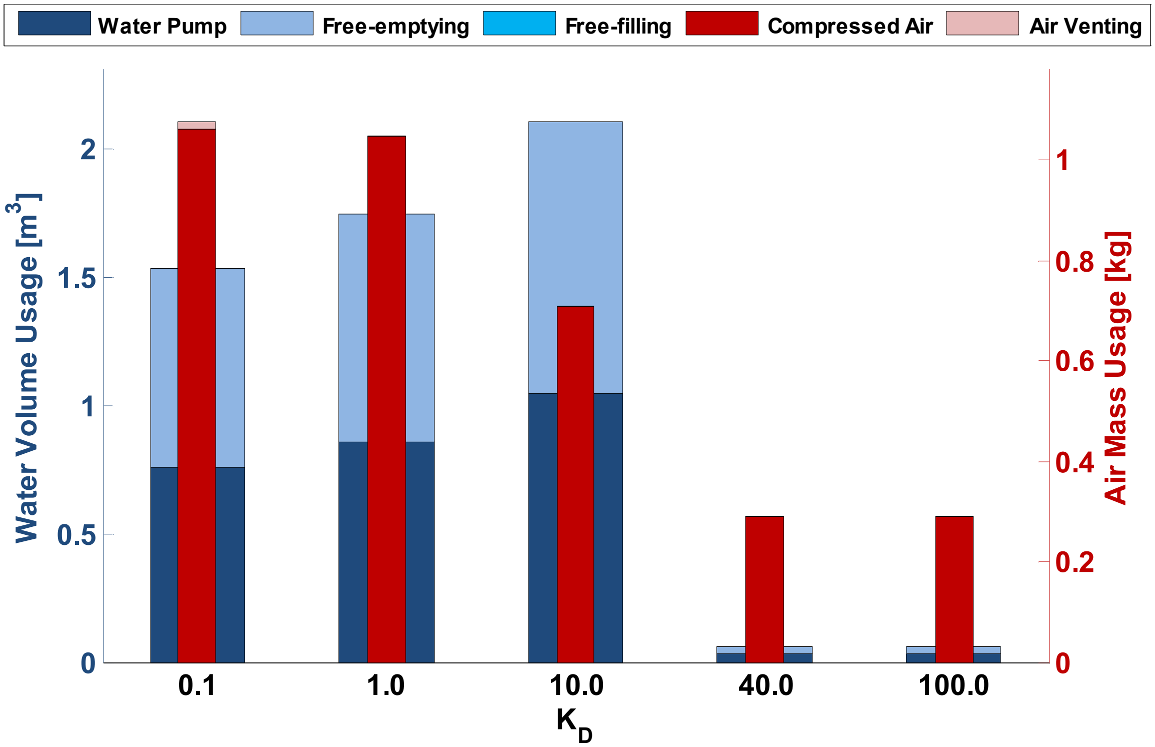

Figure 20 – PD VBS Depth Controller – Resource Usage for Various K_D Values (K_P = 0.5, t = 1000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

Figure 21 plots the AUV depth as a function of time in response to a 30 m step changes in the setpoint depth. The corresponding AUV step response from an initial depth of 100 m to a final depth of 130 m for K_P = 1.0 and K_D = 40 is represented by the dashed navy blue curve and for K_P = 1.0 and K_D = 100 is represented by the solid light blue curve. It should be pointed out from Figure 21, however, that a K_D value of 100 took longer (approximately 550 sec) to reach setpoint depth than when K_D was 40 (approximately 400 sec). From the results of Figure 21, the optimal K_P parameter was chosen to be 1.0 and K_D parameter chosen to be 40. The resulting Vanilla PD controller shows that the derivative component was able to reduce the continuous sinusoidal overshoot of the system observed in Figure 19.

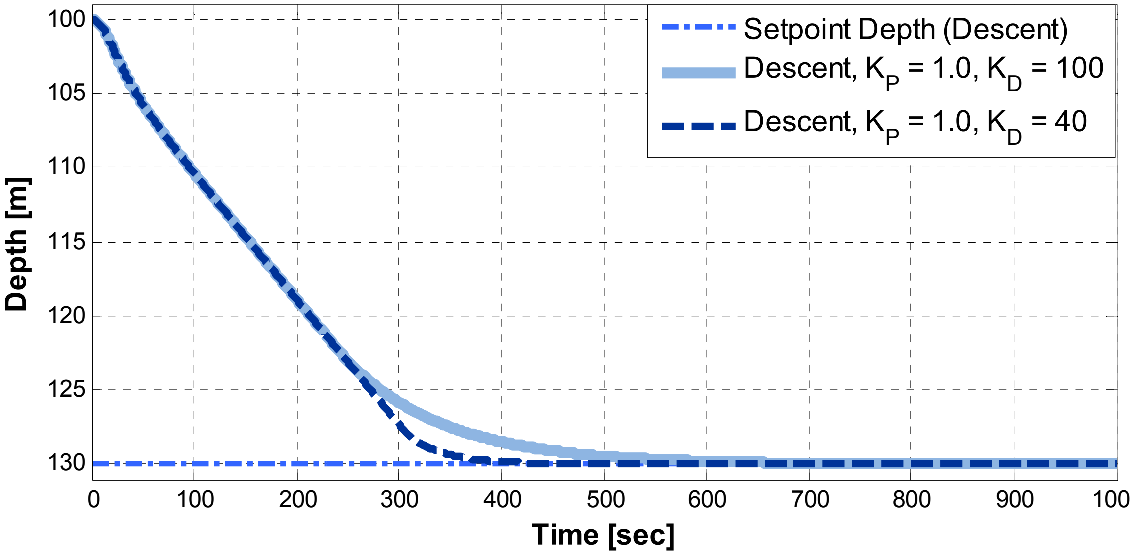

Figure 21 – PD VBS Depth Control – Optimal Vanilla PD Parameters (step change in depth from 100 m to 130 m, "small" ballast tanks)

For the rest of this thesis, the "Vanilla" PD controller utilizing these control parameters (K_P = 1 and K_D = 40) will define a near-ideal system baseline against which all other proposed controllers will be compared.

7.1.3 Resource Usage of the Vanilla PD VBS Depth Controller with Various Sized Ballast Tanks

Figure 22 compares the resource usage of the Vanilla PD VBS depth controller in the case of "small" and "large" ballast tanks when K_P and K_D were held constant with values of 1 and 40, respectively. As shown in Figure 22, the large ballast tank utilizes significantly more water and air compared to that of the small ballast tank. The resulting water pump usage is 0.03 m^3 for the small ballast tank and 1.31 m^3 (~ factor of 40) for the large ballast tank, while the free-emptying usage is 0.03 m^3 for the small ballast tank and 1.32 m^3 (over a factor of 4 increase) for the large ballast tank. The compressed air usage is 0.29 kg for the small ballast tank and 4.48 kg for the large ballast tank, and the free-filling and air venting are negligible for both ballast tanks. Notable differences can be observed between the defined "small" and "large" ballast tanks.

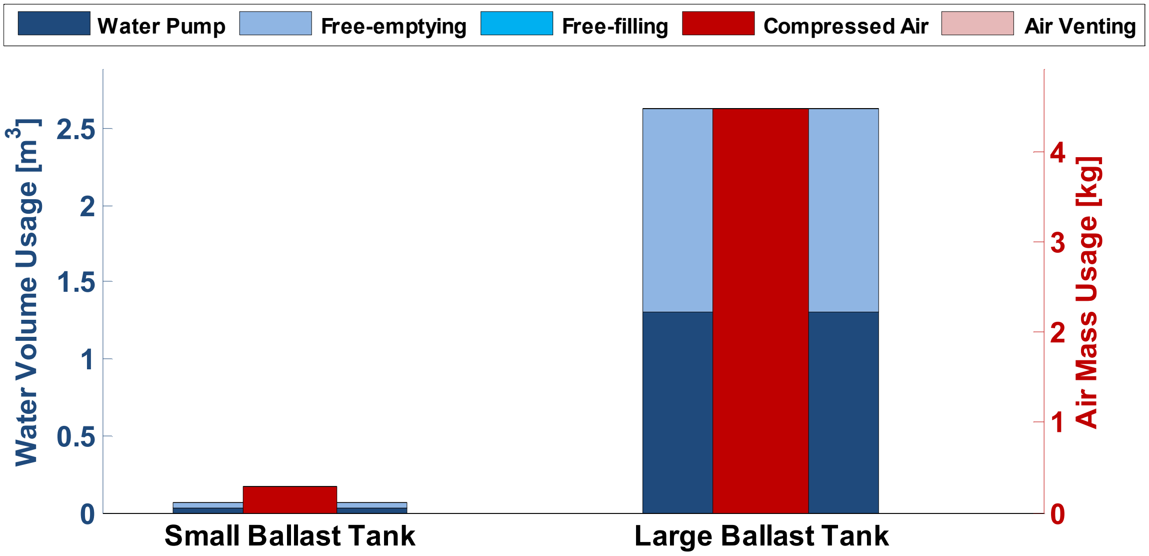

Figure 22 – Vanilla PD VBS Depth Controller – Resource Usage of "Small" and "Large" Ballast Tanks (K_P = 1, K_D = 40, t = 1000 sec, step change in depth from 100 m to 130 m)

The corresponding step responses from an initial depth of 100 m to a final depth of 130 m are shown in Figure 23 for both ballast tank sizes. It can be seen that, while the AUV with the large ballast tanks is capable of reaching the setpoint depth faster due to having a higher ballast capability, the large tank also required a longer (approximately 75 sec) time to empty, thus causing the observed increase in depth overshoot. The AUV configurations arrive at a stable setpoint depth at approximately 400 and 450 sec for the "small" and "large" ballast tanks, respectively. It should be noted that the PD controller utilizing large ballast tanks could have reduced the overshoot by further tuning of K_P and K_D, or by increasing the compressed air flow rate. By increasing this rate, the air mass would have increased faster, the ballast tank air pressure would have risen faster, and the resulting free-emptying speed would have increased overall. In an ascent, the controller performance could also be further improved by increasing the water pump flow rate. This increase in flow rate would, however, have an adverse effect on the total resource usage.

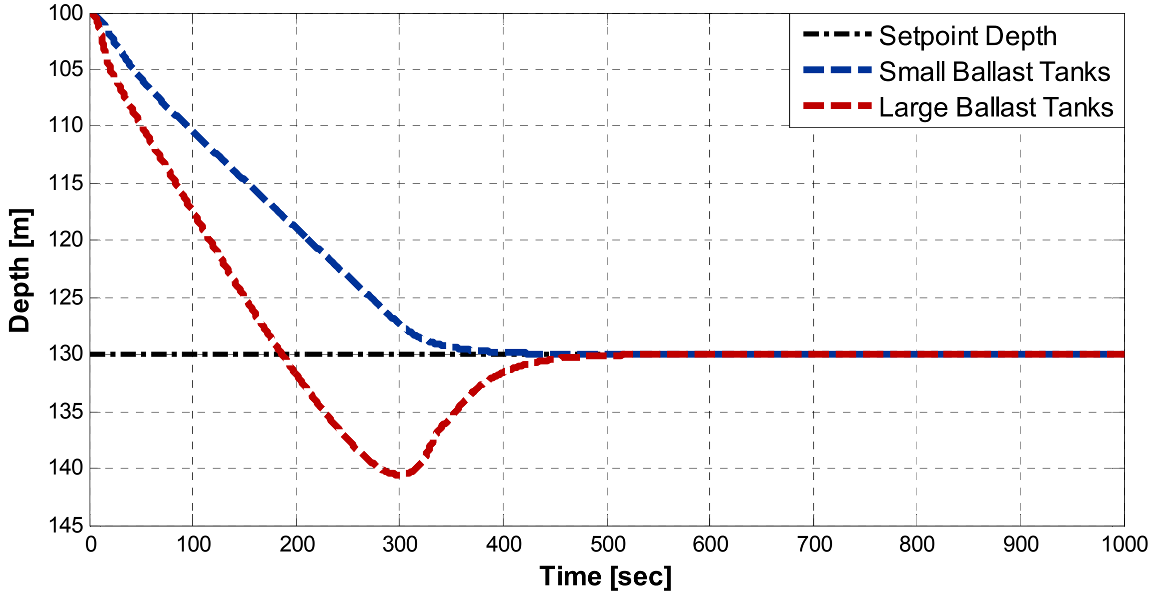

Figure 23 – Vanilla PD VBS Depth Controller Depth Control of "Small" and "Large" Ballast Tanks (K_P = 1, K_D = 40, step change in depth from 100 m to 130 m)

7.2 Additional Levels of Realism

In order to further test the Vanilla PD VBS Depth Controller proposed in Chapter 6, additional "realities" were added to the simulator. In this thesis, the following realistic effects were considered:

- Losses – Valves, Water Pump, and Compressed Air
- Sensors – Quantized Readings
- Operation Delays – Water Valve and Water Pump
- AUV Compression Due to Hydrostatic Pressure and Density Changes
- Vertical Disturbance Forces

7.2.1 Losses – Valves, Water Pump, and Compressed Air

In order to more accurately assess the water valve behaviour, valve pressure losses have been added in the simulation for both free-filling and free-emptying. These valve losses are mainly due to friction as water flows through the valve. The valve losses for various valve types will vary based on the volume flow rate of the fluid, the density of the fluid, and the geometry of the valve. For the purposes of this simulator, the effect of a drop in volume flow rate through the valve due to a pressure loss has been assumed to be a constant loss of 5% calculated using the following equation:

$$rate_{water\ valve\ actual} = rate_{water\ valve\ set} \cdot (1 - water\ valve\ loss/100) \qquad (30)$$

where $water\ valve\ loss$ represents the percentage (5%) of the volume flow rate that is lost due to friction as water flows through the water valve opening and $rate_{water\ valve\ set}$ is the setpoint volume flow rate calculated in m^3/sec.

Similarly, air valve losses have been added to the simulator for both free-filling and free-emptying. These valve losses were also assumed to be constant percentage losses as follows:

$$rate_{air\ valve\ actual} = rate_{air\ valve\ set} \cdot (1 - air\ valve\ loss/100) \qquad (31)$$

where $air\ valve\ loss$ parameter is the percentage (5%) of the mass flow rate that is lost due to friction as water flows through the air valve opening. The parameter $rate_{\text{air valve set}}$ is the setpoint mass flow rate calculated in units of kg/sec.

According to Scherer, T., water pump losses can include friction head, velocity head, and suction head losses [9]. Friction head consists of the losses in energy due to the friction within the water pump. Velocity head consists of energy losses due to the velocity of the water flow into the ballast tank through the water pump. Suction head losses consist of friction losses that occur between the water pump and the opening to the ocean, and losses that occur during the suction of water out of the ocean. Water pump losses have been included in the simulator in order to emulate the variance in the desired and actual water pump flow rate. For example, if the water pump is set to have a flow rate of 0.040 m^3/sec and the actual output is 0.035 m^3/sec, the water pump flow rate loss would be 0.005 m^3/sec. The water pump loss is modeled in the simulator as a constant flowrate loss of 5% as shown in the following equation:

$$rate_{water\ pump\ actual} = rate_{water\ pump\ set} \cdot (1 - water\ pump\ loss/100) \qquad (32)$$

where $water\ pump\ loss$ is the percentage (5%) of the volume flow rate of the water pump that is lost due to friction head, velocity head, and suction head and $rate_{water\ pump\ set}$ is the setpoint volume flow rate calculated in units of m^3/sec.

The losses for the compressed air are calculated similar to the water pump as a constant loss of 5% by the following equation:

$$rate_{comp\ air\ actual} = rate_{comp\ air\ set} \cdot (1 - compressed\ air\ loss/100) \qquad (33)$$

where $compressed\ air\ loss$ is the percentage (5%) of the air mass flow rate of the compressed air cylinder that is lost and $rate_{comp\ air\ set}$ is the setpoint air mass flow rate calculated in units of kg/sec.

Figure 24 shows the resource usage for the Vanilla PD controller with and without water valve, air valve, water pump, and compressed air losses of 5%. As can be observed in this figure there is a negligible increase in air mass usage (remained at 0.29 kg); however, there is a 20% increase in both water pump and free-emptying usage (0.030 m^3 to 0.036 m^3). There is a negligible air mass difference because the air mass in the PD controller fills the ballast tanks until their air pressure is greater than the ocean water pressure, thus allowing free-emptying to commence. Therefore, unless the ocean pressure changes (likely due to a change of depth), the air mass required to cause free-emptying will remain the same. The water usage (both water pump and free-emptying support) observed in Figure 24 can be explained by the extra time that is required for filling (via water pump) or emptying (via free-emptying) to occur. While compressed air support has the goal of increasing the air pressure within the ballast tank to allow free-emptying to commence, free-emptying itself has the goal of reducing the water from the ballast tank. Likewise, the water pump has the goal of increasing the water in the ballast tank. When there are 5% losses to the flow rates through the valve or from the water pump, it will take approximately 5% longer to fill or empty the ballast tank via these methods. Due to this extra time, there will be an increased lack of control between the position of the AUV and the setpoint depth. As the ballast tanks are filled 5% slower, the momentum of the AUV will cause it to become offset from its desired ballast control height. The controller output u_{out} of the PD controller deals with the depth error (proportional) and velocity (derivative) of the AUV. For example, if the output is 1, then water pump support will be desired (if possible). Once the ballast tank is filled (approximately 5% slower than without losses) according to the requirements of the controller, it will then be at the desired fill point. These time delays in filling or emptying the ballast tank will result in additional filling or emptying in order to compensate for accumulated depth error.

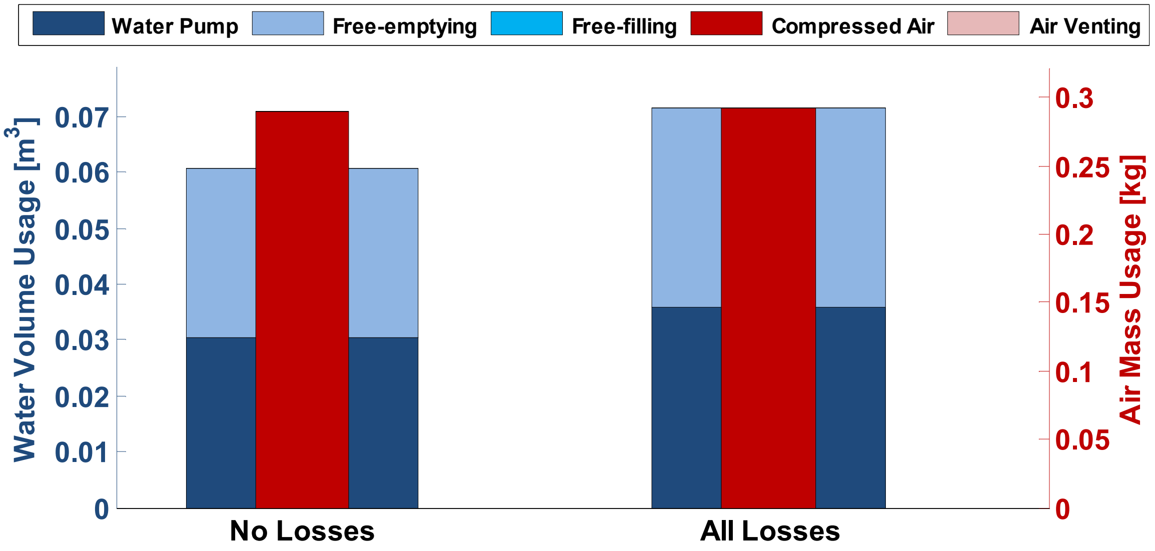

Figure 24 – Losses – Resource Usage Comparison (K_P = 1, K_D = 40, t = 1,000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

7.2.2 Sensors – Quantized Readings

When determining the depth, pitch angle, water pump volume flow rate, compressed air mass flow rate, water valve volume flow rates, air valve mass flow rates, and ballast tank water levels, MATLAB® is able to carry out simulations with very high precision (16 digits after the decimal). The level of this precision is, of course, impossible to achieve with physical sensors. The AUV sensors that have been included in the MATLAB® simulator and their corresponding quantized simulation parameters have been approximated and set as summarized in Table 8.

Table 8 – AUV Sensors and Quantized Simulation Parameter

Sensor	Type	Parameter	Precision
Compressed Air Mass Flow Rate	Mass flow meter [21]	$\dot{m}_{air}$	±0.001 kg/s
Depth	Acceleration-compensated piezoelectric sensor [18]	$depth$	±1 m

Sensor	Type	Parameter	Precision
Pitch Angle	High precision tri-axis gyroscope, accelerometer, magnetometer [19]	θ	±1°
Water Pump Volume Flow Rate	Capacitance volume flow rate sensor [20]	Q_{water}	±0.001 m^3/s
Water Valve Free-fill/Empty Rate	Capacitance flow rate sensor [20]	v_{water}	±0.0001 m/s

In the proposed VBS system, the water pump is controlled by a computer and the corresponding pump flow rates vary between a minimum value and a maximum value. With a minimum volume flow rate of 0 m^3/sec and a maximum volume flow rate of 0.01 m^3/sec the computer control quantization for water pump flow rate was assumed to be 0.001 m^3/sec for the simulations carried out in this thesis.

As with the water pump volume flow rates, the compressed air mass flow rates were also quantized. With a minimum mass flow rate of 0 kg/sec and a maximum mass flow rate of 0.01 kg/sec the corresponding quantization was assumed to be 0.001 kg/sec (1/10th the maximum flow rate) for the simulations carried out in this thesis.

The height of water in the ballast tank will use a capacitance level sensor [20]. Since the water pump and water valve rates (fill and empt) have been quantized, it is not desired to further quantize the ballast tank water height directly. The pump and valve rates directly equate to the height of water in the ballast tank and by quantizing the ballast tank height, it will end up being quantized once before integration and again after. For this main reason, the ballast tank quantization has been left out. Also, the simulator has minor issues with the quantization of heights when dealing with the compressibility of air due to a near-filled value quantizing to a filled value. This factor, along with the re-quantization, has led to the addition of ballast tank height quantization to be left out of the simulator.

To avoid numerical issues in the simulation that result when desired flow rates are close to zero, a minimum obtainable water pump and air mass flow rate was incorporated into the simulator.

Figure 25 compares the predicted resource usage for 30 m step changes in depth over a 1000 sec period when quantization is not present and when all sources of quantization (depth, pitch angle, water valve, water pump flow rate, compressed air mass flow rate, and ballast tank water level) are added from Table 8. As shown in Figure 25 when quantization is added, a significantly large increase in resource usage is predicted. Free-emptying usage increases from 0.03 m^3 to 1.79 m^3 (~5,900%), water pump usage increases from 0.03 m^3 to 1.76 m^3 (~5,800%), and compressed air mass usage increases from 0.29 kg to 1.51 kg (~420%).

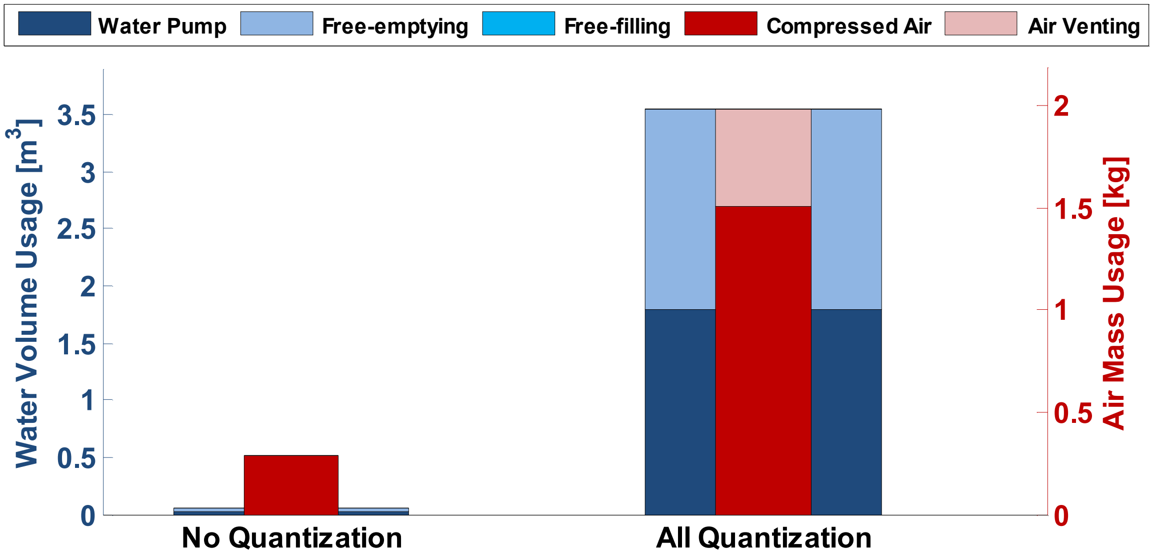

Figure 25 – Quantization – Resource Usage (K_P = 1, K_D = 40, t = 1,000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

Figure 25 shows the resource usage for the PD VBS controller. In Figure 25, each type of quantization is activated separately to determine the magnitude of effect when activated on top of the Vanilla PD controller. As shown in figure Figure 25, quantizing depth has the largest effect on resources usage on the water and air mass usage. During a descent towards the setpoint depth, as the depth error of the AUV becomes smaller in magnitude, the value of the controller output u_{out} will generally become negative due to the magnitude of the derivative component of the AUV being larger than the magnitude of the proportional depth error component. While the u_{out} remains greater than 1 or less than −1, the proportional component will be set to its maximum value. Since the proportional component K_P equals 1, the controller output will be equal to the depth error of the AUV with the addition of the derivative control portion, $K_D \cdot de_d(t)/dt$ where $de_d(t)/dt$ is the inertial Z-axis rate of change of the AUV. If the inertial Z-axis velocity of the AUV remains low, then the depth error will likely be the most influential component in the controller output. When the depth is ±0.5 away from the setpoint depth, the depth will be rounded up to the closest integer value ±1.0. Rounding up to the closest integer value will essentially double the proportional component of the controller output, $K_P \cdot e_d(t)$, where $e_d(t)$ represents the depth error of the system. Therefore, in situations where the controller output may be, for example, approximately ±0.6, the quantization of the depth of the AUV will cause it to become approximately ±1.0. Therefore, as a result of an increased controller output due to depth quantization, the AUV will require an increase in resource usage as commanded by the larger controller output u_{out}. The resource increase is approximately 5,800%, 5,700, and 420% for water pump, free-emptying, and compressed air, respectively.

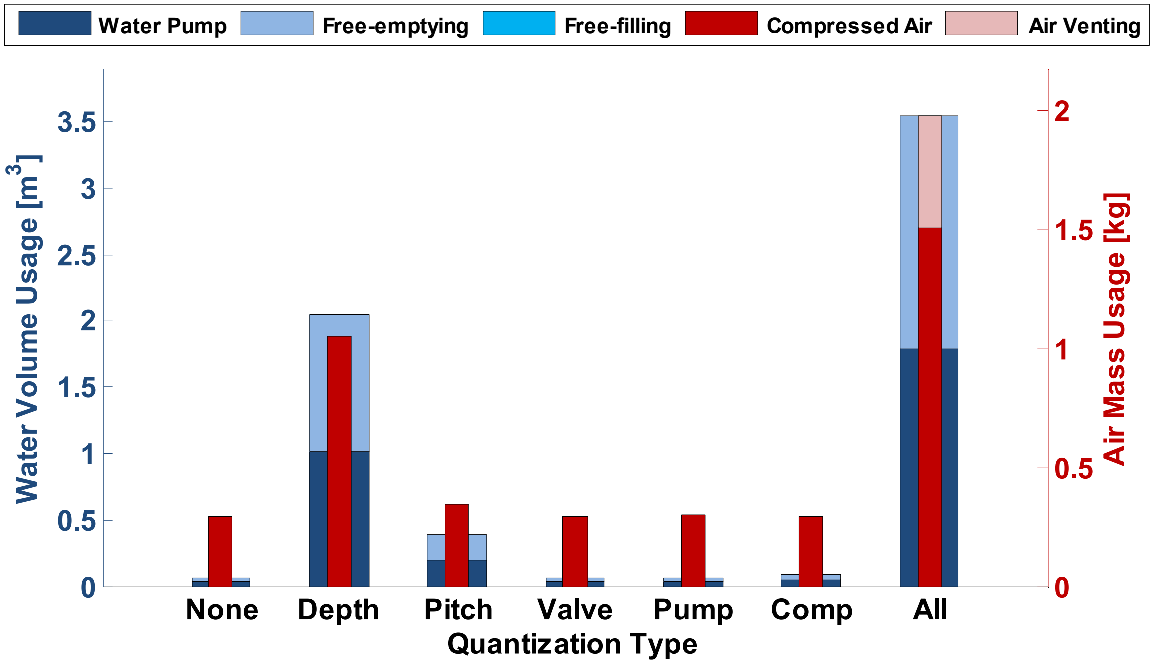

Figure 26 – Specific Quantization – Resource Usage (K_P = 1, K_D = 40, t = 1,000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

7.2.3 Operation Delays

With the vanilla PD VBS Depth controller, the water valves are assumed to open and close instantly. Because of this assumption, the AUV is able to switch between water pump support and free-filling or free-emptying support instantaneously. Both free-filling and free-emptying require usage of the water valve. In order to add realism to the water valves, they were given a finite opening and closing velocity. This valve opening and closing velocity was determined by the following equation:

$$v_{valve} = \frac{w_{valve}}{t_{open/close}} \tag{34}$$

where w_{valve} represents the width of the water valve opening and $t_{open/close}$ represents the time it takes for the water valve to fully open or close.

Valve opening and closing times between 0 and 5 seconds were simulated. The results of no delay, 1 second delay, 2 second delay, and 5 second delay can be observed in Figure 27. As observed in this figure, the addition of any valve delay appears to increase the free-emptying usage, water pump usage, and air mass usage when compared against the simulation with no delay. Compared with no delay, a 5 second delay increased free-emptying usage from approximately 0.03 m^3 to 1.28 m^3 (~4,200%), water pump usage from approximately 0.03 and 1.27 m^3 (~4,100%), and the compressed air mass usage from approximately 0.30 kg to 0.89 (~210%). The reason for the observed increases in resource usage due to the water valve delay can be understood by considering the following: When the water valve is completely opened and there is no more desired free-emptying, the water valve is told to close. Since it takes up to 5 seconds to close, there will be a significant amount of additional water that will flow out of the AUV that was not intended to flow out of the AUV. This situation will cause the AUV's ballast tank height to drop below the level that was intended by the controller output u_{out} and, as a result, the AUV will rise. Once the valve finally closes, the controller output will indicate that more water is required to cause the AUV to, once again, descend to the setpoint depth. This increase in water pump usage is a direct result of the increase in free-emptying usage. Since the air mass is increasing over the course of the simulation, there is no need to significantly increase the air mass due to this valve delay. Over time, the extra vertical movement that is being caused by the continued filling and emptying of the ballast tank will push the AUV to where it has more water than is required. As more water resources are required, the air mass must be increased to provide the necessary air pressure to allow the desired free-emptying to commence.

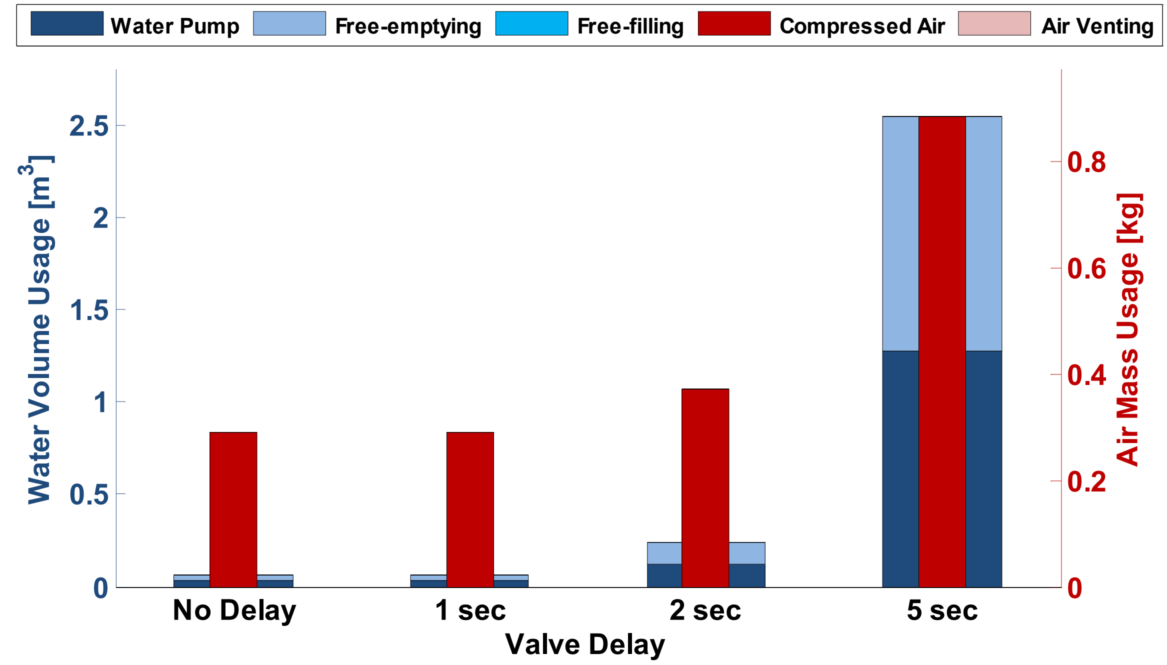

Figure 27 – Valve Delays – Resource Usage (K_P = 1, K_D = 40, t = 1,000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

There is, in reality, a finite amount of time that the water pump takes in order to start-up or shutdown its activity of moving water. The start-up time delay of water pump activity can vary from seconds to minutes, depending on the size of the pump and the length of pipe to the reservoir (i.e. the ocean). Within the simulator, a start-up and shut-down time of two seconds has been chosen. To implement this effect within the simulator, when the pump is desired to be shut off, it will take an additional two seconds in order for the water to cease being moved through the pump. Figure 28 plots the water pump volume flow rate as a function of time for simulations with various water pump operation delays. As it can be seen in Figure 28, a larger delay, represented by the dotted magenta curve, causes the water pump volume flow rate to increase and decrease at a reduced rate. If the pump is to be started or stopped within the start-up or shutdown delay (whichever is the opposite action), respectively, an additional two seconds will be added to the delay provided there is no more than two seconds for any shutdown or start-up sequence. Similar start-up and shutdown time delays were added to simulate the compressed air support.

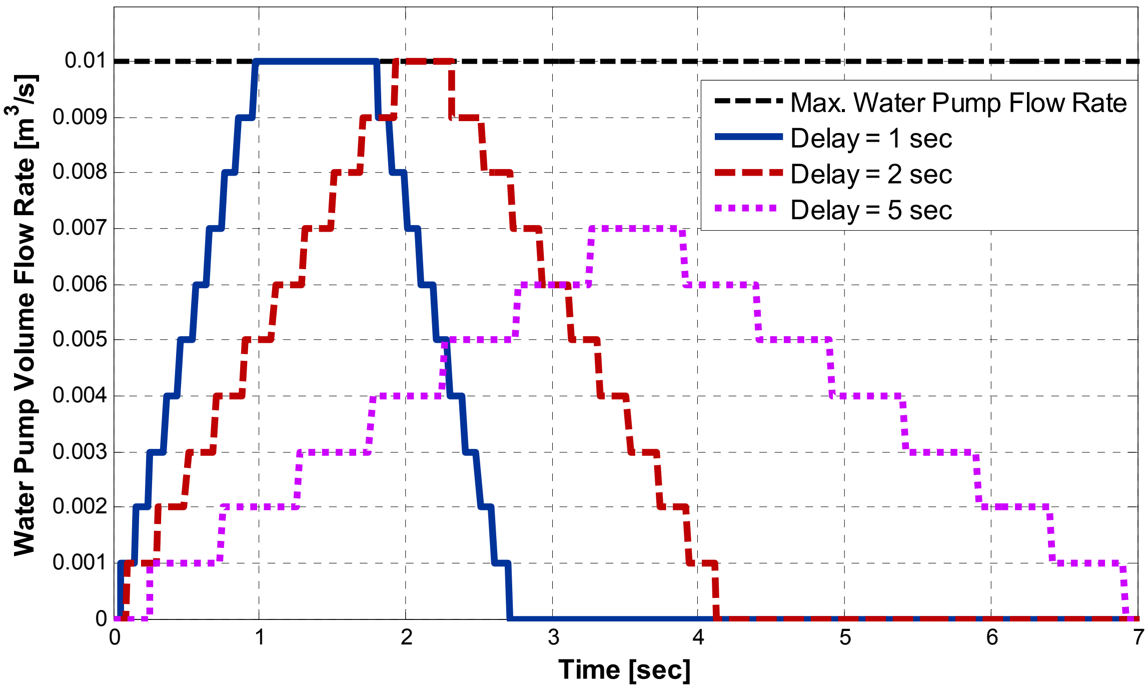

Figure 28 – Water Pump Delay (K_P = 1, K_D = 40, "small" ballast tanks)

7.2.4 AUV Compression Due to Hydrostatic Pressure, Temperature, and Salinity

As an AUV dives deeper, the ocean pressure increases linearly with depth. This hydrostatic pressure increase will create a compressive strain on the AUV, thus decreasing its overall displacement. According to Burcher and Rydill [17], since an AUV typically exhibits a fairly uniform shape (mostly circular cylinder), it tends to also compress uniformly, reducing its buoyancy. Due to this uniform compression, there is typically a negligible change in the center of buoyancy of the AUV. As the AUV goes deeper, the relationship between weight and buoyancy change and the AUV can become less buoyant or even negatively buoyant. When negatively buoyant, the overall weight in water of the AUV is higher than the magnitude of the buoyancy. At a negatively buoyant state, an unaided and non-propelled AUV will continue to sink deeper in the ocean.

According to Rudnick [16], the large, aluminum, underwater *Seaglider* was machined so that it had the same compressibility as sea water. According to his research, this feature can lead to energy savings at large depths by reducing the amount of energy required for the increased water pumping. Increased water pumping is required when the compressibility of the AUVs hull is lower than that of the surrounding ocean water. The material of the Theseus' pressure hull was made from Aluminum 7075-T6 [1]. For the simulations in this thesis, based on the research of Rudnick, it will be assumed that the compressibility of such an aluminum AUV can be approximated to that of sea water and an approximation will be used for the purposes of calculating the compressive effect that depth and temperature can have on an AUV.

Based on Archimedes law and corollary, the buoyancy of a submersed vehicle is dependent on both the volume of the vehicle and also the density of the water. According to Burcher and Rydill [17], the density of water can change as a function of temperature and salinity. In the Vanilla PD VBS depth controller, the temperature of the water was assumed to be constant and the pressure was assumed to be only a direct function of depth. According to Young and Freedman [14], when the pressure of water is increased by 1 Pa, the compressibility k_{water} of water would be $45.8\text{x}10^{-11}$ Pa^{-1}. The compressibility of water is the fractional amount of volume that the water will compress over a given pressure, given by the following equation:

$$\mathrm{k}_{water} = \mathrm{k}_{\mathrm{AUV}} = -\frac{\left(\Delta V_{AUV}/V_{AUV\,ref}\right)}{\Delta P} \tag{35}$$

where $\mathrm{k}_{\mathrm{water}}$ and $\mathrm{k}_{\mathrm{AUV}}$ represent the compressibilities of the water and AUV, respectively. $V_{AUV\,ref}$ represents the volume of the AUV at a depth of 0 m (hence the reference volume). The parameter ΔV_{AUV} represents the change in volume of the AUV that will occur at the current depth. The change of pressure ΔP is calculated by the following equation:

$$\Delta P = P_{current} - P_o = (P_{atm} + \rho_{wat} \cdot g \cdot depth) - P_{atm} = \rho_{wat} \cdot g \cdot depth \tag{36}$$

where P_{atm} represents atmospheric pressure (101,325 Pa) and $depth$ represents the current depth of the AUV relative to the free surface. This relationship would result in the following AUV volumetric decrease:

$$\Delta V_{AUV} = -\mathrm{k}_{\mathrm{AUV}} \cdot V_{AUV\,ref} \cdot \rho_{wat} \cdot g \cdot depth. \tag{37}$$

With a reference buoyancy of approximately 84,366 N (8,600 kg), the AUV will have a reference volume of 8.39 m^3 given that the buoyancy of the AUV is equated as follows:

$$B_{AUV} = \rho_{wat} \cdot g \cdot V_{AUV\,ref} \tag{38}$$

where the parameter B_{AUV} represents the buoyancy of the AUV.

As shown by Archimedes law, when the hull compresses, the AUV will be assumed to compress the same amount. Therefore, the buoyancy of the AUV will reduce by approximately the same percentage as the volume has decreased (i.e. if the AUV compresses 2%, the AUV buoyancy will decrease 2%). Figure 29 plots the AUV buoyancy decrease as a function of depth. Therefore, according to the provided data, the buoyancy of the AUV will decrease according to Figure 29 and the buoyancy change will be given by the following equation:

$$\Delta B_{AUV} = \rho_{wat} \cdot g \cdot \Delta V_{AUV} \tag{39}$$

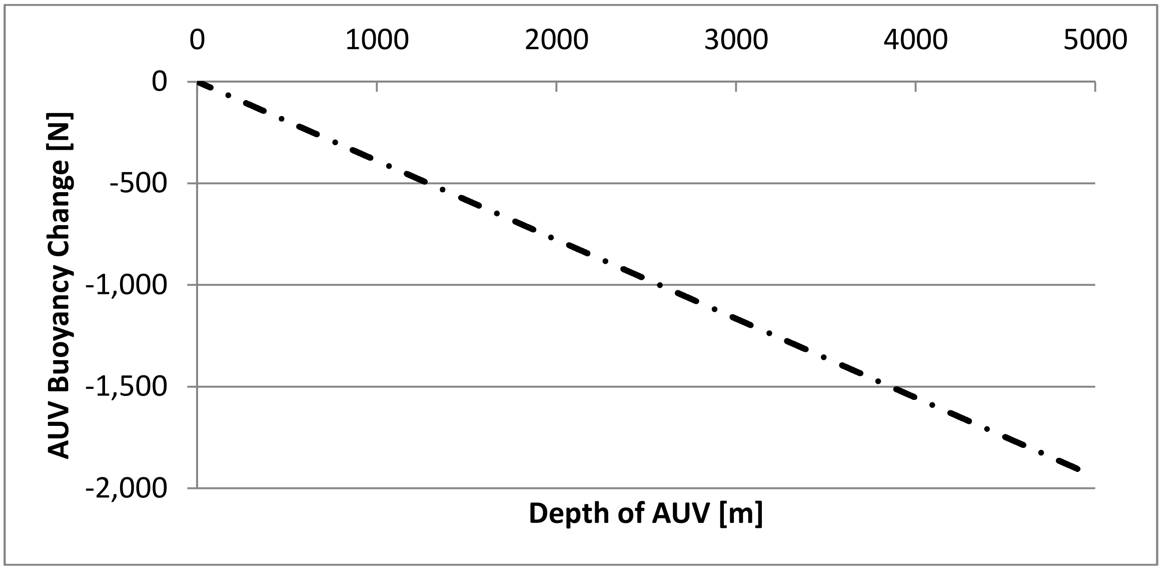

Figure 29 – AUV Buoyancy Decrease as a Function of Depth

The temperature of the ocean can range between –2°C and +35°C [10], depending on the location of the water, time of year, time of day, and water condition. Burcher and Rydill [17] also state that water density can increase from 1,000 kg/m^3 to as high as 1,030 kg/m^3 due to the effect of increased salinity levels in the water. Fluctuations in ocean water density, due to variances in temperature and salinity, will cause changes in the buoyancy of the AUV. As the buoyancy of the AUV increases, due to increased ocean water pressure, the AUV will further compress, further reducing the buoyancy of the AUV. Burcher and Rydill state that the buoyancy of an AUV can vary by as much as 3% due to variations in temperature and salinity [17].

Since the AUV has been machined to compress at the same rate at the ocean water, as the volume of the AUV decreases, the density of the ocean water will increase at the same rate and the net buoyancy of the AUV will not change (recall: $B_{AUV} = \rho_{wat} \cdot g \cdot V_{AUV}$, where V_{AUV} is the volume of the AUV). Therefore, the AUV will be capable of going to any depth regardless of AUV compression. However, it is important to note the effect of increasing AUV compression as a result of increasing depth.

According to Butler and Hertog [13], the Theseus can change buoyancy by 400 lbs (1779 N) or more due to changes in temperature and salinity. The potential change in buoyancy of 1,779 N is close to the decrease in buoyancy of 2,000 N that will occur at a depth of 5,000 m, as observed in Figure 29.

7.2.5 Vertical Disturbance Forces

There are many different disturbances in the ocean such as underwater currents. In order to approximately emulate such sea-state disturbances, constant and continuous vertical (inertial) forces can be added to the simulator. While disturbance forces also exist in the horizontal plane, the disturbance forces utilized in the simulations of this thesis have been simplified to only exists within the vertical plane.

In order to see the effect of a constant vertical disturbance force only, a disturbance force term F_{dist} (bolded) was incorporated into the equations of motion (as defined in Chapter 2), as follows:

$$\begin{aligned} mass \cdot [u_{dot} + w \cdot q - x_G \cdot q^2 + z_G \cdot q_{dot}] \\ = F_{hyd,x} - (W_Z + B_Z + \boldsymbol{F_{dist}}) \cdot sin(\theta) + F_{prop} \end{aligned} \tag{40}$$

$$\begin{aligned} mass \cdot [w_{dot} - u \cdot q - z_G \cdot q^2 - x_G \cdot q_{dot}] \\ = F_{hyd,z} + (W_Z + B_Z + \boldsymbol{F_{dist}}) \cdot cos(\theta) \end{aligned} \tag{41}$$

$$I_y \cdot q_{dot} + mass \cdot [z_G \cdot (u_{dot} + w \cdot q) - x_G \cdot (w_{dot} - u \cdot q)]$$
$$= M_{hyd,y} - (x_G \cdot W_Z + x_B \cdot B_Z + x_B \cdot \boldsymbol{F_{dist}}) \cdot \cos(\theta) \tag{42}$$
$$- (z_G \cdot W_Z + z_B \cdot B_Z + z_B \cdot \boldsymbol{F_{dist}}) \cdot \sin(\theta)$$

Both positive (down) and negative (up) vertical forces with a range of constant disturbance forces were simulated. The force was assumed to act at the center of buoyancy (at the hydrodynamic origin). With an AUV reference mass of 8,600 kg (approximately 84,366 N), small vertical disturbance forces below 100 N would not likely have a significant effect on the AUV.

The maximum theoretical disturbance force that the AUV can handle via ballast tank water level modification is equal to approximately 271 N for dual ballast tanks with a total fillable volume of 0.027 m^3. This volume is calculated using the fact that the ballast tanks are neutrally-buoyant when half-filled (the assumed reference for the simulator setup). The calculations for this maximum theoretical distribution force can be found, below:

$$F_{dist\,max} = 2 \cdot V_{bal\,tank} \cdot \rho_{wat} \cdot g. \tag{43}$$

where $V_{bal\,tank}$ represents the volume of the ballast tank. As the pitch angle of the AUV increases (nose-up) or decreases (tail up), the direction of the disturbance force will remain constant in the inertial frame of reference (either positive or negative). The effect in the body-fixed frame of reference of the AUV, however, will change depending on the pitch angle. With a smaller magnitude pitch angle, the vertical effect of the disturbance force in the body-fixed frame of reference will be at a maximum. With a pitch angle of 0°, the vertical component of F_{dist} can be found as follows:

$$F_{dist,z} = F_{dist} \cdot cos(\theta) = F_{dist} \cdot cos(0) = F_{dist}. \tag{44}$$

As the pitch angle varies from 0°, the disturbance force will have an increasing effect along the body-fixed x-axis as $F_{dist,x}$ increases. The parameter $F_{dist,x}$ is given by the following:

$$F_{dist,x} = F_{dist} \cdot sin(\theta). \tag{45}$$

$F_{dist,x}$, in turn, can slow down or speed up the AUV, depending on the orientation of the AUV and the direction of the disturbance force.

While the calculated maximum disturbance force can be, in theory, corrected by the ballast tanks, there is a limit. Depending on the air mass in the AUV and the corresponding volume of that air, the water will only be able to fill a portion of the AUV's ballast tanks. For example, with an air mass of approximately 0.0163 kg, the ballast tanks cannot fill higher than 0.298 m out of 0.30 m (99.3%). The calculations for this effect are as follows:

$$h_{air\ min} = \frac{m_{air} \cdot R_{air} \cdot T_{air}}{P_{air\ max} \cdot w_{bal}{}^2} = \frac{(0.0163)(286.9)(193)}{(50{,}000{,}000)(0.30)^2} = 0.0002\ m \qquad (46)$$

where m_{air} represents the mass of the air, R_{air} represents the gas constant of air, T_{air} represents the temperature of air, $P_{air\ max}$ represents the maximum allowable air pressure, and w_{bal} represents width of the ballast tank.

Figure 30 plots the resulting predicted AUV depth as a function of time for a 30 m step change in setpoint depth for different vertically-applied disturbance forces. As can be observed in Figure 30, at a disturbance force of 300 N, represented by the solid light blue curve, the AUV does not have the ability to counter the loss of neutral buoyancy.

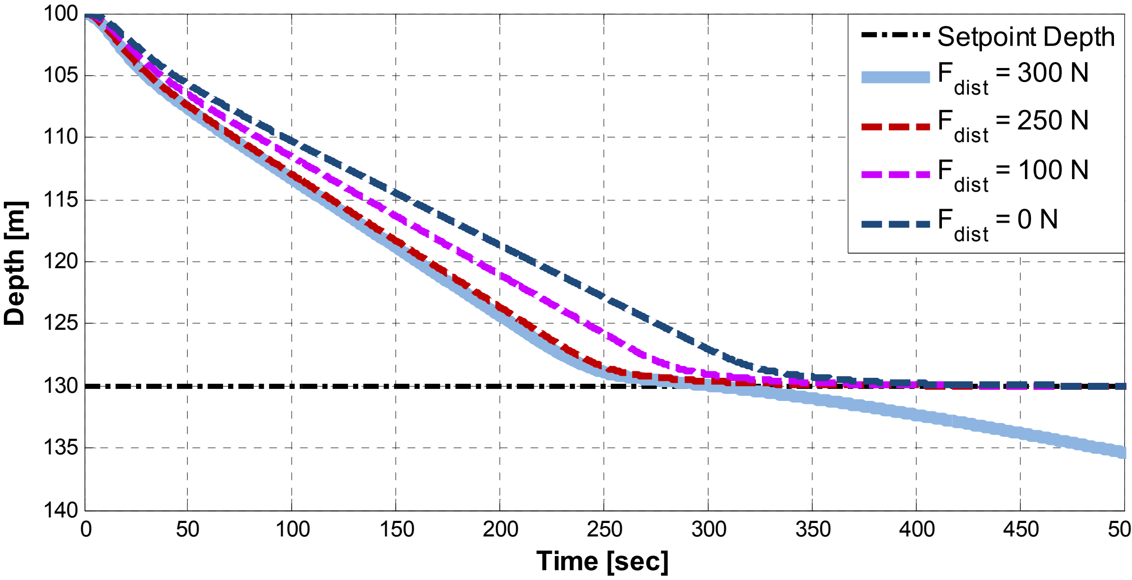

Figure 30 – Disturbance Forces Against an AUV (K_P = 1, K_D = 40, step change in depth from 100 m to 130 m, "small" ballast tanks)

Every disturbance force of 250 N (represented by the dashed red curve) or more (which are under the theoretical capable amount of 271 N) did not have an overall adverse effect on the control of depth of the AUV utilizing the Vanilla PD VBS depth controller.

7.3 Summary

The results of the simulations carried out in this chapter show how various levels of realism can affect an AUV utilizing a PD VBS depth controller. The following are a list of realistic elements which have been incorporated into the simulator. Each of the aspects of realism, whether separately or simultaneously, are activated alongside the Vanilla PD depth controller. The bold labels in the following list correspond to the *x*-axis labels in Figure 31:

▪ All Losses	*5%*	*“Losses”*
▪ All Quantizations	*various*	*“Quant”*
▪ Water Valve Delay	*2 sec*	*“Valve”*
▪ Water Pump Delay	*5 sec*	*“Pump”*

Figure 31 summarizes the corresponding resource usage results for a 30 m step change in depth (100 m to 130 m) over 1000 sec when different elements are activated separately and then simultaneously in the simulator. As can be observed in Figure 31, all aspects of realism incorporated into the simulator considerably increase both the water and air mass resource usage over that of the Vanilla PD VBS Depth controller. When activated separately, quantization provides the highest water pump usage at 1.79 m^3, free-filling usage at 1.76 m^3, and compressed air usage at 1.51 kg. The simulator with all of the listed aspects of realism activated simultaneously has water pump usage increased to 2.07 m^3, free-filling usage increased to 2.08 m^3, and compressed air usage increased to 2.85 kg.

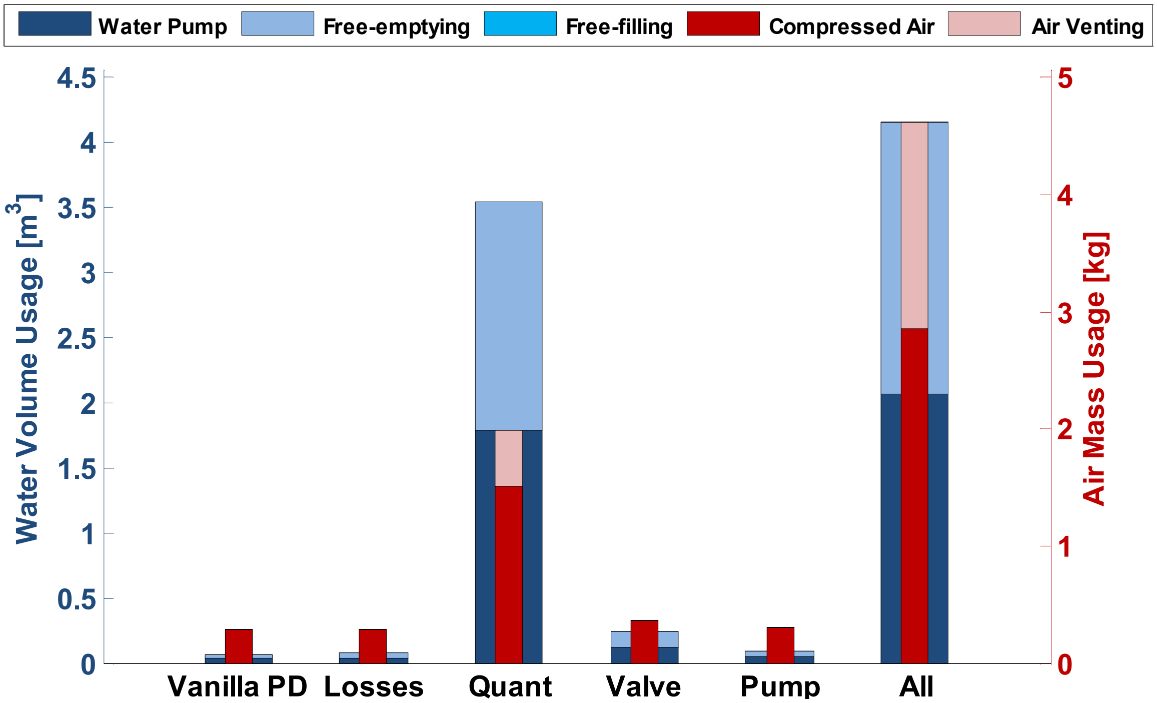

Figure 31 – Resource Usage – Multiple Situations of Realism (K_P = 1, K_D = 40, step change in depth from 100 m to 130 m, "small" ballast tanks)

Overall, the trend can be seen that adding levels of realism either separately or simultaneously to the Vanilla PD VBS depth controller appears to increase the resource usage for both water and air mass. The water pump and free-empty usage increases by approximately 6,800% and the compressed air usage increases by approximately 880%. The next chapter focuses on the design of a hybrid PD condition-based (HPDCB) VBS depth controller to try to reduce the overall resource usage without having a significant impact on the ability to effectively control the depth of the AUV.

Chapter 8 Controlling the AUV's Depth with a Hybrid PD Condition-Based VBS Depth Controller

As observed in Chapter 7, a PD VBS depth controller exhibits excellent performance when in near-ideal situations (referred to as the *Vanilla PD VBS depth controller*). However, once various realistic effects are added to the simulator, as outlined in section 7.2, the PD VBS depth controller suffers a large increase in both water and air mass resource usage. In order to improve upon this increase in resource usage in the presence of realistic effects, a hybrid PD condition-based (HPDCB) VBS depth controller was developed. There are several ways that this controller has the potential to improve upon the PD VBS depth controller, including:

- Allows free-filling to reduce water pump usage.
- Enhanced free-emptying to try to reduce compressed air usage.
- Neutral buoyancy correction to try to reduce inertial vertical AUV movement and restore or increase bowplane and sternplane control authority.
- Saturation dead zone to limit water pump, compressed air, and water valve usage, thus reducing unnecessary resource usage.
- Enhanced air venting control in order to limit air venting and conserve air mass resources.

8.1 Controller Features and Design

The following are some of the different features that are included in the HPDCB VBS depth controller and their purpose. Each feature will be turned on separately to show their utility and effect on the system.

One of the features of the HPDCB VBS depth controller is that, during free-filling and free-emptying, the water valve is opened based on the capability of the controller. In this case, the free-filling or free-emptying is only able to occur based on whether the controller output is positive (free-filling) or negative (free-emptying), making them indirect functions of the controller output. During water pump or compressed air usage, the water pump rate and compressed air rates are direct functions of the controller output. The following features and designs will be described with their impact on the controller and purpose:

- Ballast States
  - Ballast State 1 – Free-filling Support
  - Ballast State 2 – Water Pump Support
  - Ballast State 3 – Enchanced Air Venting Support
  - Ballast State 4 – Enhanced Free-emptying Support
  - Ballast State 5 – Compressed Air Support
  - Ballast State 6 – No Action
- Neutrally Buoyancy Correction
- Dead Zone Saturation
- Enhanced Air Venting Control

8.1.1 Ballast States

The goal of the HPDCB VBS depth controller is to achieve and maintain setpoint depth while using a minimum amount of water and air mass resources. In order to achieve this goal, the HPDCB VBS depth controller will determine the appropriate state at which the ballast tank should be at, known as *ballast states*. There are six ballast states: free-filling (ballast state 1), water pump support (ballast state 2), enhanced air venting support (ballast state 3), enhanced free-emptying (ballast state 4), compressed air support (ballast state 5), and no action (ballast state 6). The ballast states are summarized in the flow chart of Figure 32 and are used to determine the action that the ballast tank will perform.

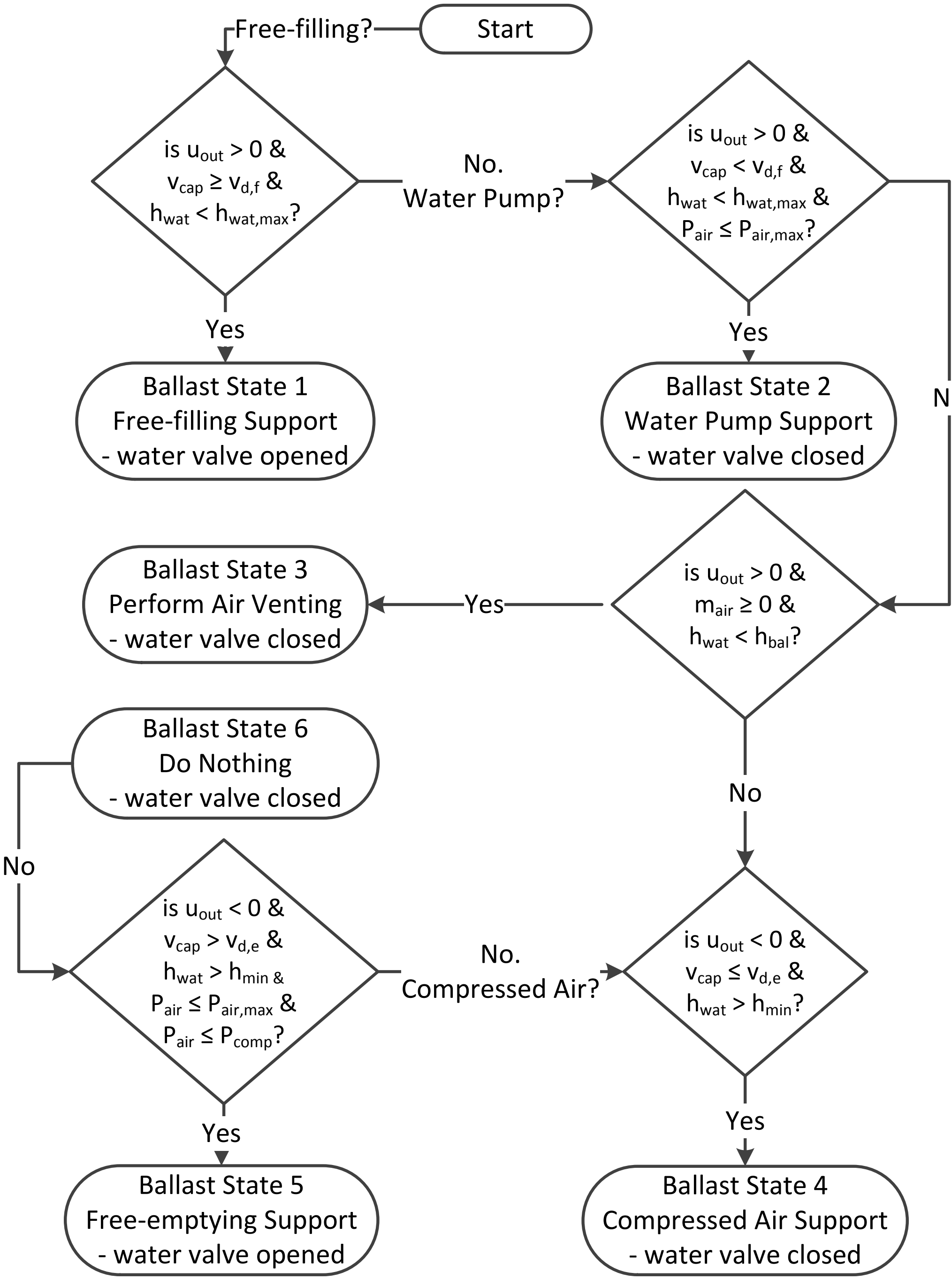

Figure 32 – Ballast States – Flow Chart Detailing Action to be Performed by Controller

Each of the six ballast states will be discussed briefly in the following sections.

8.1.2 Ballast State 1 – Free-Filling Support

The HPDCB VBS depth controller utilizes the same controller output (u_{out}) saturation as with the PD VBS depth controller developed in Chapter 7. One of the features that the condition-based controller supports is the ability to allow for free-filling into the ballast tank, as described in Chapter 3. For the HPDCB VBS depth controller, free-filling will occur when the ballast tank pressure difference (P_{diff}) can provide a minimum positive ballast tank water level velocity. When the contents of the ballast tank can achieve this velocity, specific to the HPDCB VBS depth controller, the ballast tank is said to be capable of free-filling. The controller output u_{out} is defined as can be found in Equation *(29)* on page 43, as found in Chapter 7. To determine the capable controller output values, the currently-attainable free-fill velocities v_{cap} are required as given by the following equation:

$$v_{cap} = \frac{P_{diff}}{pd_{con} \cdot w_{bal}^2} \tag{47}$$

where P_{diff} represents the pressure difference between water and air in ballast tank, pd_{con} represents the pressure difference constant, and w_{bal} represents the width of the ballast tank. The water valve area, during free-filling, is defined as:

$$A_{wat\ val} = A_{wat\ val\ max} \cdot \left|\frac{v_{d,f}}{v_{cap}}\right| \tag{48}$$

where $A_{wat\ val\ max}$ represents the maximum water valve area, $v_{d,f}$ represents the desired ballast fill velocity, and v_{cap} represents the capable fill or empty velocity. Free-filling support is activated only when the following two conditions are met:

- Condition 1: $u_{out} > 0$ AND
- Condition 2: $v_{cap} \geq v_{d,f}$.

8.1.3 Ballast State 2 – Water Pump Support

For the HPDCB VBS depth controller, the volumetric flow rate of the water pump will be given by the current rate of the water pump, defined as:

$$Q_{wat\ pump} = Q_{wat\ pump\ max} \cdot u_{out} \tag{49}$$

where $Q_{wat\ pump\ max}$ represents the maximum water volume flow rate of the water pump. Water pump support is activated only when the following three conditions are met:

- Condition 1: $u_{out} > 0$ AND
- Condition 2: $v_{cap} < v_{d,f}$ AND
- Condition 3: $P_{air} \leq P_{air\ max}$

where P_{air} represents the air pressure and $P_{air\ max}$ represents the maximum allowable air pressure in the ballast tank.

8.1.4 Ballast State 3 – Enhanced Air Venting Support

Air venting occurs whenever it is desired to increase the water height but it is not possible due to the current water height being higher than the maximum allowable water height. In this case, it is due to the pressure of the air being higher than the set maximum value of 100 MPa. However, there are many times when this situation holds true, yet it is not necessary to vent the air because the water height is already high enough to complete the desired task of descending the AUV. Therefore, an enhanced air venting control was set for the HPDCB VBS depth controller that will only allow it to vent when the maximum water height reduces below 0.17 m (this value can be modified for flexibility). The height of 0.17 m would only allow a height difference of 0.02 m assuming a neutrally-buoyant state at 0.15 m (half-filled for the small ballast tank). This 0.02 m will allow the AUV to descend. However, the rate at which the AUV will descend will be slower than that of an AUV with ballast tanks that are 90% filled or more. Therefore, in order to reduce the occurance of air venting, the potential descent velocity at times when the maximum water height is near 0.15 m (yet still equal to or above 0.17 m), will be reduced. In order to achieve a maximum water height of 0.17 m, the ballast tanks must accumulate approximately 13.9 kg of air mass. This amount of air mass can be accumulated over time or due to a large depth. The ballast tank height of 0.17 m can be increased or decreased, allowing for either more or less of a reliance on air venting. Given the limited nature of air mass available in the compressed air cylinder, this value is suggested for the small ballast tank in order to conserve air mass resources without having a large impact on performance. Benefits of this enhanced air venting control can be seen in comparisons between the PD and HPDCB VBS depth controllers, later in this chapter. The PD VBS depth controller does not utilize the enhanced air venting control.

8.1.5 Ballast State 4 – Enhanced Free-Emptying Support

The HPDCB VBS depth controller has an enhanced ability for free-emptying over that of the PD VBS depth controller from Chapter 7. The water valve area, during free-emptying, is defined as:

$$A_{wat\ fwd\ val} = A_{wat\ val\ max} \cdot \left|\frac{v_{d,e}}{v_{cap}}\right| \tag{50}$$

where $v_{d,e}$ represents the desired ballast empty velocity. Free-filling support is activated only when the following two conditions are met:

- Condition 1: $u_{out} < 0$ AND
- Condition 2: $v_{cap} \leq v_{d,e}$.

8.1.6 Ballast State 5 – Compressed Air Support

For the HPDCB VBS depth controller, the mass flow rate of the compressed air cylinder is given by the following equation:

$$\dot{m}_{air} = \dot{m}_{air\ max} \cdot \mathrm{u}_{out} \tag{51}$$

where $\dot{m}_{air\ max}$ represents the maximum mass flow rate of the compressed air cylinder Compressed air support is activated only when the following three conditions are met:

- Condition 1: $u_{out} < 0$ AND
- Condition 2: $v_{cap} > v_{d,e}$ AND
- Condition 3: $(P_{air} - P_{air\ max}) < 0$.

8.1.7 Ballast State 6 – No Action

If all of the other five ballast states are not activated, then the ballast tank will be commanded to close its water valve, turn off water pump, and turn off compressed air support, if necessary. Even if the ballast is defined to perform *no action*, there is still the potential for the ballast tanks to be activated for neutral buoyancy correction which will be discussed in the next subsection.

8.1.8 Neutral Buoyancy Correction

In order to try to reduce the resource usage to a minimum, a neutral buoyancy correction (NBC) has been incorporated into the controller's logic. It is desired to cause the AUV to become neutrally buoyant so that the desired depth and velocity can be maintained without requiring further ballasting control or the use of the bowplane and sternplane controller (which would require an adequate forward AUV velocity in order to be effective). The controller monitors the height of the water in the ballast tanks in order to determine whether the AUV is neutrally buoyant. The state of neutral buoyancy depends upon knowledge of the current height of the water in the ballast tanks that is required to create a neutrally buoyant state. With a larger magnitude of weight than buoyancy, the AUV will sink. On the other hand, having a larger magnitude of buoyancy than weight will cause the AUV to ascend. If this imbalance is left unchecked, the bow- and sternplanes will experience a loss of control authority as they try to continuously adjust the depth of the vehicle from changing due to the weight and buoyancy imbalance. By filling the ballast tanks appropriately, causing the AUV to become neutrally-buoyant, it may be possible to restore this control authority. In real-world situations, neutral buoyancy can be difficult to achieve as there are constant changes in the local water density, salinity, temperature, and the mass of the AUV.

The neutral buoyancy correction is performed only if *all* of the four following conditions are met:

- Condition 1: $\left|h_{wat} - h_{neutral\ buoyant}\right| \geq h_{allowed\ variance}$ AND
- Condition 2: $v_{Gz} \leq +0.05\ m/s$ AND
- Condition 3: $v_{Gz} \geq -0.05\ m/s$ AND
- Condition 4: $|depth\ error| \leq +2.0$ m

where h_{wat} represents the height of water in the ballast tank, $h_{neutral\ buoyant}$ (i.e. 0.15 m) represents the height of the water at which the ballast tank is considered neutrally-buoyant, $h_{allowed\ variance}$ represents the height variance that is allowed to exist between the actual water tank height and the neutral buoyancy level (i.e. 0.001 m), and v_{Gz} represents the velocity at the center of mass of the AUV in the inertial Z direction. For this logic check, symmetry is applied such that if one ballast tank is neutrally buoyant, then the other ballast tank is also neutrally buoyant. Any magnitude of difference between these water heights that is larger than this allowed variance will result in the AUV not being considered neutrally buoyant. Once the state of neutral buoyancy is determined, the ballast tanks will be filled or emptied in ordered to allow them to be within this neutrally-buoyant state.

During neutral buoyancy correction, filling the ballast tank will consist of water pump support. Free-filling is not supported during neutral buoyancy. Compressed air support along with the consequential free-emptying will be supported. Compressed air support and free-emptying are supported during neutral buoyuancy correction. Free-filling will not enhanced as discussed in 8.1.5. That is to say that free-filling will commence immediately once the controller output u_{out} is positive and the pressure difference P_{diff} between the ocean water and ballast tank air pressure is negative as a result of compressed air support. During neutral buoyancy, the *maximum* water pump volume flow rate and compressed air mass flow rates will be set to be equivalent to their *minimum* rates. Reducing the flow rates will allow the ballast tank to fill up at the lowest rate possible, which will be helpful in attempting to void all four conditions of the neutral buoyancy correction, simultaneously. Reducing the flow rates is important in lowering the magnitude of the height difference $(h_{wat} - h_{neutral\ buoyant})$ in the ballast tank so that Condition 1 no longer complies. Once the ballast tank is neutrally buoyant, the AUV will begin slowing down, voiding Conditions 2 and 3. Once Conditions 1, 2, and 3 are void and the AUV is within ±2 m of the setpoint depth, the AUV will be considered neutrally buoyant.

8.1.9 Depth Error Dead Zone Saturation

While not likely as effective as the neutral buoyancy correction, the HPDCB VBS depth controller will also employ dead zone saturation to try to reduce resource usage even further. The dead zone saturation checks to determine if the following two conditions have been met:

- Condition 1: $|depth\ error| \leq boundary\ error$ AND

- Condition 2: $|h_{wat} - h_{neutral\ buoyant}| < h_{allowed\ variance}$

where $boundary\ error$ represents the region (in meters) above or below the setpoint depth range that the AUV is permitted. If these two conditions are met, then all of the AUV ballast actions are set to zero. These ballast actions include closing the water and air valves (if opened), turning off the water pump (if on), and turning off the compressed air support (if on). This dead zone saturation works alongside the neutral buoyancy correction.

8.2 Comparison with the PD VBS Depth Controller

In order to determine whether or not the HPDCB VBS depth controller is more efficient than the PD VBS depth controller, simulations have been performed using the same PD parameters as with the previous results tested in Chapter 7 (K_P = 1, K_D = 40).

Figure 33 plots the AUV depth as a function of time for a 30 m step change in depth for the Vanilla PD VBS depth controller, PD VBS depth controller with *all realism* options enabled, and HPDCB VBS depth controller with *all realism* options enabled. For the purposes of these simulations, the term "*all realism"* includes full quantization, all losses at 5%, a water valve delay of 2 sec, and a water pump delay of 5 seconds. As can be observed in Figure 33, the HPDCB VBS depth controller, represented by the solid navy blue curve, is able to follow the setpoint depth error to within 2 m at all times once it is initially reached. The error of 2 m is within the depth error dead zone of ±2 m; therefore, it can be concluded that the HPDCB VBS depth controller has an effectiveness that is approximately equivalent to that of both the PD VBS depth controller and the Vanilla PD VBS depth controller.

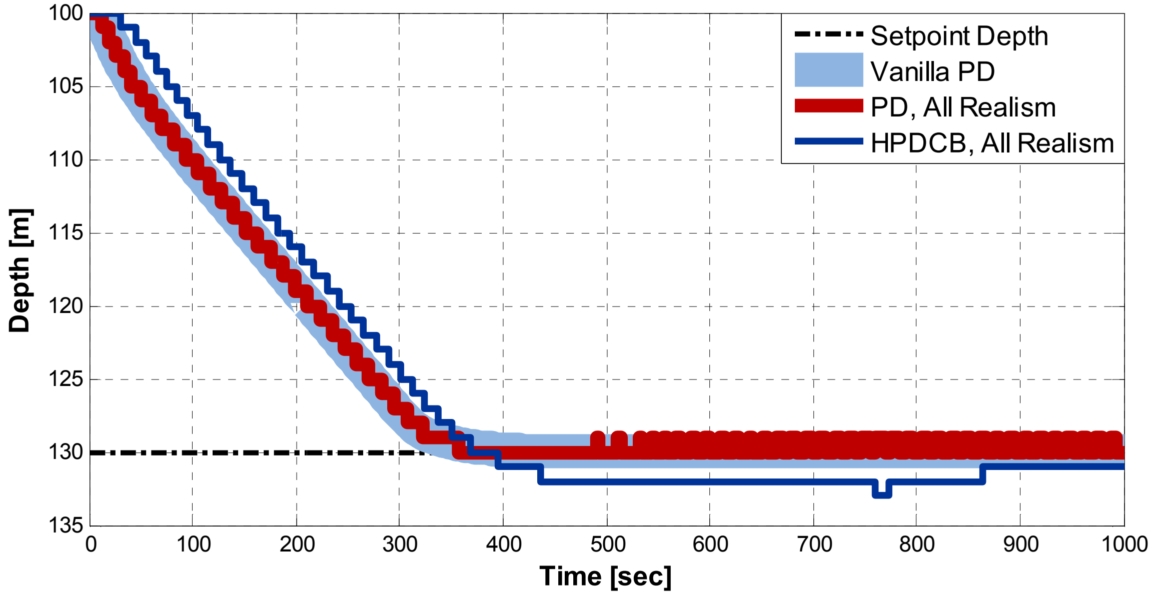

Figure 33 – PD vs. HPDCB VBS Depth Controller – Setpoint Depth Control ($K_{P,VBS}$ = 1, $K_{D,VBS}$ = 40, step change in depth from 100 m to 130 m, "small" ballast tanks)

Figure 34 plots the AUV depth as a function of time for a 30 m step change in setpoint depth. Figure 34 displays the ability for the HPDCB VBS depth controller to track a setpoint depth as it moves from one end of the depth error dead zone (plotted in Figure 34 for reference) to another over a 2000 second simulation.

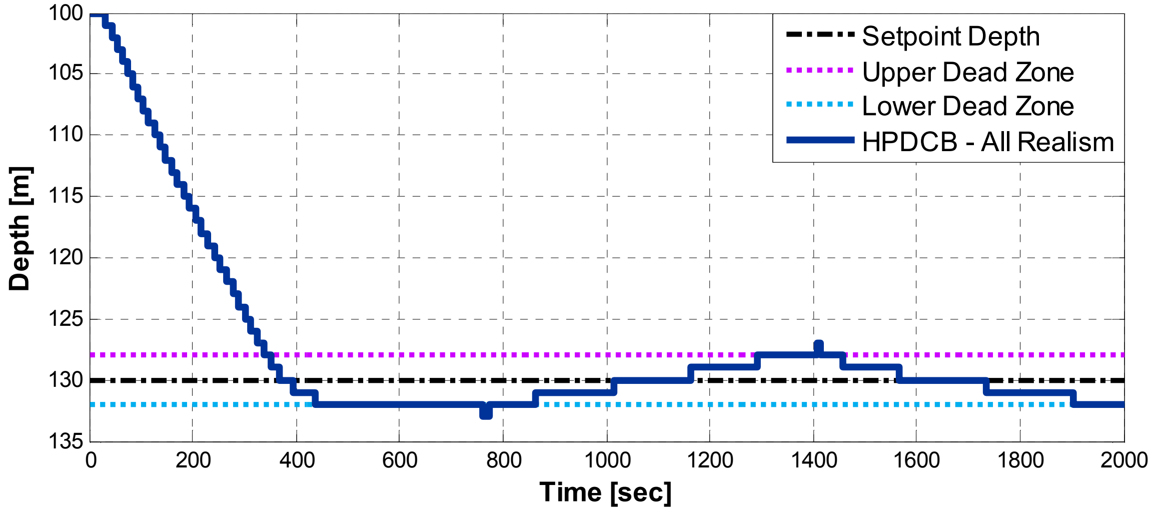

Figure 34 – HPDCB VBS Depth Controller – Sustained Setpoint Depth Control (K_P = 1, K_D = 40, step change in depth from 100 m to 130 m, "small" ballast tanks)

Figure 35 shows the resource usage for both the PD and HPDCB VBS depth controllers for when there is no realism and when there is realism. As stated in section

7.1, the Vanilla PD VBS depth controller does not include any levels of realism. In Figure 35, “PD All” refers to the PD VBS depth controller with levels of realism including losses, quantization, minimum flow rates, and operational delays (*all realism)*. In Figure 35, the HPDCB VBS depth controller is as “HPDCB Baseline” and “HPDCB All”. “HPDCB Baseline” refers to the HPDCB VBS depth controller without any levels of realism and “HPDCB All” refers to the HPDCB VBS depth controller with all levels of realism including losses, quantization, minimum flow rates, and operational delays. It can be observed in Figure 35 that the resource usage of the HPDCB VBS depth controller is very close to that of the Vanilla PD VBS depth controller. When *all realism* effects are enabled, the PD VBS depth controller has a significant increase in resource usage as was concluded in Chapter 7. When *all realism* effects are enabled on the HPDCB VBS depth controller, there is not a large impact on the resource usage of the controller. For the HPDCB VBS depth controller, adding the realistic options did not cause any significant change in the free-filling water usage, as it remained at approximately 0.02 m^3. The increase in free-emptying water was negligible. The water pump usage remained negligible. The compressed air mass usage increased from 0.34 kg to 0.43 kg (~29%). Only the PD VBS depth controller, with *all realism*, showed air mass usage as a result of venting. Compared with the PD VBS depth controller with *all realism* options enabled, there were the following resource usage changes: free-filling increased from 0 m^3 to 0.02 m^3, free-emptying reduced from 1.57 m^3 to 0.04 m^3 (~–98%), water pump reduced from 1.58 m^3 to a negligible amount(~–100%), and compressed air mass reduced from 1.80 kg to 0.43 kg (~–76%). The most significant results here are the significant decrease in both water pump usage and compressed air mass usage. Free-filling and free-emptying usage relies only on the physics of water pressure differences and valves. However, water pump usage relies on a machine that uses considerable energy, and compressed air support relies on a compressed cylinder that contains a finite internal amount of air. Therefore, the limit of water pump and compressed air usage is the most important aspect in the conservation of resources for longer or more complex missions.

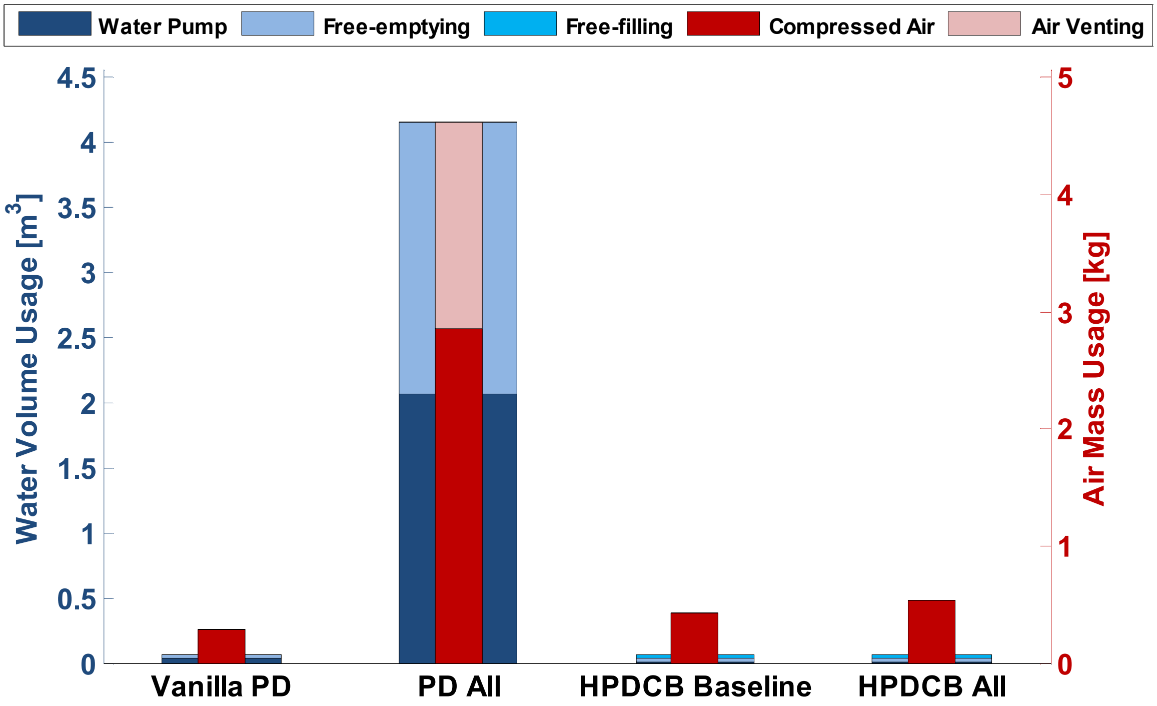

Figure 35 – Vanilla PD vs. HPDCB VBS Depth Controller – Resource Usage ($K_{P,VBS}$ = 1, $K_{D,VBS}$ = 40, t = 1,000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

8.3 Comparison with the Bowplane and Sternplane Proportional Controller

Figure 36 compares the bowplane and sternplane proportional depth and pitch controller, PD VBS depth controller and newly-proposed HPDCB VBS depth controller by plotting the AUV depth and pitch for a 30 m depth change as a function of time. The results of the bowplane and sternplane proportional depth and pitch controller (see Chapter 6) are based on an AUV average mean velocity of 1.00 m/s.

As shown in Figure 36, the bowplane and sternplane proportional controller with a mean forward AUV velocity of 1.00 m/s achieves a response approximately equivalent to that of the dual 0.027 m^3 ballast tanks using both PD and HPDCB VBS depth control. With a velocity above 1.00 m/s, the bowplane and sternplane controller performs faster than the ballast tank control approaches. In situations where the AUV is unable to maintain a forward velocity of 1.00 m/s, the AUV can attain higher response times with the ballast tank controllers. In these situations, the ballast tank method of control is more effective and efficient for low-velocity and large depth changes (1000 m+).

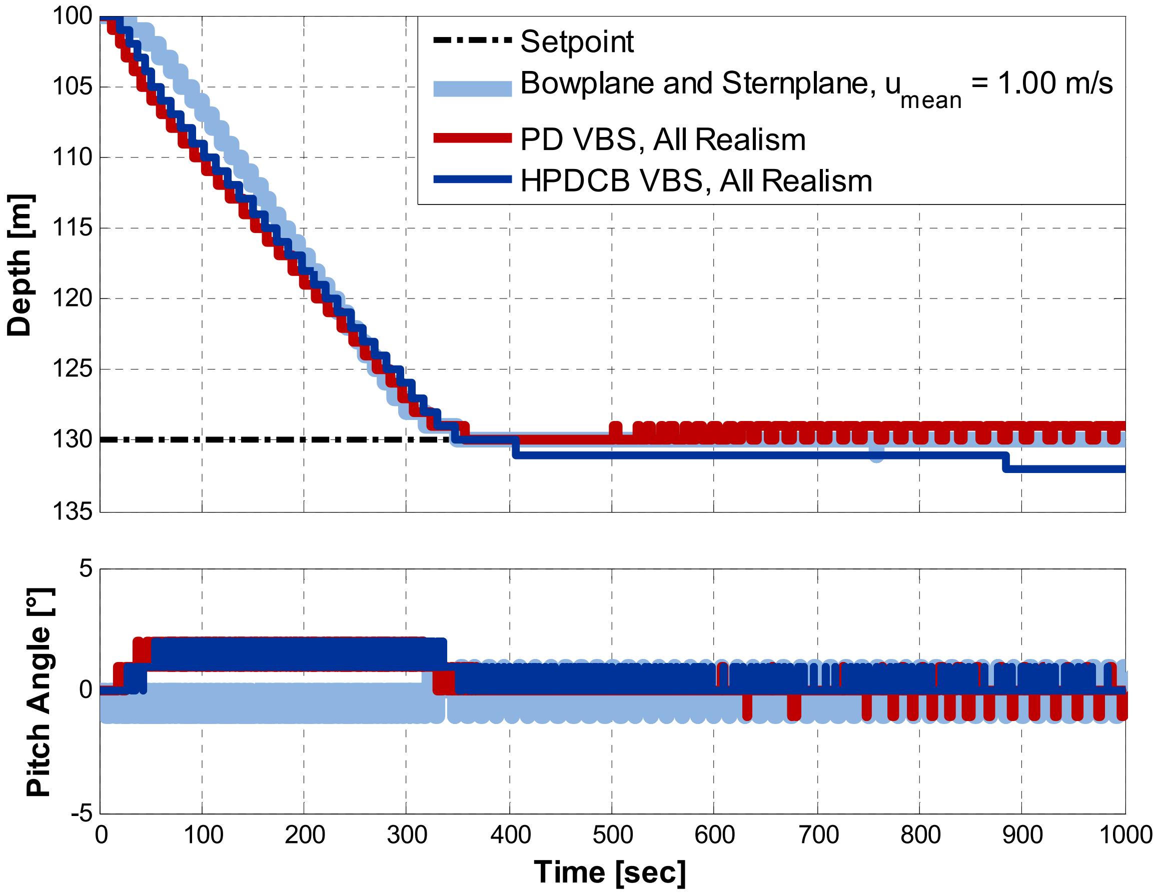

Figure 36 – Controller Comparison – Setpoint Depth Control and Pitch Angle ($K_{P,VBS}$ = 1, $K_{D,VBS}$ = 40, $K_{P,bow}$ = 0.1, $K_{P,ste}$ = 8, step change in depth from 100 m to 130 m, "small" ballast tanks)

8.4 Summary

Given the results presented in this chapter, the HPDCB VBS depth controller demonstrates reduced free-emptying (~—98%), water pump (~—100%), and compressed air (~—76%) usage over that of the PD controller when simulated with several realistic options enabled. The HPDCB VBS depth controller is capable of free-filling which allows for a reduction of the water pump support as it provides another means for water entering the ballast tanks. The neutral buoyancy correction, under specific conditions, can significantly reduce resource usage by filling or emptying the ballast tanks to maintain a neutrally-buoyant state. The HPDCB VBS depth controller also utilizes a depth error dead zone that can request that water valves close and that water pump and compressed air mass support cease. The HPDCB VBS depth controller with a "small" ballast tank allows the AUV to reach a setpoint depth with a performance that is similar to that of the

AUV utilizing bowplane and sternplane control at a mean forward velocity of approximately 1.00 m/s. Overall, the HPDCB VBS depth controller provides an alternative depth control solution utilizing variable ballast tanks. It provides variables ballast control over a large depth range (dependent on ballast tank size) while attempting to keep resource usage to a minimum. The controller can easily be modified and adapted to work alongside a bow- and sternplane controller to give the best range of control capabilities for an AUV that utilizes variable ballast tanks. Suggested controller operating conditions can be found in section 10.2.

Chapter 9 Controlling the AUV's Center of Gravity Along the Body-Fixed x-axis

In order to try to provide the bowplane and sternplane with more control authority, by reducing their steady-state fin angles, a novel variable ballast tank x_G shifting controller is proposed. Fundamentally, in order to compensate for shifts in x_G in an attempt to return full control authority to the bow- and sternplanes, the ballast tank contents must be modified. By modifying the water levels in the two ballast tanks, it should be possible to shift x_G forward or aft as necessary.

9.1 Simulation Conditions

For the following simulations that involve shifting the AUV center of gravity along the body fixed x-axis, the "realism" simulation features described in chapters 7 and 8 were disabled. These realistic effects were not included in the following simulations because the x_G shifting controller is not being directly compared to any other controllers discussed in this thesis. Rather, the purpose of these simulations is to illustrate the concept, viability, and potential benefits of shifting the center of gravity via changing the water contents of one or both ballast tanks. For the simulations discussed in this chapter (unless otherwise noted), the center of gravity component z_G along the body-fixed z-axis, will be located at 0.075 m (below the origin). By moving this center of gravity lower, it will increase the stability of the AUV and thus decrease the requirement for shifting x_G in order to recover stability. There will be certain situations where the center of gravity cannot be lowered further, such as after reducing the mass of the AUV considerably (e.g. after laying all the cable in the payload bay). It is in these situations where the shifting of x_G is more applicable.

9.2 Consequences of a Shifted Center of Gravity on the HPDCB VBS Depth Controller

Based on the physical parameters of the AUV system, a desired x_G shift of ±0.0129 m, for example, from ballast tank heights of 0.15 m (forward) and 0.15 m (aft), will require that the forward tank be filled to 0.30 m and the aft tank be emptied to 0.00 m – the theoretical maximum shift in x_G (±0.0258 m) for the scenario when both ballast tanks are initially half-filled. However, since any present air mass in the ballast tank will prevent water from completely filling the 0.30 m height of the tank, a center of gravity shift of ±0.0129 m cannot be achieved without removing all of the air content from the aft ballast tank. Therefore, a shift that is slightly smaller than ±0.0129 m has been studied in this paper (±0.0128 m) and can be achieved from a reference of 0.15 m height for each of the ballast tanks. The largest possible shift will be twice the amount (0.0258 m) and will occur when one tank is completely filled (i.e. forward) and the other (i.e. aft) is completely emptied.

Figure 39 plots the AUV depth and pitch angle as a function of time in response to a 30 m step change in setpoint depth. Figure 39 shows the effect of shifting the center of gravity x_G on the resulting control performance of the HPDCB VBS depth controller on an AUV with "small" ballast tanks. Figure 39 presents a positive and negative shift in the center of gravity represented by the solid red and magenta curves, respectively. It can be seen in Figure 39 that both a positive and negative shift in x_G has a negligible effect on the setpoint depth control capability as the AUV is able to control the depth within the 2 m dead zone. There is a noticeable effect on the pitch angle of the AUV: the positive shift in x_G causes the pitch angle to become approximately +10° and the negative shift in x_G causes the pitch angle to oscillate continuously between approximately –10° and –13°.

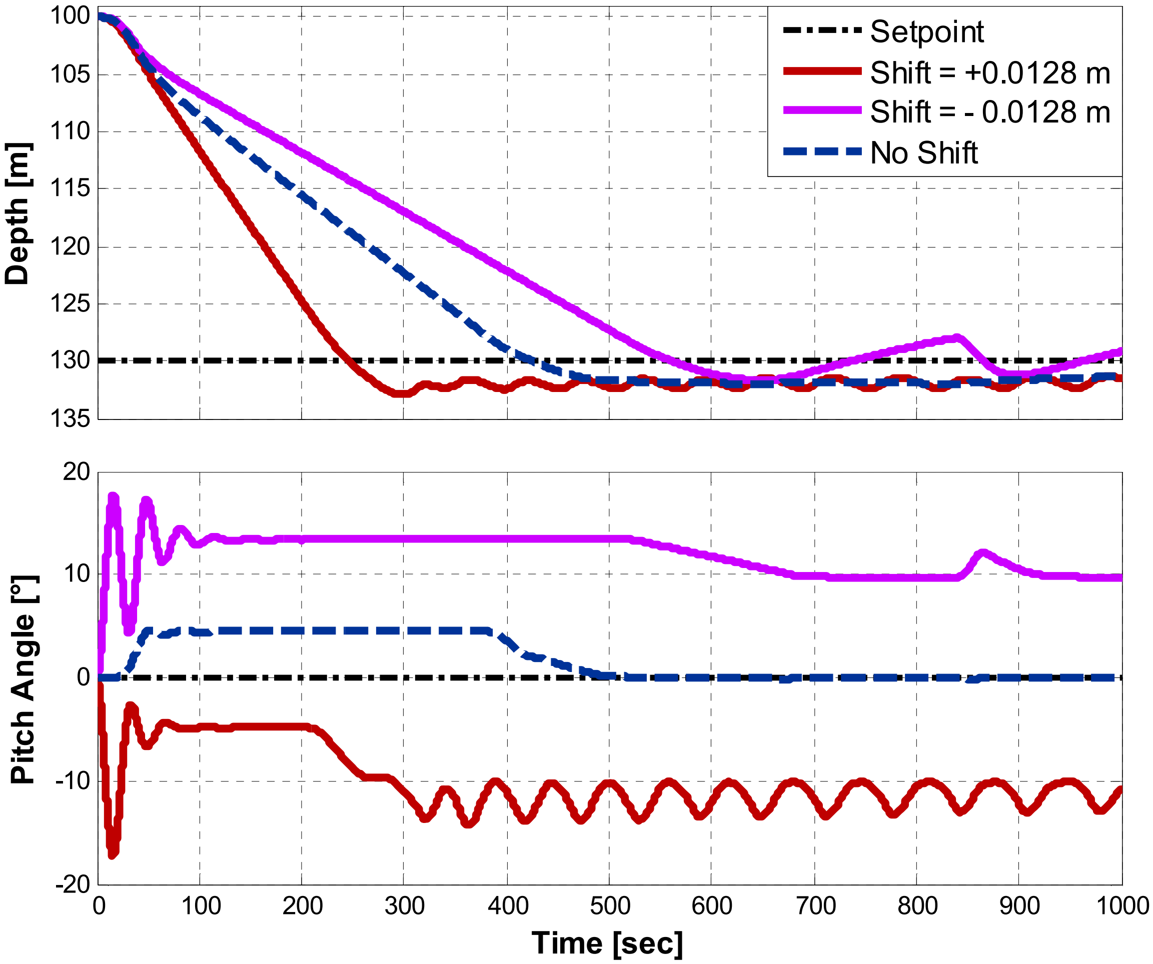

Figure 37 – HPDCB VBS Depth Controller with x_G Shifted – Setpoint Depth Control (K_P = 1, K_D = 40, step change in depth from 100 m to 130 m, "small" ballast tanks)

Figure 38 displays the predicted resource usage for the HPDCB VBS depth controller undergoing no shift, a +0.0128 m shift, and a –0.0128 m shift on an AUV with "small" ballast tanks. While there may not be a large adverse effect on the setpoint depth control of the HPDCB VBS depth controller, it can be seen in Figure 37 that there is a large increase in both water and air mass usage (~870% free-emptying, ~2400% water pump, and ~57% air mass) when undergoing the positive shift. Undergoing a negative shift brings a minor increase in water volume (~39% free-emptying and ~110% water pump) usage and a significant increase in the air mass usage (~49%).

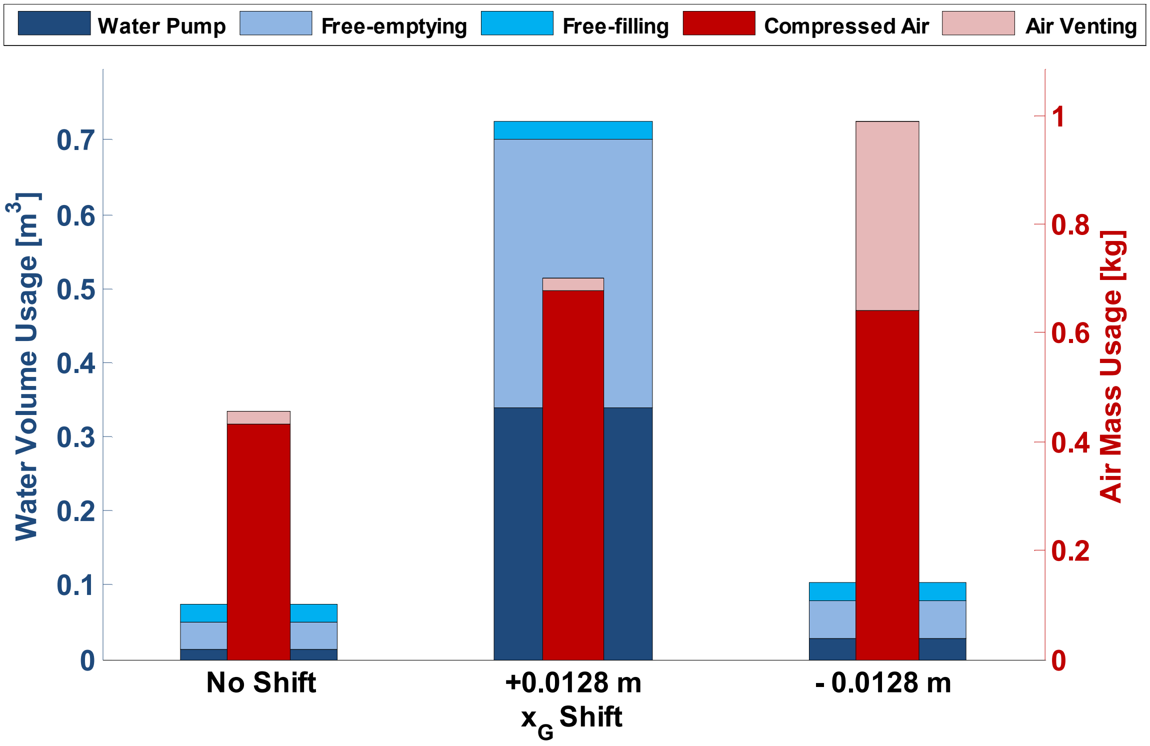

Figure 38 – HPDCB Controller with x_G Shifted – Resource Usage (K_P = 1, K_D = 40, t = 1,000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

Figure 37, shows an an AUVwith a shifting center of gravity x_G. While shifts in the center of gravity x_G can have significant effects on an AUV, it is not common for these shifts to be of a size at which these significant effects can occur. However, within this thesis, simulations will be performed where shifts occur that can produce such significant effects on an AUV. Shifts in x_G can be attributed to loss or shifting of mass during AUV operation, or the AUV may have initially exhibited an x_G offset. As shown in Figure 16 and Figure 39, even small shifts of ±0.0128 m can cause considerable offsets in the resulting AUV pitch angles for both ballast tank control as well as bowplane and sternplane control.

9.3 Creating a Controller to Shift x_G

Knowing the current x_G and the desired x_G, it is possible to calculate the necessary ballast tank water level changes to achieve the required shift in x_G. However, due to limitations in ballast tank size, there will be a maximum shifting limit both in the positive (towards forward) and negative (towards aft) directions. Figure 39 presents the logic for the proposed VBS x_G shifting controller for an AUV with “small” ballast tanks:

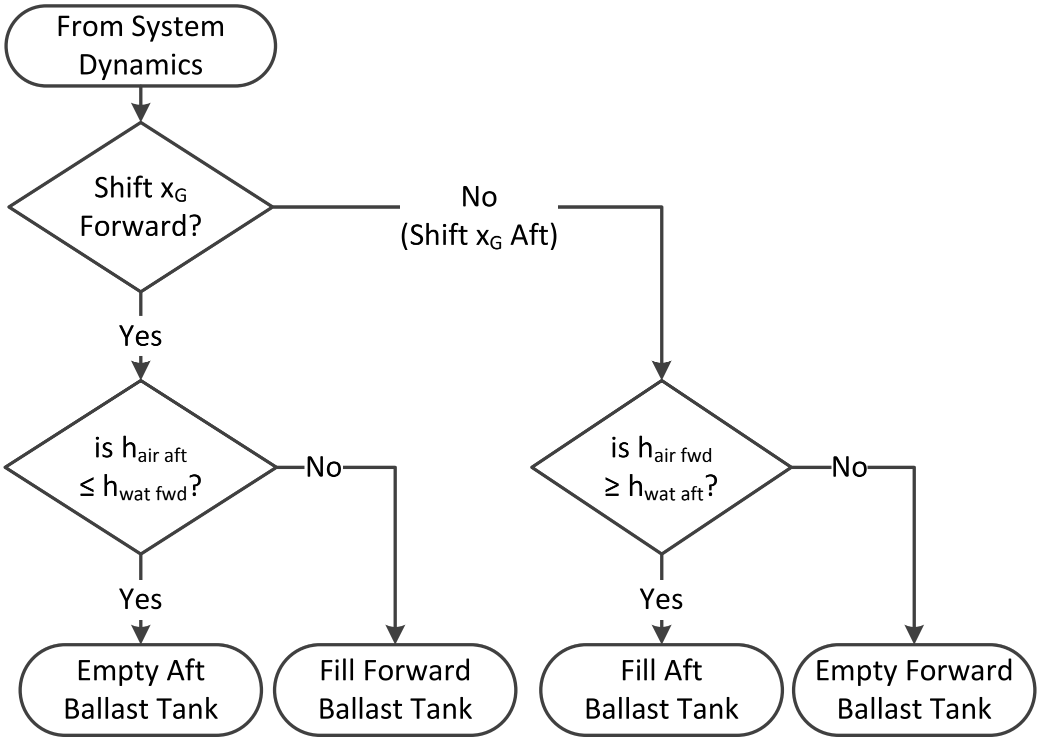
From System Dynamics
Shift x_G Forward?
No (Shift x_G Aft)
Yes
is $h_{air\ aft} \le h_{wat\ fwd}$?
No
Yes
is $h_{air\ fwd} \ge h_{wat\ aft}$?
No
Yes
Empty Aft Ballast Tank
Fill Forward Ballast Tank
Fill Aft Ballast Tank
Empty Forward Ballast Tank

Figure 39 – VBS x_G Shifting Controller Logic Flow Chart for a “Small” Ballast Tank

The logic suggests that when shifting x_G forward, when the height of water in the aft ballast tank is equal or greater then the height of water in the forward ballast tank, then the aft ballast tank will be emptied (no action to the forward ballast tank). However, when the height of water in the aft ballast tank is less than the height of water in the forward ballast tank, the forward ballast tank will be filled. Both scenarios of emptying the aft ballast tank or filling the forward ballast tank aim to shift the x_G forward along the x-axis due to shifting the overall balance of the AUVs mass forward. When desiring to shift x_G aft, the logic is reversed from the forward x_G shift logic: fill the aft ballast tank when the forward water height is equal to or less than the aft water height or empty the forward ballast tank when the forward water height is greater than the aft water height.

Once the desired shift x_G is achieved, the AUV will likely not be neutrally buoyant. When the ballast tank is larger and the x_G shift to correct is also larger, this lack of neutral buoyancy can have a significant impact on the system. For an AUV with "large" ballast tanks, the largest shift from an initial half-filled (h_{water} = 0.375 m) ballast tank will occur when one ballast tank is filled while the other remains half-filled. Any further decrease in water level of the other ballast tank will reduce the effect of losing neutral buoyancy. Filling one ballast tank while emptying the other with identical changes in water level (i.e. filling the forward ballast tank by 0.1 m while emptying aft ballast by 0.1 m simultaneously), the neutral buoyancy can be maintained or at least the effect of lacking it can be kept to a minimum. However, by filling and emptying in this manner, the ability to correct a shifted x_G will be limited. Figure 40 presents the logic for the proposed VBS x_G shifting controller for an AUV with "large" ballast tanks:

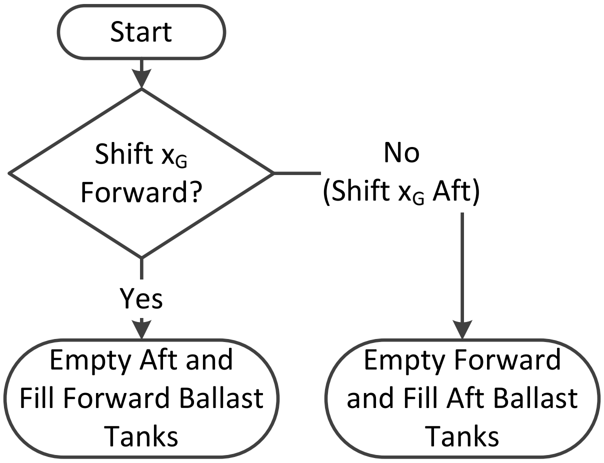
Start
Shift x_G Forward?
No (Shift x_G Aft)
Yes
Empty Aft and Fill Forward Ballast Tanks
Empty Forward and Fill Aft Ballast Tanks

Figure 40 – VBS x_G Shifting Controller Logic Flow Chart for a "Large" Ballast Tank

Comparing the logic flow chart in Figure 40 with that of the logic flow chart in Figure 39, it can be seen that an AUV with "large" ballast tanks requires slightly different logic. This difference in logic occurs because of the requirement for minimizing the loss of neutral buoyancy that can occur due to changing the level of one ballast tank more than the other. Therefore, shifting x_G forward can only be achieved by emptying the aft ballast tank and filling the forward ballast tank. Shifting x_G aft can only be achieved by emptying the forward ballast tank and filling the aft ballast tank.

9.4 Recovering Control Authority

The ability to recover control authority for the bow and stern planes depends upon the amount of weight that the ballast tanks are capable of adding or removing in order to shift the center of gravity. It is expected that a set of larger ballast tanks would shift the center of gravity further forward or aft compared to a set of smaller ballast tanks.

9.4.1 Shifting with a Smaller Ballast Tank

According to the shifting algorithm shown in Figure 39, an AUV with "small" ballast tanks that have volumes of 0.027 m^3 (0.30 m x 0.30 m x 0.30 m) will have a maximum shift capability of approximately ±0.0129 m if the ballast tanks are initially half-filled (h_{water} = 0.15 m). Figure 41 plots the AUV depth, AUV pitch angle, bowplane angle, and sternplane angle as a function of time in response to a 30 m change in the setpoint depth. The simulation involves an AUV utilizing the bow- and sternplane proportional depth and pitch controller with a propulsion force of 150 N ($\bar{u}$ = 0.82 to 0.87 m/s) using "small" ballast tanks to shift the x-axis position x_G. Figure 41 includes simulation results for the bowplane and sternplane proportional depth and pitch controller with no x_G shift which are represented by thick light-blue curves, an x_G shift of −0.0128 m (x_G is shifted towards aft) which are represented by solid red curves, and an x_G shift of −0.0128 m with the proposed VBS x_G shifting controller in effect to correct for this shift in x_G (represented by dashed navy blue curves). The dashed dark-blue curves in Figure 41 clearly show the improved performance achieved with the VBS x_G shifting controller. For the solid red curves in Figure 41 corresponding to an x_G shift without correction, the steady-state depth error is approximately 2.5 m. For this case, the bowplane angle remains at approximately –13.5° while the sternplane angle is approximately 7°. Once the shift correction is introduced via the proposed ballast x_G shifting controller (represented by the dashed navy blue curve in Figure 41), the depth error eventually aligns with that of the non-shifted base case (*depth error* ≅ 0 m) and the pitch angle performance is nearly identical to the non-shifted base case (*depth error* ≅ 0 m). For this case, the bowplane angle is in the range of –0.5°, giving it an additional 13° of control authority in the negative direction without impacting the control authority in the positive direction. As shown in Figure 41, the sternplane angle is also in the range of approximately –0.5° to +0.5°, giving it an additional 6.5° of control authority in the negative direction without significantly impacting the control authority in the positive direction.

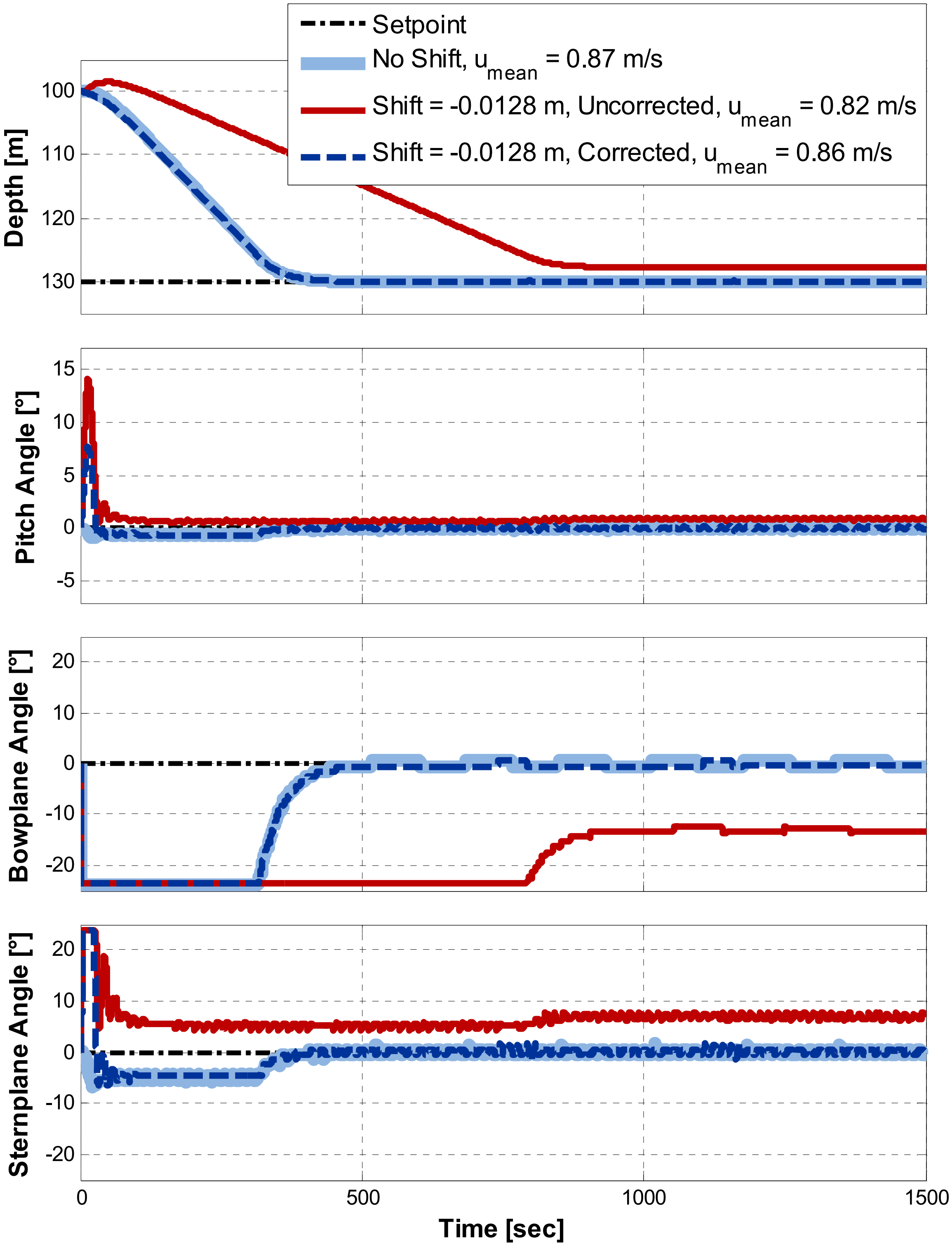

Figure 41 – Small Ballast Tank – No x_G Shift, x_G Shift Uncorrected, and x_G Shift with Correction ($K_{P,VBS}$ = 1, $K_{D,VBS}$ = 40, $K_{P,bow}$ = 0.1, $K_{P,ste}$ = 8, step change in depth from 100 m to 130 m)

9.4.2 Shifting with a Larger Ballast Tank

According to the shifting algorithm shown in Figure 40, an AUV with "large" ballast tanks having volumes of 0.216 m^3 (0.60 m x 0.60 m x 0.60 m) will have a maximum shift capability of approximately ±0.103 m if the ballast tanks are initially half-filled (h_{water} = 0.30 m). Similar to Figure 41 for the small ballast tank, Figure 42 plots the AUV depth, AUV pitch angle, bowplane angle, and sternplane angle as a function of time in response to a 30 m change in the setpoint depth as a function of time for different x_G shift scenarios. For the large ballast tank case, the x_G shift is –0.09 m (x_G is shifted towards aft) and the mean forward velocity $\bar{u}$ ranges between 0.72 and 0.87 m/s. The results shown in Figure 42 ("large" ballast tank) follow a similar trend to those presented in Figure 41 ("small" ballast tank) with the dashed blue curve clearly showing the improved performance achieved with the ballast tank assisted x_G shifting controller. For the red curve shown in Figure 42 corresponding to an x_G shift without correction, the steady-state depth error cannot be calculated as the AUV was not able to maintain the desired trajectory with such a large magnitude in x_G shift. The bowplane angle remains at approximately –23.5° while the sternplane angle is +25°. Once the shift correction is introduced via the proposed ballast x_G shifting controller, the depth error eventually aligns with the non-shifted base case (*depth error* ≅ 0 m) and the pitch angle is also nearly identical to the non-shifted base case (*pitch error* ≅ 0°). The bowplane is in the range of –0.5° to +0.5°, giving it an additional 23° of control authority in the negative direction without significantly impacting the control authority in the positive direction. The sternplane is also in the range of approximately –0.5° to +0.5°, giving it an additional 24.5° of control authority in the positive direction without significantly impacting the control authority in the negative direction.

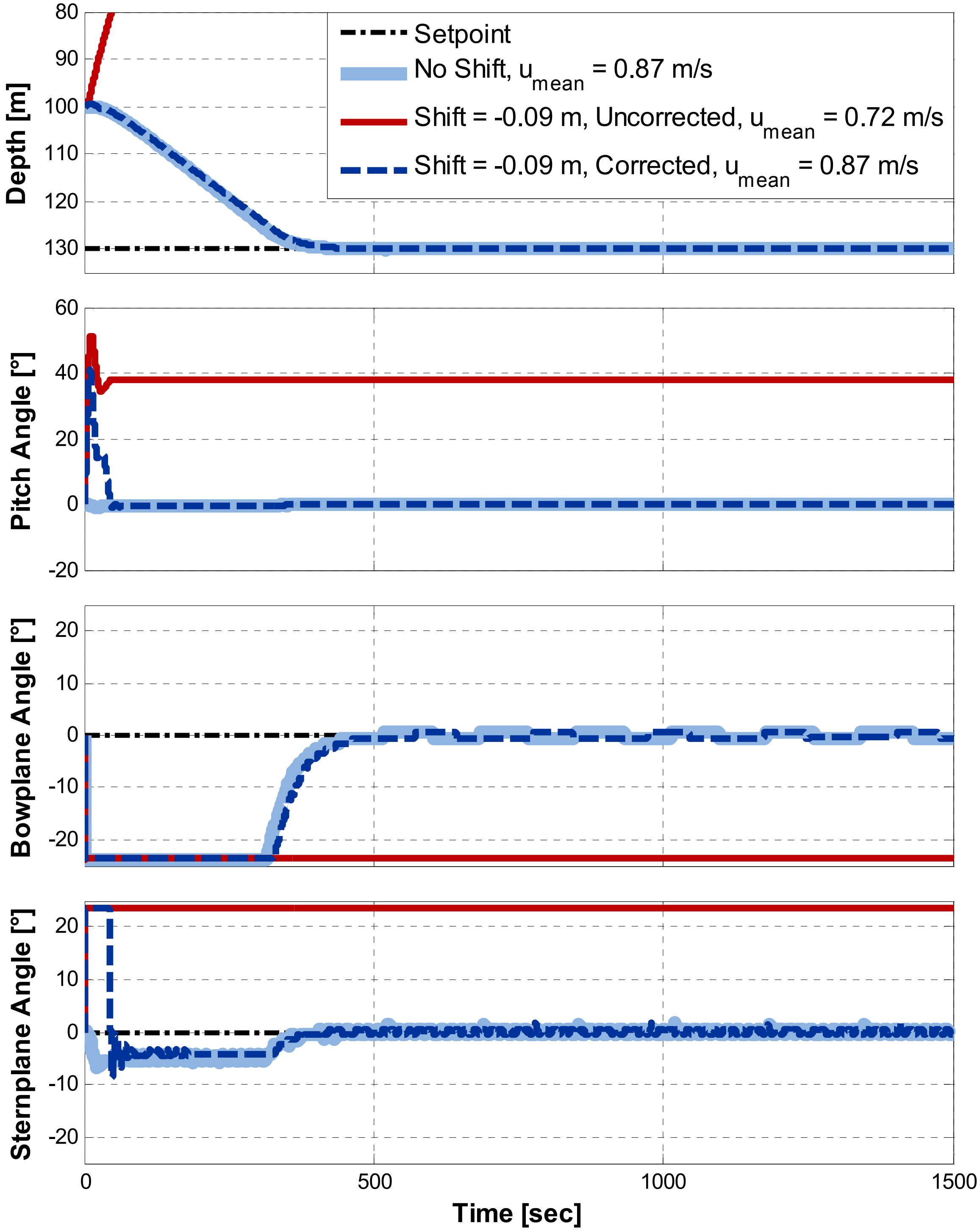

Figure 42 – Large Ballast Tank – No x_G Shift, x_G Shift Uncorrected, and x_G Shift with Correction ($K_{P,VBS}$ = 1, $K_{D,VBS}$ = 40, $K_{P,bow}$ = 0.1, $K_{P,ste}$ = 8, step change in depth from 100 m to 130 m)

9.5 Summary

As shown throughout Chapter 9, both small and large ballast tanks are capable of shifting the center of gravity along the body-fixed x-axis. “small” ballast tanks are capable of compensating for shifts in x_G that are only approximately 0.0129 m in magnitude. The large ballast tanks, on the other hand, were able to shift the center of gravity nearly 0.1 m back towards the hydrodynamic origin. Due to the higher shifting capability of the “large” ballast tanks, a greater amount of control authority could be provided to bowplane and sternplane fins than the smaller shifting capability of the “small” ballast tanks. As the mass of the AUV was shifted, as a result of the VBS x_G shifting controller, control authority was increased. As a result of increased control authority, the bowplane and sternplane proportional depth and pitch controller was more capable of controlling the depth and pitch angle of the AUV.

As an AUV changes its center of gravity, a VBS x_G shifting controller can be useful due to its ability to increase control authority to the bow- and sternplanes that has been lost as a result of a shift in x_G along the x-axis. In situations where an x_G shift of less than 0.0129 m is required, a small ballast solution may be adequate. If water and air mass resource usage is a major concern, then an AUV with “small” ballast tank will be a definite improvement over that of a larger ballast tank, as shown in Figure 22. In situations where an x_G shift greater than 0.0129 m occurs, it is recommended that a ballast tank with a volume greater than 0.027 m^3 (“small” ballast tank) be used, if possible.

Chapter 10 Evaluation of Proposed Control Systems

EXTENDED, DEEP, AND SETPOINT DEPTH PROFILE SIMULATIONS

CONTROLLER OPERATING CONDITIONS

TRADE-OFF ANALYSIS

Overall, the proposed VBS x_G shifting controller is able to utilize the ballast tanks while the AUV is undergoing forward motion and shift the center of gravity along the body-fixed x-axis so that it can be closer to the center of buoyancy, thus improving the following (compared to the uncorrected x_G shifted result, during AUV descent):

- Setpoint depth response
- Pitch angle response
- Decrease in bowplane angle
- Decrease in sternplane angle

Since a larger ballast tank can result in a larger shift, the size of the ballast tank will be an important consideration when determining what shifts are within the realm of possibility for the AUV. The VBS x_G shifting controller cannot be effectively used with the VBS depth controllers as they both utilize the ballast tanks for a different purpose and would conflict with each other. The VBS x_G shifting controller fills and empties the ballast tanks with the sole goal of shifting x_G along the body-fixed x-axis. The VBS x_G shifting controller does not have any regard for the depth error of the AUV. The VBS depth controllers, on the other hand, fill and empty the ballast tanks without any regard for the location of x_G and focus exclusively on achieving the setpoint depth for the AUV.

10.1 Extended, Deep, and Setpoint Depth Profile Simulations

Previous simulations involved 30 m step depth changes from 100m to130 m in order to study trends associated with the different AUV controllers. In order to further test the setpoint depth control for proposed HPDCB VBS controller, additional simulations exhibiting different ascents, descents, and custom trajectories were performed. Note that results from the PD VBS depth controller presented in Chapter 7 were also included to, once again, provide a baseline for comparing and highlighting the benefits that the HPDSB model can provide. The following list summarizes the additional simulations that were performed:

Table 9 – Summary of Additional Simulations

Test Name	Initial Depth m	Setpoint Depth m	Simulation Length sec
Extended Ascent	700	1,000	7,500
Extended Descent	1,000	700	7,500
Deep Ascent	4,900	5,000	5,000
Deep Descent	5,000	4,900	5,000
Custom Trajectory	100	various	3,000

10.1.1 Extended Simulations

It is important to determine whether or not the proposed HPDCB VBS depth controller is capable of handling large ascents and descents. Figure 43 plots the AUV depth as a function of time for a simulation involving a 300 m ascent from 1,000 m to 700 m and a descent from 700 m to 1,000 m. Both PD and HPDCB VBS depth controllers are included in the figure for comparison.

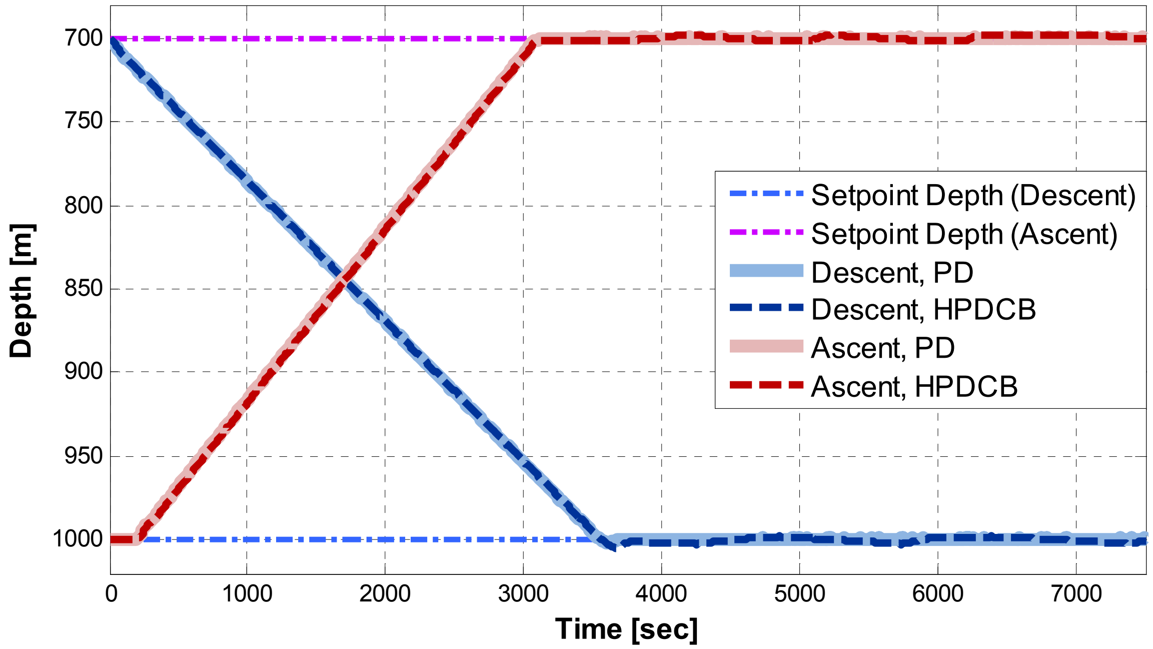

Figure 43 – Extended 300 m Depth Change (Descent & Ascent) – Setpoint Depth Control (K_P = 1, K_D = 40, "small" ballast tanks)

As can be seen, both the PD and HPDCB VBS depth controllers are capable of controlling the AUV to reach and sustain the desired setpoint depth (700 m for ascent, 1,000 m for descent).

While the PD VBS depth controller shows a smaller and more stable setpoint depth, the HPDCB VBS depth controller has considerably less water and air resource usage as can be seen in Figure 44. For the simulation with the 300 m extended descent, the water pump usage of the HPDCB VBS depth controller is 0.12 m^3 (99% decrease from the PD VBS depth controller usage of 8.21 m^3) and the compressed air mass usage is 4.50 kg (45% decrease from the PD VBS depth controller usage of 8.21 kg). For the simulation with the 300 m extended ascent, the water pump usage of the HPDCB is 0.18 m^3 (98% decrease from the PD VBS depth controller usage of 10.26 m^3) and the compressed air mass usage is 6.29 kg (50.0% decrease from the PD VBS depth controller usage of 12.59 kg). The trends between the resource usage of the VBS depth controllers is identical to that found in Chapter 8 suggesting that the HPDCB VBS depth controller is capable of near-equivalent control while reducing resource usage considerably.

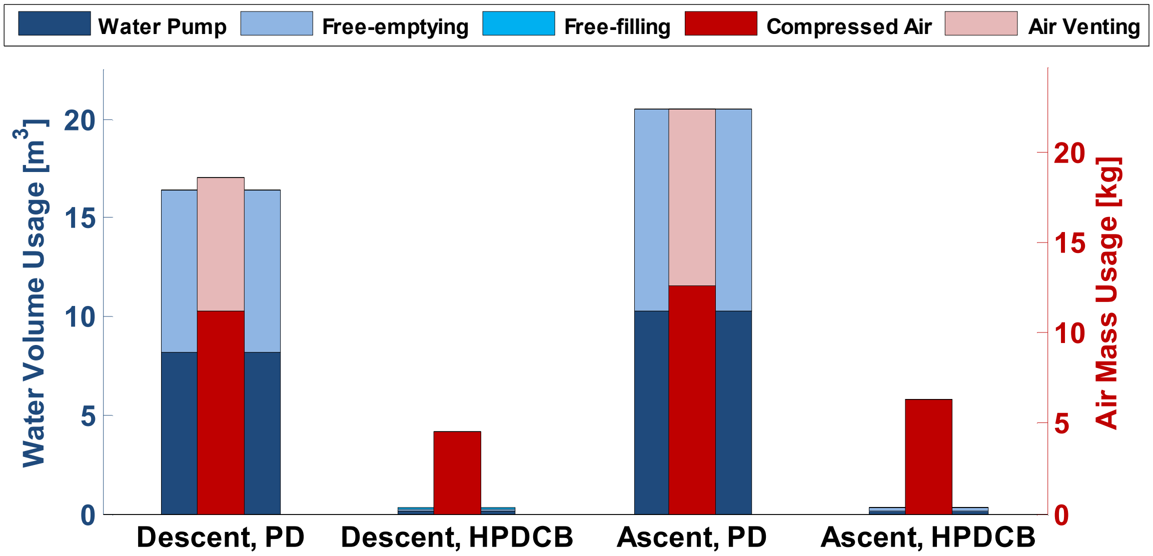

Figure 44 – 300 m Depth Change (Descent & Ascent) – Resource Usage (K_P = 1, K_D = 40, t = 1,000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

Overall, the 300 m depth change simulations show that the HPDCB VBS depth controller, with realistic options, is capable of making large descents or ascents and holding setpoint depth after the descent or ascent.

10.1.2 Deep Simulations

As with the exteneded depth change simulations, it is also important determine whether AUV and the presented HPSCB VBS depth controller are capable of control at a maximum operating depth of 5,000 m. Figure 45 plots the AUV depth as a function of time in response to a 100 m step change in setpoint depth. The dashed red curve represents a 100 m ascent from 5,000 m to 4,900 m and the dashed navy blue curve represents a descent from 5,000 m to 4,900 m. Both PD and HPDCB VBS depth controllers are included in the figure for comparison, represented by the solid light-blue curve and solid light-red curve, respectively. As can be seen, both the PD and HPDCB VBS depth controllers are capable of controlling the AUV to achieve and sustain these setpoint depths.

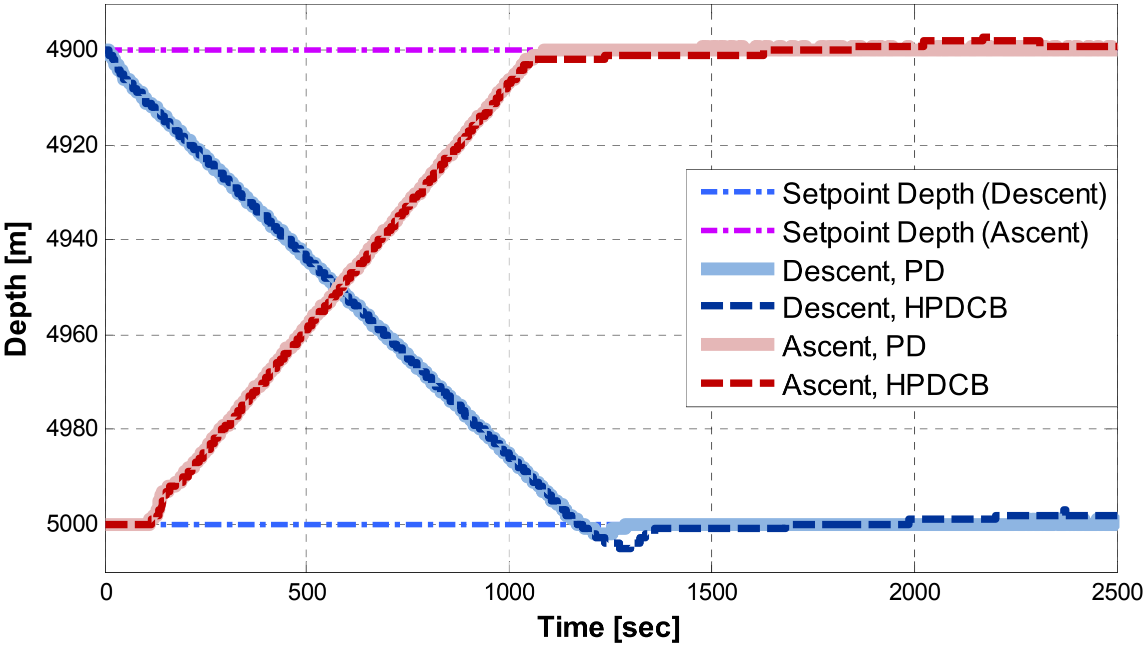

Figure 45 – Deep 100 m Depth Change (Descent & Ascent) – Setpoint Depth Control (K_P = 1, K_D = 40, "small" ballast tanks)

The deep depth change simulations show trends similar to those found in the extended depth change simulations. However, the reduction in air mass usage is lower than that found in the extended simulations. The resource usage of both the PD and HPDCB VBS depth controllers for the descent and ascent simulations can be seen in Figure 36. For the simulation with the 100 m deep descent, the water pump usage of the

HPDCB VBS depth controller is 0.04 m^3 (99% decrease from the PD VBS depth controller usage of 2.55 m^3) and the compressed air mass usage is 22.26 kg (5% decrease from the PD VBS depth controller usage of 23.48 kg). For the simulation with the 100 m deep ascent, the water pump usage of the HPDCB is 0.06 m^3 (98% decrease from the PD VBS depth controller usage of 3.22 m^3) and the compressed air mass usage is 31.74 kg (15% decrease from the PD VBS depth controller usage of 37.23 kg).

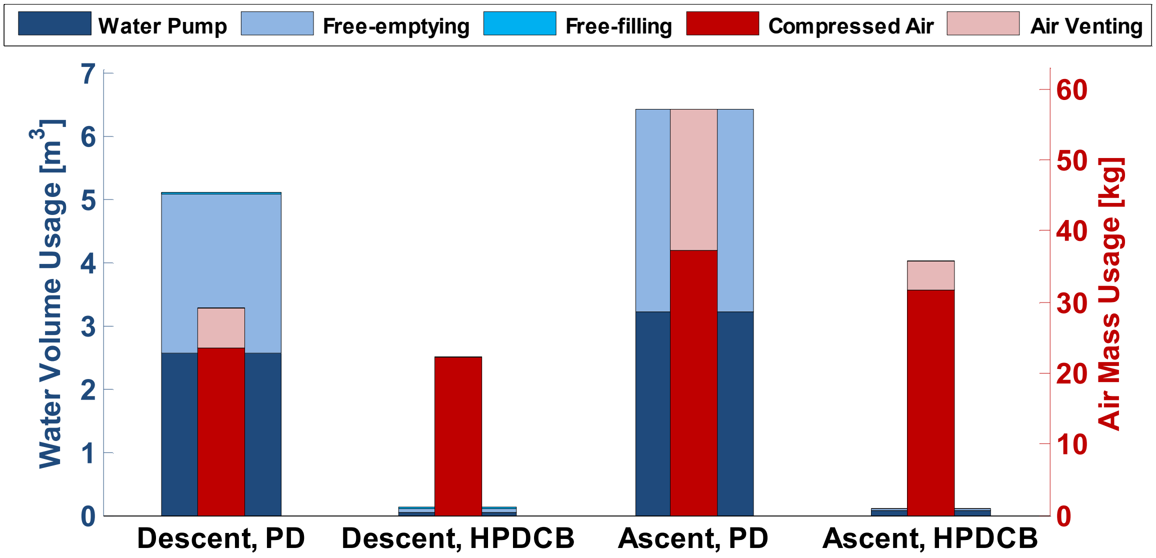

Figure 46 – Deep 100 m Depth Change (Descent & Ascent) – Resource Usage (K_P = 1, K_D = 40, t = 1,000 sec, step change in depth from 100 m to 130 m, "small" ballast tanks)

The increased amount of required air mass is due to the large depth of 5,000 m that the AUV is operating at. At this depth, the pressure is approximately 50.3 MPa compared to a pressure of 10.1 MPa at 1,000 m in the previous extended depth simulations. With the "small" ballast tank, according to the theory found in Chapter 3, to cause the AUV to become neutrally-buoyant at 5,000 m, the ballast tank requires an air mass of 16.14 kg to decrease the water level by 0.15 m (half the ballast tank height) in both ballast tanks. Any air mass usage beyond this point is for controlling the AUV as it moves above and below setpoint depth (or the setpoint depth dead zone). The ballast tanks will also remove more than 0.15 m of water in order to reduce or remove any initial setpoint depth overshooting as can be seen in both the PD and HPDCB descent simulation. In order to completely remove all of the water in the ballast tanks will require a total of approximately 32.28 kg, which is approximately the mass used by the HSBPD VBS depth controller for the descent simulation. Since the ballast tank will be nearly filled while descending, the ballast tank water level will require a decrease of approximately 0.15 m in order to create this neutral buoyancy state. These simulations show that the HPDCB VBS depth controller is capable of near-equivalent control to the PD VBS depth controller while reducing resource usage considerably.

10.1.3 Setpoint Depth Profile

When an AUV is operating, it may be required to follow a pre-determined setpoint depth profile. For the purposes of this thesis, only the depth will be focused on. The setpoint depth profile simulation is performed using all of the realism options that were included in the HPDCB simulations in Chapter 8 (i.e. losses, quantization, minimum water pump and air compressor flow rates, water valve delay of 2 seconds, water pump delay of 5 seconds). Figure 47 plots the custom set of setpoint depth of the AUV as a function of time. It is, of course, not an expected trajectory as it would not be possible for the AUV to make instantaneous descents or ascents.

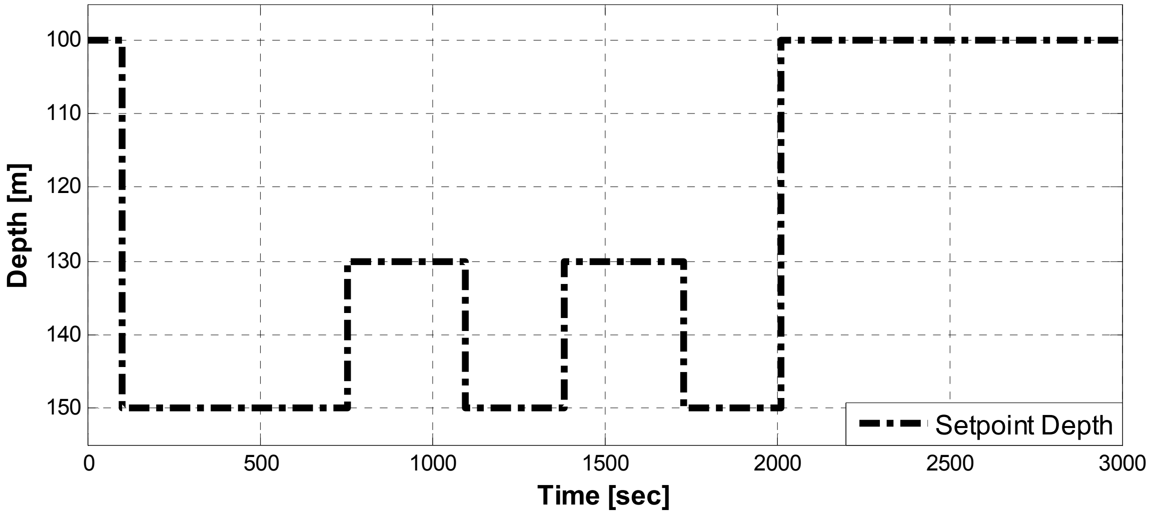

Figure 47 - Custom Setpoint Depth

Figure 48 plots the actual depth of the AUV and the setpoint depth superimposed as a function of time. As observed in Figure 48, the HPDCB VBS controller is able to attain each of the setpoint depths. It should be noted that a dead zone of 2 m was added to setpoint depth amounts which is equivalent to the HPDCB controller's built-in setpoint dead zone.

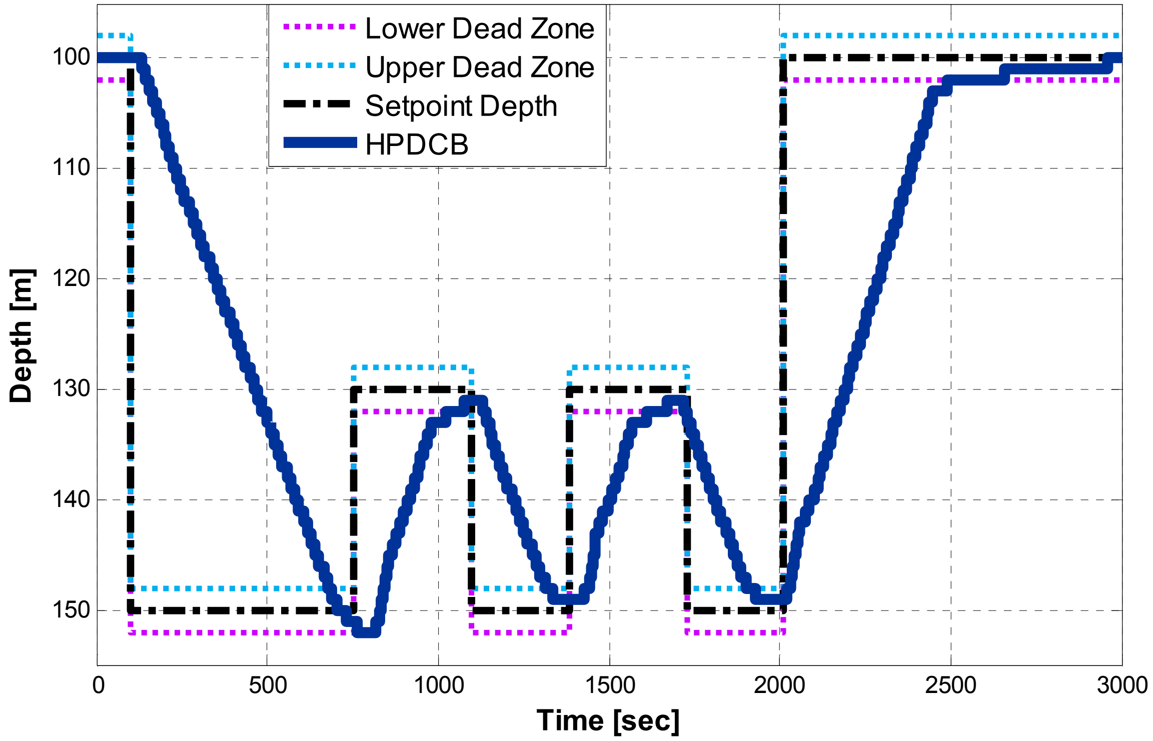

Figure 48 – Custom Setpoint Depth Control (K_P = 1, K_D = 40, various step change in depth, "small" ballast tanks)

From the results of Figure 48, it can be concluded that the HPDCB VBS controller is able to perform stable successful control over the custom set of setpoint depths shown in Figure 47.

10.2 Controller Operating Conditions

Depending on the situation that the AUV is in, different controllers may be utilized to achieve the setpoint control variables. Table 10 lists some of the general scenarios and the corresponding suggested operating conditions of when the specific controllers should be implemented. These operating conditions will allow for an AUV to make the best use its resources and allow it to operate in scenarios requiring multiple forms of control (i.e. bowplane and sternplane depth and pitch control, VBS control).

Table 10 – Suggested Controller Operating Conditions

Controllers	Forward AUV Velocity m/s	Depth Error m	Pitch Error °	x_G Offset m
Bow- and Sternplane Proportional Depth and Pitch Controller (no VBS)	≥ 1.00	≥ 2.0	≥ 5.0	
Proposed HPDCB VBS Depth Controller (no x_G shifting)	< 1.00	≥ 2.0		< 0.05
Proposed Hybrid Bow- and Sternplane Proportional Controller with VBS x_G Shifting	≥ 1.00			≥ 0.05

10.3 Trade-off Analysis

Table 11 shows a summary of advantages and disadvantages of the four main controllers tested in this thesis: bow- and sternplane proportional depth and pitch controller, PD VBS depth controller, proposed HPDCB VBS depth controller, and proposed VBS state-based x_G shifting controller. The check mark represents that the selected feature is part of the specific controller.

Table 11 – Trade-off Analysis for the Four Main Controllers Tested

Features	Controller			
	Bow- & Sternplane Proportional	PD VBS	HPDCB VBS	VBS x_G Shifting
Depth Control	✓	✓	✓	
Free-Emptying Usage		✓	✓	✓
Free-Filling Usage			✓	
Ineffective Over Small Distances	✓			

Features	Controller			
	Bow- & Sternplane Proportional	PD VBS	HPDCB VBS	VBS x_G Shifting
Mechanical Ballast Usage		✔	✔	✔
Minimum Velocity Required	✔			
Modify Centre of Mass: All Directions				✔
Modify Centre of Mass: Body-Fixed Vertically			✔	✔
Pitch Control	✔			✔

Chapter 11 Conclusions and Recommendations

This thesis presents different AUV ballasting control solutions. AUVs typically utilize bowplane and sternplane proportional depth and pitch controller to achieve both setpoint depth and pitch.

11.1 Conclusions

Effective bowplane and sternplane control requires the AUV to have significant forward velocity (~1 m/s or greater). The greater the forward velocity, the more effective the fins are able to provide depth and pitch control.

In order to provide depth control at lower velocities, or without the need for any propulsion at all, a proposed hybrid proportional and derivative (PD) state-based variable ballast control solution is tested and compared with a proportional and derivative (PD) VBS controller. Both PD and hybrid PD state-based ballast controllers are able to control depth by controlling changes in the water levels in the ballast tanks and both ballasting controllers are able to achieve and maintain a setpoint depth.

The design and implementation of a bowplane and sternplane proportional depth and pitch controller was performed to provide a baseline representation for the current configuration of the Theseus AUV. From the simulations, there were several statements that could be made about the performance of this bowplane and sternplane proportional depth and pitch controller. First, higher forward AUV velocities provide faster control response than when the AUV is moving slower. When undamped by the sternplane fins, there was no significant initial depth overshoot at vertical forward velocities such as 0.48 m/s, 1.11 m/s, 1.58 m/s, and 1.95 m/s. When a proportional sternplane component was added to the controller, the depth error control response was slower than without the sternplane component. For this AUV system, it was shown that the most effective bowplane and sternplane proportional control parameters were $K_{P,bow}$ = 0.1 and $K_{P,ste}$ = 8. Both a positive and negative shift of 0.0128 m (at a z-axis z_G of 0.075 m), when moving with an average forward velocity between 0.63 m/s (negative shift) and 0.66 m/s (negative shift), cause the AUV to not be able to attain the setpoint depth or pitch angle. An alternative method of depth control was desired as well as a method of controlling instability due to shifts in the x-axis center of gravity x_G.

The PD VBS controller was designed as a baseline case for comparing the presented hybrid PD condition-based controller. The ideal PD controller parameters were found to be K_P = 1 and K_D = 40. Aspects of realism were introduced into the design of the AUV utilizing a PD VBS controller. These aspects of realism included quantization, losses, operation delays, AUV compression, and vertical disturbance forces. It was shown that, quantization caused increases in free-emptying, water pump, and compressed air mass usages of 5,800%, 5,700%, and 420%, respectively. Losses resulted in 20% increases in both water pump and free-emptying usage, without a significant change in compressed air mass usage. Adding a water valve opening and closing delay of 5 seconds increased the water pump by approximately 4,100 %, free-emptying usage by approximately 4,200% and compressed air mass usage by 210%. AUV compression due to increasing pressure due to depth, water temperature variations, and salinity variations will result in a decrease in AUVvolume and, in turn, decrease in the buoyancy of the AUV. Adding a vertical disturbance force to the AUV caused the ballast tanks to use more percentage of emptying or filling in order to counter the external force. With a positive (downward) vertical disturbance force of 300 N, the “small” ballast tanks were unable to counter, only having a total ballast capability of approximately 271.5 N. All aspects of realism have an adverse effect on the PD controller, especially with respect to resource usage. Therefore, it is desired to design a controller that can perform with similar depth control performance as the PD control, while reducing total water and air mass usage.

The hybrid proportional and derivative condition-based (HPDCB) controller was developed mainly to provide an alternative setpoint depth control to the bowplane and sternplane PD controller. It has also been designed to improve upon the PD controller in terms total resource usage. On top of providing the same forms of control as with the PD controller, it also includes enhanced free-emptying support, free-filling support, neutral buoyancy correction, and depth error dead zone saturation. The HPDCB VBS depth controller provided a response approximately equal to the bowplane and sternplane proportional depth and pitch controller when the AUV is traveling at a forward velocity of 1.00 m/s. Therefore, any velocity that is lower than 1.00 m/s will provide slower performance than the ballast tank control solutions (30 m drop in approximately 350 sec). Results for tests involving quantization, losses, and operation delays, show that, compared with the PD controller, the HPDCB VBS controller was able to decrease free-emptying usage by 98%, water pump usage by nearly 100%, and compressed air mass usage by 76%. These are significant reductions in overall resource usage and paint a picture of success for the feasibility of the HPDCB VBS controller. As with the bowplane and sternplane proportional depth and pitch controller, the PD VBS depth controller and HPDCB VBS depth controller were also both found to incur significant setpoint depth error and increased resource usage; mainly increasing water volume usage for a positive shift and compressed air mass usage for a negative shift.

As mentioned in this thesis, one of the issues with AUVs is that they can be susceptible to shifts in the AUVs center of gravity x_G along the body-fixed x-axis. These shifts are due to changes in the AUVs mass and can be a result of operations such as laying cable. At low enough velocities (i.e. 0.82 m/s), a shift of 0.0128 m positive or negative can cause a loss of bowplane control authority. In order for the AUV to overcome a shift of ±0.0128 m, the bow- and sternplanes must provide an adequate angle. Once the bow- or sternplane has lost control angle authority, as they attempt to control changes in depth and pitch that cannot be achieved at the AUVs current velocity and orientation, it is unable to perform to its full extent. The proposed VBS x_G shifting controller is able to restore nearly all of the bowplane and sternplane control authority by correcting the x_G shift to the desired value (where x_G is directly below x_B). With a propulsion force of 150 N, the uncorrected x_G shifted AUV has a setpoint depth bowplane angle of approximately -13.5° and a sternplane angle of approximately +7°. After performing the shift correction for x_G, the depth and pitch control performance of the bow- and sternplane controller is approximately the same as the baseline (non-shifted) case. By use of the x_G shifting controller, the ballast tanks are able to provide the bowplane and sternplane fins with nearly full control authority (±0.5 m and ±0.5°).

Simulation results involving a large ballast tank show that, while it is able to perform larger shifts of x_G, the resource usage increases significantly. Control authority will be more important at lower velocities or during any scenario where the bow- or sternplanes are near saturation and have little control authority due to large shifts in the x_G.

Extended, deep, and custom trajectory simulations were performed in order to study the depth control performance and resource usage of the AUV in simulations involving larger depth changes (from the previous simulations with depth changes of 30 m), simulation times (greater than 1,000 sec), and complex sets of setpoint depths. It was shown that the HPDCB VBS depth controller is capable of performing 100 m descents and ascents near a depth of 2,000 m. The HPDCB VBS depth controller is also capable of performing 300 m ascents and descents while controlling ballast tank contents.

It is proposed that, depending on the operating conditions, different controllers should be used for different conditions. By utilizing the suitable controller, the AUV may be able to undergo efficient and effective control by utilizing the available resources and given trajectory appropriately. The simulations results show that the AUV should also be able to properly adapt to shifts in x_G along the body-fixed x-axis up to ±0.0129 m for the assumptions and conditions tested, and 0.027 m^3 ("small") and 0.216 m^3 ("large") ballast tanks studied in this thesis. The controllers proposed herein have the potential to greatly increase the autonomy of AUVs engaged in long duration missions where they are out of contact with a human operator.

11.2 Contributions

Contributions include an in-depth study of current and proposed VBS controllers on a large AUV. Proposed VBS controllers include a hybrid proportional derivative condition-based VBS depth controller and a VBS x_G shifting controller. The HPDCB VBS depth controller has been designed to improve upon a basic PD VBS depth controller by reducing overall resource usage while retaining depth control performance. In this case, resource usage refers to the water and air resources that are brought both into and out of the ballast tank via a valve, water pump, or compressed air cylinder. The VBS x_G shifting controller is expected to provide the bow- and sternplane controller with assistance by shifting the center of mass towards the hydrodynamic origin, thus reducing potential pitch angle oscillations and offset. By decreasing these pitch angle oscillations and offsets, the depth and pitch angle control authority of the bow- and sternplane fins were increased. Provided discussion and simulation results showed the capabilities of the proposed VBS controllers including depth control, resource usage, and x_G shifting.

11.3 Recommendations for Future Work

There are several aspects of this AUV system that can be improved on to create a more complete and accurate model. These recommendations include: removing system symmetry, improved air mass venting, improved ballast tank features, and full system dynamics. Each of these recommendations will be discussed briefly.

11.3.1 Remove System Symmetry

In order to maintain a simpler system, symmetry has been introduced for the PD VBS depth controller and HPDCB VBS depth controller. This symmetry keeps the actions in the forward ballast tank and aft ballast tank identical. Due to valve and water pump losses, which are relatively random and likely different for each valve or pump, each ballast tank may end up filling differently. As a result, they will require the calculation and storage of unique parameters for each during simulations. By allowing the system to overcome this symmetry, a more accurate overview of the system performance can be achieved. Also, this enhancement will give greater overall control of the system and allow potential issues, such as water valve malfunctions, to be partially resolved.

11.3.2 Improved Air Mass Venting

Air mass venting is currently a process whereby air flows out of the ballast tank at a specific constant rate. In order to improve the process by which air venting occurs, it should take place similar to that of water free-filling and free-emptying out of the ballast tanks. Air will flow out of the ballast tank at a rate based upon the pressure gradient driving it, the pressure of the external ocean water, and the size of opening. In order to obtain a more appropriate air mass venting model, it is suggested that small-scale experimental tests be performed to determine an approximate rate that can be used. By having an air mass venting rate that is dynamic and based on several system parameters, the realism of the modelled system can be improved.

11.3.3 Improved Ballast Tank Features

There are several approximations made in the current ballast tank models. The following are the various improvements that can be made on this system:

- **Improved Free-Fill Free-Empty Rate Approximations**

Currently, the free-fill free-empty rate of the ballast tank is made on an approximation related to both the pressure difference of the ballast tank $\left(P_{diff} = P_{water} - P_{air}\right)$ and a linear extrapolation of desired time to free-fill/empty a ballast tank (currently, 10 sec). One of the methods that can be used to develop a better understanding of the free-filling/emptying of a ballast tank is through direct experimentation. Using *similarity of flows*, relating to Reynolds number, model-scale flows can be used to approximate full-scale flows to obtain a more accurate approximation of the free-fill/empty rates of the ballast tanks.

- **Valve, Water Pump, and Compressed Air Cylinder Losses**

Currently, the calculated losses in this system are only approximations. They are percentage-based without respect to the dynamics or geometrical design of the system. Valve, water pump, and compressed air cylinder losses will occur due to a number of reasons including friction, pressure losses, and component geometry. A better understanding of the specific components used in this system will help in determining more accurate and appropriate losses incurred.

- **Ballast Tank Angle**

Water within the ballast tank will eventually (with enough time and lack of any further motion) become the same as the angle of the AUV due to the gravitational effect on the water. Depending on the angle, this reality may end up having an adverse effect on the filling and emptying of the ballast tanks.

- **Filling and Emptying Effects**

Filling and emptying effects, including water sloshing within the ballast tank and highly turbulent flows can be present within the ballast tank system. Further study can show these effects may impact the system.

- **Tracking the Air Mass of the Compressed Air Cylinder**

Compressed air cylinders have a static volume and, therefore, they have a maximum air mass capacity. The pressure of the air in the compressed air cylinder can be found by Equation *(20)*. As air is supplied into the ballast tank, the pressure of the compressed air cylinder will decrease. As the pressure decreases, the capability for the compressed air cylinder to add more air will also reduce. If the pressure of the compressed air cylinder ever becomes equal to or less than the pressure of the air within the ballast tank (or water if completely filled), the air will not be able to flow in and the compressed air cylinder can be damaged due to back pressure. The impact of tracking air pressure becomes more important in longer simulations where large amounts of air mass are required for ballasting efforts, or where large depths are travelled to where a large quantity of air mass is required to cause the ballast tank to become negatively buoyant (where AUV buoyancy > AUV mass). In these simulations, it will be required to keep track of how much remaining air is in the ballast tank so that calculations can be performed as to whether or not specific tasks can be performed. Having an AUV go to a depth of 5,000 m in hopes that it will use air mass to remove water from the ballast tanks to become negatively buoyant, just to find there is not enough air mass remaining, would be an example where advanced planning is imperative. AUV's utilizing "large" ballast will have to focus more on remaining air mass quantity as the ballast tanks will require and consume far more air mass resource, as outlined in Chapter 3.

Tracking of the remaining air in the compressed air cylinder has not been performed in any of the simulations in this thesis in order to observe trends of setpoint depth control and resource usage without complicating the results with this resource-limiting factor.

11.3.4 <u>Full System Dynamics</u>

Currently, the modeled system only incorporates two dimensions of motion. It is recommended that the system dynamic and hydrodynamic equations of motion be modified to allow for the inclusion of the third dimension of motion. This third dimension will integrate translation along the y-axis and rotation around the roll and yaw axes into the current system.

References

Thesis References

[1] *Theseus – Cable Laying Autonomous Underwater Vehicle*. Port Coquitlam, Canada: International Submarine Engineering Ltd., (2011). http://www.ise.bc.ca/theseus.html (accessed February 1, 2012).

[2] *ISE AUVs – Arctic Explorer Autonomous Underwater Vehicle*. Port Coquitlam, Canada: International Submarine Engineering Ltd., (2012). http://www.ise.bc.ca/pdfs/ISE_Arctic_Explorer_AUV_Datasheet_2012.pdf (accessed March 12, 2012).

[3] Xu, M., & Smith, S.M. (1994). *Adaptive Fuzzy Logic Depth Controller for Variable Buoyancy System of Autonomous Underwater Vehicles*. Roca Raton, USA: Advanced Marine Systems Group, Florida Atlantic University, pg. 1191-1196.

[4] Tangirala, S., & Dzielski, J. (2007). *A Variable Buoyancy Control System for a Large AUV.* USA: IEEE Journal of Oceanic Engineering, Vol. 32, No. 4, pg. 762-771.

[5] Sumantr, B., & Karsiti, M.N. (2008). *Development of Variable Ballast Mechanism for Depth Positioning of Spherical URV.* Malaysia: Universiti Teknologi PETRONAS.

[6] Byron, J., & Tyce, R. (2007). *Designing a Vertical-Horizontal AUV for Deep Ocean Sampling*. Narragansett, USA: University of Rhode Island.

[7] Ferguson, J., Pope, A., Butler, B., & Verrall, R.I. (1999). *"Theseus AUV-two record breaking missions,"* Sea Technology, vol. 2, no. 40, pp. 65-70.

[8] Feldman, J. (1979). *DTNSRDC Revised Standard Submarine Equations of Motion*. Bethesda, USA: David W. Taylor Naval Ship R&D Center.

[9] Scherer, T. *Irrigation Water Pumps*. North Dakota State University: 1993.

http://www.ag.ndsu.edu/pubs/ageng/irrigate/ae1057w.htm (accessed February 1, 2012).

[10] Dauphinee, T.M., *"Deep Ocean Temperature Profile Measurements"*

[11] Butler, B. *Field Trials of the Theseus AUV*. ISE Research, Ltd., 1999. http://www.ise.bc.ca/auv1002.html (accessed February 1, 2012).

[12] Butler, B., & Hartley, P. *AUV Fiber Optic Cable Laying*. ISE Research, Ltd., 2008. http://www.ise.bc.ca/auv1001.html (accessed February 1, 2012).

[13] Butler, B., & Hertog, V. Theseus: *A Cable-Laying AUV*. ISE Research, Ltd., 1999. http://www.ise.bc.ca/auv1003.html (accessed February 1, 2012).

[14] Young, H., Freedman, R., Sandin, T, & Ford, A. *University Physics*. 13th Ed., Section 11.4.

[15] Fredlund, D. *Density and Compressibility of Air-water Mixtures*. Canadian Geotech. Journal, Vol. 13, 1976, pg. 386 – 396.

[16] Rudnick, D., Davis, R., et al. *Underwater Gliders for Ocean Research*. Marine Technology Society Journal: 2004, pg. 48-59, Volume 38, Number 1.

[17] Burcher, R., & Rydill, L. *Concepts in Submarine Design.* Cambridge: Cambridge University Press, 1994.

[18] Omega Engineering, Inc. *Process Pressure Measurement: Pressure Gauges & Switches.* http://www.omega.com/literature/transactions/volume3/pressure2.html (accessed February 1, 2012).

[19] Analog Devices, Inc. *ADIS16405: High Precision Tri-Axis Gyroscope, Accelerometer, Magnetometer*. 2009. http://www.analog.com/static/imported-files/data_sheets/ADIS16400_16405.pdf (accessed February 1, 2012).

[20] Sapcon Instruments. *MPROF Rate of Flow Indicator*. http://www.sapconinstruments.com/level-measurement/rof/72#td (accessed February 1, 2012).

[21] Sierra Instruments, Inc. *Smart Trak 2 Model 100 Mass Flow Meters & Controllers*. http://www.sierrainstruments.com/products/c100.html (accessed February 1, 2012).

[22] Hughes, P.C. (1986). *Spacecraft Attitude Dynamics*. USA: John Wiley & Sons.

Recommended Reading

[1] Allen, R., & Merry, S. (1993). *Simulation of the Hydrodynamics of Underwater Vehicles as an Aid to Control System Design.* Department of Mechanical Engineering, University of Southampton / The Institution of Electrical Engineers, pg. 1-3.

[2] Anouck R. Girard, Joao B. Sousa (2007). *Autopilots for Underwater Vehicles - Dynamics, Configurations, and Control*. University of Michigan / Manuscript, pg. 1-6.

[3] Appleby, B., & Bonnice, W. (1990). *Robustness Analysis Methods for Underwater Vehicle Control Systems.* The Charles Stark Draper Laboratory, Inc., pg. 74-80.

[4] Beer, F.P. (1998). *Vector Mechanics for Engineers: Statics* (Third SI Metric Edition). Toronto, Canada: McGraw-Hill Ryerson.

[5] Beer, F.P. (1999). *Vector Mechanics for Engineers: Dynamics (Third SI Metric Edition).* Toronto, Canada: McGraw-Hill Ryerson.

[6] Binns, P., & Elgersma, M. (2004). *Statistical Verification of Two Non-linear Real-time UAV Controllers*. Honeywell Laboratories / Proceedings of the 10th IEEE Real-time and Embedded Technology and Applications Symposium, 10 pages.

[7] Bo, W., & Bing-jie, G. (2008). *Research on Motion Simulation System for Underwater Vehicle*. College of Shipbuilding Engineering, Harbin Engineering University / Chinese Control and Decision Conference (CCDC), pg. 3349-3352.

[8] Craven, P.J., & Sutton, R. (1998). *Neurofuzzy Control of a Non-Linear Multivariable System*. University of Plymouth / UKACC International Conference on CONTROL '98, No. 455, pg. 531-536.

[9] Gertler, M., & Hagen, G.R. (1967). *Standard Equations of Motion for Submarine Simulation*. Naval Ship Research and Development Center.

[10] Glad, S.T., & Helmersson, A. (2003). *Uncertain LTI-models for Linear Control Design of Non-Linear Systems*. Proceedings of the 40th IEEE Conference on Decision and Control, pg. 2692-2694.

[11] Herman, P. (2008). *Ocean Engineering: Transformed equations of motion for underwater vehicles* (Vol. 36, pg. 306-312). Poznań, Poland: Elsevier.

[12] Hong, S.W., & Lee, P.M. (2000). *Design of an Underwater Vehicle-Mounted Manipulator System and Non-Regressor Based Adaptive Control*. Ocean Development System Research Center (KRISO) / Proceedings of the Tenth International Offshore and Polar Engineering Conference, pg. 314-319.

[13] Hong-jian, W., & Xiao-cheng, S. (2004). *A Semi-physical Virtual Simulation System for AUV*. Harbin Institute of Technology, pg. 1560-1563.

[14] Jantapremjit, P., & Wilson, P.A. (2008). *Control and Guidance Approach Using an Autonomous Underwater Vehicle*. International Journal of Maritime Engineering / Royal Institution of Naval Architects, Vol. 150, No. 2, pg. 1-11.

[15] Kamakura, Y., & Sasajima, H. (2001). *Chapter 1 - Practical Synthesis for Ships and Floating Systems*. Practical Design of Ships and Other Floating Structures / Elsevier Science Ltd., Vol. 1, pg. 29-356.

[16] Khanmohammadi, S., & Alizadeh, G. (2007). *Design of a Fuzzy Controller for Underwater Vehicles to Avoid Moving Obstacles*. Department of Control Engineering, University of Tabriz, 6 pages.

[17] Kreyszig, E. (2006). *Advanced Engineering Mathematics* (Ninth Edition). New York, USA: John Wiley & Sons.

[18] Kumar, G.V.N., & Rao, K.A.G. (2008). *Robustness of Fuzzy Logic based Controller for Unmanned Autonomous Underwater Vehicle*. IEEE Region 10 Colloquium and the Third Conference on Industrial and Information Systems, pg. 1-6.

[19] Lam, W.C., & Ura, T. (1996). *Non-Linear Controller with Switched Control Law for Tracking Control of Non-Cruising AUV*. University of Tokyo, pg. 78-85.

[20] Lee , S.W., & Hwang, Y.S. (2003). *A Development of 3000-ton Class Submarine and the Study on its Hydrodynamic Performances*. Proceedings of the Thirteenth International Offshore and Polar Engineering Conference, pg. 363-368.

[21] Lee, P.M., & Jun, B.H. (2006). *Navigation and Control System of a Deep-sea Unmanned Underwater Vehicle 'HEMIRE'*. Maritime and Ocean Engineering Research Institute, pg. 1-8.

[22] Li, J.H., & Jun, B.H. (2005). *A Hierarchical Real-time Control Architecture for a Semi-autonomous Underwater Vehicle*. Korea Research Institute of Ships & Ocean Engineering, Volume 32, pg. 1631-1641.

[23] Li, J.H., & Lee, P.M. (2004). *An Adaptive Nonlinear Controller for Diving Motion of an AUV*. Korea Research Institute of Ships & Ocean Engineering, pg. 282-287.

[24] Liles, E.G., & Dai, R. (1989). *Semisubmersible Ballast Systems - A Practical Evaluation Method*. The Society of Naval Architects and Marine Engineers, Spring Meeting / STAR Symposium, Vol. 14, pg. S6-5-1 - S6-5-14.

[25] Lv, C. (2009). *Improved PD Controller for AUV Based on MPSO, Hydrodynamic Model*. State Key laboratory of AUV / International Asia Conference on Informatics in Control, Automation and Robotics, pg. 24-28.

[26] McCluskey, D.K., & Holdo, A.E. (2005). *A Review of Ballast Water Technologies*. Journal of Marine Design and Operations, No. B9, pg. 21-29.

[27] Moattari, M., & Khayatian, A. (2008). *A Neural Network Controller for Diving of a Variable Mass Autonomous Underwater Vehicle*. SICE Annual Conference, The University Electro-Communications, pg. 1339-1344.

[28] Nishida, S., & Ishii, K. (2007). *Self-Organizing Decision-Making System for AUV*. The University of Kitakyushu / Kyushu Institute of Technology, pg. 506-511.

[29] Ogata, K. (2002). *Modern Control Engineering (Fourth Edition)*. Prentice Hall, USA: Prentice Hall.

[30] Oh, M.H., & Oh, J.H. (2002). *Homing and Docking Control of AUV Using Model Predictive Control*. Korea Advanced Institute of Science and Technology, 5 pages.

[31] Payne, K.L. (1998). *Bias Effects on Motion Stability of Submersible Vehicle*. Naval Postgraduate School / Thesis, pg. i-viii, 1-57.

[32] Pohler, C.H., & Bement, A.A. (1969). *Submarine Main Ballast Tanks - Theory and Methods for Refined Structural Design*. Marine Technology, Vol. 6, pg. 171-196.

[33] Sagahyroon, A., & Jarah, M.A. (2004). *Design and Implementation of a Low Cost UAV Controller*. IEEE International Conference on Industrial Technology, pg. 1394-1397.

[34] Santhakymar, M., & Asokan, T. (2008). *Coupled, Non-linear Control System Design for Autonomous Underwater Vehicle (AUV)*. 10th International Conference on Control, Automation, Robotics and Vision / Indian Institute of Technology Madras, pg. 2309-2313.

[35] Sanyal, A.K., & Chyba, M. (2009). *Robust Feedback Tracking of Autonomous Underwater Vehicles with Disturbance Rejection*. American Control Conference, pg. 3585-3590.

[36] Sayyaadi, H., & Ura T. (1999). *Multi Input - Multi Output System Identification of AUV Systems by Neural Network*. Institute of Industrial Science, The University of Tokyo, pg. 201-208.

[37] Shibuya, K., & Kado, Y. (2006). *Underwater Robot with a Buoyancy Control System Based on the Spermaceti Oil Hypothesis*. Department of Mechanical and Systems Engineering, Ryukoku University, pg. 3012-3017.

[38] Shimmin, D.W., & Lucas, J. (1993). *Control Strategies for Underwater Vehicle Autopilots*. Department of Electrical Engineering and Electronics, The University of Liverpool / The Institution of Electrical Engineers, pg. 1-3.

[39] Silverstre, C., & Pascoal, A. (1998). *Plant - Controller Optimization with Applications to Integrated Surface Sizing and Feedback*. Institute Superior Tecnico / Proceedings of the American Control Conference, pg. 1604-1644.

[40] Suzuki, H., & Kato N. (2007). *Motion Simulation of an Underwater Vehicle with Mechanical Pectoral Fins Using a CFD-based Motion Simulator*. Osaka University, Graduate School of Engineering, pg. 384-390.

[41] Syahroni, N., & Seo, Y.B. (2007). *Open Control Platform Implementation for Autonomous Underwater Vehicle*. SICE Annual Conference 2007, Kagawa University, pg. 2042-2047.

[42] Tan, K.M., & Liddy, T. (2008). *The Advancement of an Autonomous Underwater Vehicle (AUV) Technology*. School of Mechanical Engineering, The University of Adelaide / Defence Science and Technology Organization, pg. 336-341.

[43] Tennakoon, W.T.M.S., & Munasinghe, S.R. (2008). *Design and Simulation of a UAV Controller System with High Maneuverability*. Department of Electronics and Telecommunications, University of Moratuwa / ICIAFS, pg. 431-436.

[44] Thomasson, P.G. (2000). *Equations of Motion of a Vehicle in a Moving Fluid*. Journal of Aircraft / Cranfield University, Vol. 37, No. 4, pg. 630-639.

[45] Tiano, A. (2004). *Comparison of Non Linear Identification Methods for Underwater Vehicles*. University of Pavia, Department of Informatics and Systems, pg. 549-552.

[46] Tomasi, J., & Ro, K. (2006). *Analytical Modeling and Simulation of an Autonomous Underwater Vehicle with Five Control Systems*. Department of Mechanical and Aeronautical Engineering / Modeling and Simulation for Military Applications, Vol. 6228, pg. 1-10.

[47] Tung, J.M., & Guo, M.F. (2007). *Design of an Underwater Glider with Fore and Aft Buoyancy Engines*. Department of Engineering Science and Ocean Engineering, pg. 446-450.

[48] Vieira, J., & Mota, A. (2004). *Parameter Estimation of Non-Linear Systems With Hammerstein Models Using Neuro-Fuzzy and Polynomial Approximation Approaches*. Ecole Superior de Tecnologia de Castelo Branco / Universidade de Aveiro, pg. 849-854.

[49] Wasserman, K.S., & Mathueu, J.L. (2003). *Dynamic Buoyancy Control of an ROV using a Variable Ballast Tank*. Massachusetts Institute of Technology / MIT Department of Ocean Engineering, pg. 2888-2893.

[50] Weiquan, H., & Yanling, H. (2004). *Research of the Inertial Navigation System with Variable Damping Coefficients Horizontal Damping Networks*. Harbin Engineering University, pg. 1272-1276.

[51] Wilson, W., & Chang, P. (2006). *Computational and Experimental Analysis of Ballast Water Exchange*. American Society of Naval Engineers / Naval Engineers Journal, Vol. 118, No. 3, pg. 25-36.

[52] Wu, J., & Zhang, M.J. (2007). *Research of System Identification Method for Underwater Vehicle Based on Neural Network*. Harbin Engineering University / Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, pg. 705-710.

[53] Xia, G., & Tang, L. (2009). *Design of a Hybrid Controller for Heading Control of an Autonomous Underwater Vehicle*. College of Automation, Harbin Engineering University, 5 pages.

[54] Xu, J., & Zhang, M. (2008). *Neural Network Modeling and Generalized Predictive Control for an Autonomous Underwater Vehicle*. College of Mechanical and Electrical Engineering, Harbin Engineering University, pg. 487-491.

[55] Zhao, S., & Sun, J. (2008). *Finite-time Stability of Linear Time-varying Singular Systems with Impulsive Effects*. Tongji University / International Journal of Control, Vol. 81, No. 11, pg. 1824-1829.

Appendix A Full Derivation for the Equations of Motion

The system dynamics determine the angular acceleration, velocity, and positions. The following are a list of parameters used in the derivation (refer to the List of Abbreviations and Symbols Used for additional parameter definitions):

Weight

- $W = [W_X, W_Y, W_Z] = [0,0, W_Z]$ *(52)*

where $W_Y = 0, W_Y = 0$.

Buoyancy

- $B = [B_X, B_Y, B_Z] = [0,0, B_Z]$ *(53)*

where $B_X = 0, B_Y = 0$.

Linear Momentum Components

- $v_o = [u, v, w] = [u, 0, w]$
- $\dot{v}_o = [\dot{u}, \dot{v}, \dot{w}] = [\dot{u}, 0, \dot{w}]$ *(54)*

where $v = 0, \dot{v} = 0$.

Angular Momentum Components

- $\omega = [p, q, r] = [0, q, 0]$
- $\dot{\omega} = [\dot{p}, \dot{q}, \dot{r}] = [0, \dot{q}, 0]$ *(55)*

where $p = 0, r = 0, \dot{p} = 0, \dot{q} = 0$.

Center of Gravity

- $r_G = [x_G, y_G, z_G] = [x_G, 0, z_G]$ *(56)*

where $y_G = 0$.

Center of Buoyancy

- $r_B = [x_B, y_B, z_B] = [x_B, 0, z_B]$ *(57)*

where $y_B = 0$.

Eulurian Angles

- $Eulurian = [\phi, \theta, \psi]$ *(58)*

where $\psi = 0, \phi = 0$.

The dynamic equations are calculated for the body-fixed frame of reference and then transformed into the inertial frame of reference. There are certain parameters that are expressed in the inertial frame of reference (i.e. weight and buoyancy) that must be transformed. The following transformation matrices are used for transforming between the inertial and body-fixed frame of reference:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{59}$$

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{60}$$

$$R_z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{61}$$

$$C_{ab} = R_z \cdot R_y * R_x \tag{62}$$

$$C_{ba} = (C_{ab})' \tag{63}$$

here R_x, R_y, and R_z represent the *x*-, *y*-, and *z*-axis components of the transformation matrix, C_{ab} represents the inertial to body-fixed transformation matrix and C_{ba} represents the body-fixed to inertial transformation matrix.

Due to the continually-moving center of gravity, the mass moments of inertia will need to be recalculated for each step. The mass moments of inertia are required for calculating the angular momentum of the AUV. The following equations are related to the calculation of the mass moments of inertia:

$$I_{xx} = \frac{mass}{12} \cdot (width^2 + height^2) \tag{64}$$

$$I_{yy} = \frac{mass}{12} \cdot (length^2 + height^2) \tag{65}$$

$$I_{zz} = \frac{mass}{12} \cdot (length^2 + width^2) \tag{66}$$

$$I_x = I_{xx} + mass \cdot \left({y_G}^2 + {z_G}^2\right) \tag{67}$$

$$I_y = I_{yy} + mass \cdot ({x_G}^2 + {z_G}^2) \tag{68}$$

$$I_z = I_{zz} + mass \cdot \left({x_G}^2 + {y_G}^2\right) \tag{69}$$

$$J_b = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{70}$$

where *width*, *height*, and *length* are the corresponding width, height, and length of the AUV. The parameters I_{xx}, I_{yy}, and I_{zz} represent the *xx*, *yy*, and *zz* components of the mass moments of interia and J_b represents the mass moments of inertia with respect to point A, as shown in Figure 49.

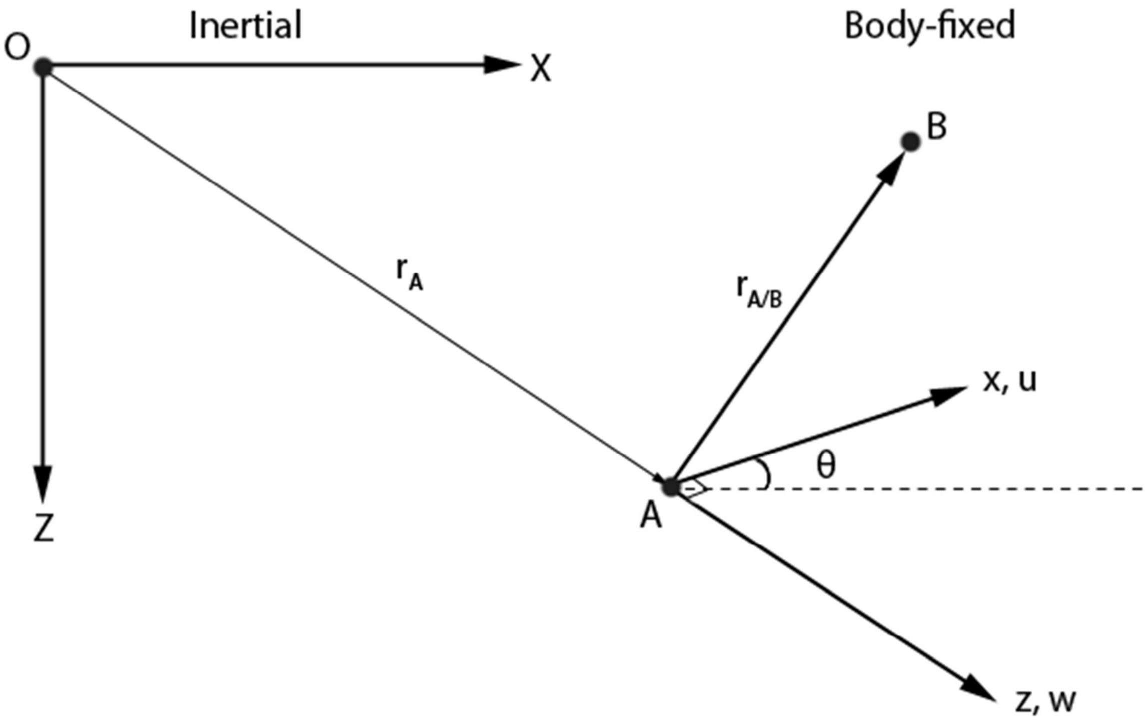

Figure 49 – Reference Axes

The forces and moment due to forces are the backbones required for calculating the linear and angular momentum. The only known forces are those of the ballast tank weight and buoyancy. The following are the equations related to calculating both the forces and moment due to these forces:

$$f_b = C_{ba} \cdot (W_Z + B_Z) \tag{71}$$

$$g_{ob} = \left(\left(r_B \times \left(C_{ba} \cdot B + F_{hyd}\right)\right) + \left(r_G \times (C_{ba} \cdot W)\right) + M_{hyd}\right) \tag{72}$$

where f_b represents the weight and buoyancy forces, and g_{ob} respresents the moment due to forces. The hydrodynamic forces and moments of the AUV are specific for a given profile of the AUV. They can be found in detail in 0 and are grouped as follows:

- $F_{hyd} = \left[F_{hyd,x}, F_{hyd,y}, F_{hyd,z}\right]$
- $M_{hyd,z} = \left[M_{hyd,x}, M_{hyd,y}, M_{hyd,z}\right]$
- $M_{hyd,z} = \left[M_{hyd,x}, M_{hyd,y}, M_{hyd,z}\right]$.

The goal of the system dynamics portion of the simulation is to calculate the current acceleration, velocity, and position of the AUV. The two major concepts required for calculating these parameters are the linear and angular momentum of the system. From the linear and angular momentum outputs from the system, the velocity and position are determined. Equations related to calculating both the linear momentum and angular momentum were presented by Hughes [22]:

$$c = mass \cdot r_G \tag{73}$$

$$\dot{v}_o = \left(\frac{1}{mass}\right)\left(f_b + F_{hyd} - mass \cdot (\omega \times v_o) - \left(\omega \times (\omega \times c)\right) - (\dot{\omega} \times c)\right) \tag{74}$$

$$\begin{aligned}\dot{\omega} = J_b{}^{-1} \cdot \Big(g_{ob} - \left(v_o \times (\omega \times c)\right) - \left(\omega \times (\mathrm{c} \times v_o)\right) - \left(\omega \times (J_b \cdot \omega)\right) \\ - (\mathrm{c} \times \dot{v}_o)\Big)\end{aligned} \tag{75}$$

where c represents the 1st mass moment of inertia with respect to point A, $\dot{v}_o$ represents acceleration, and $\dot{\omega}$ represents angular momentum. The following are the full equations of motion:

$$\begin{aligned}mass \cdot [u_{dot} + w \cdot q - x_G \cdot q^2 + z_G \cdot q_{dot}] \\ = F_{hyd,x} - (W_Z + B_Z) \cdot sin(\theta) + F_{prop}\end{aligned} \tag{76}$$

$$mass \cdot [w_{dot} - u \cdot q - z_G \cdot q^2 - x_G \cdot q_{dot}] = F_{hyd,z} + (W_Z + B_Z) \cdot cos(\theta) \qquad \textit{(77)}$$

$$I_y \cdot q_{dot} + mass \cdot [z_G \cdot (u_{dot} + w \cdot q) - x_G \cdot (w_{dot} - u \cdot q)] = M_{hyd,y} - (x_G \cdot W_Z + x_B \cdot B_Z) \cdot \cos(\theta) - (z_G \cdot W_Z + z_B \cdot B_Z) \cdot \sin(\theta). \qquad \textit{(78)}$$

After substituting the hydrodynamics into equations *(76)*, *(77)*, and *(78)*, these equations can be rearranged to solve for $\dot{u}, \dot{v},$ and $\dot{w}$ using MATLAB® or another capable simultaneous equation solver.

Appendix B MATLAB® Code

The following list includes all of the necessary MATLAB® files for this AUV simulator, listed in alphabetical order (and in the same order as will be found later for the full code):

[1] air_mass.m
[2] bal_ctrl_system.m
[3] ballast_tank.m
[4] bow_ste_control.m
[5] dydt_matrix.m
[6] final_calculations.m
[7] final_quant.m
[8] h_water_max.m
[9] hPDsb_control.m
[10] initial_conditions.m
[11] PD_control.m
[12] pre_controller.m
[13] resource_usage_air_aft.m
[14] resource_usage_air_fwd.m
[15] resource_usage_wat_aft.m
[16] resource_usage_wat_fwd.m
[17] sat_val.m
[18] water_height.m
[19] xg_shift_control.m

FULL MATLAB® CODE

All of the essential MATLAB® code required to perform most of the simulations found in Chapter 6, Chapter 7, Chapter 8, Chapter 9, Chapter 10, and Appendix C has been listed in this section. All of the code has been separated alphabetically by the MATLAB® m-file that the code belongs in. The full MATLAB® code has been shortened to better fit the margins of this thesis and to reduce the overall length. However, no essential code has been removed. One way of shortening the code has been to place two or more defines within the same row. All of the m-files belong in the same directory folder. ***Note:*** Performing simulations with the VBS x_G Shifting controller requires installation of the Symbolic Math Toolbox to utilize a symbolic variable necessary for a calculation.

[1] air_mass.m

```
% Air Mass
    m_air_cas_fwd  = y(9);   m_air_cas_aft  = y(10);
    m_air_vent_fwd = y(23);  m_air_vent_aft = y(24);
    m_air_init_fwd = y(25);  m_air_init_aft = y(26);
    m_air_fwd = m_air_cas_fwd + m_air_vent_fwd + m_air_init_fwd;
    m_air_aft = m_air_cas_aft + m_air_vent_aft + m_air_init_aft;
```

[2] bal_ctrl_system.m

```
function bal_ctrl_system()
% AUV WITH VBS CONTROL
%  Shawn Woods // March 30th, 2012

%% Basic System Defines

% System Clears
    clc; close all; clear all;      % clear data and figures
    time1 = cputime;
    evalin('base','clear tout');    % clear tout
    evalin('base','clear yout');    % clear yout

% System Defines
    tmax          = 3000;   % maximum sample time [s]
    sample_time   = 0.1;    % sample time         [s]
    initial_depth = 100;    % initial depth       [m]
    depth_s       = 100;    % setpoint depth      [m]

% Ballast Tank
    w_bal_tank = 0.30;          % ballast tank width     [m]
    h_bal_tank = w_bal_tank;    % height of ballast tank [m]
    h_air_fwd  = w_bal_tank/2;  % initial air height     [m]
    h_air_aft  = w_bal_tank/2;  % initial air height     [m]

% VBS (PD & HPDCB) Controller Coefficients
    Kp = 1;     % proportional
    Kd = 40;    % derivative

% Controllers
    PD_ctrl    = 0;     % PD VBS depth controller
    HPDCB_ctrl = 1;     % HPDCB VBS depth controller
    xgs_ctrl   = 0;     % VBS xG shifting controller
    bow_ctrl   = 0;     % bowplane proportional component
    ste_ctrl   = 0;     % sternplane proportional component

% Realism Options
    losses = 1;     % losses - valves, water pump, and compressed air
    quants = 1;     % sensors - quantized readings
    delays = 1;     % water valve and pump operational delays
    mrates = 1;     % minimum water pump and comp. air flow rates

% Losses
    wv_loss = 5*(losses == 1);  % water valve       [5% / 0%]
    av_loss = 5*(losses == 1);  % air valve         [5% / 0%]
    wp_loss = 5*(losses == 1);  % water pump        [5% / 0%]
    ca_loss = 5*(losses == 1);  % compressed air    [5% / 0%]
```

```
% Quantization
    de_quant = quants == 1;     % depth
    pa_quant = quants == 1;     % pitch angle
    wv_quant = quants == 1;     % water valve
    wp_quant = quants == 1;     % water pump volume flow rate
    ca_quant = quants == 1;     % compressed air mass flow rate

% Quantization Precision
    wvq = 0.0001;   % water valve free-fill/empty water velocity
    wpq = 0.001;    % water pump flow rate
    caq = 0.001;    % compressed air flow rate

% Minimum Rates
    wpfr_min = mrates;   % minimum water pump volume flow rate
    cafr_min = mrates;   % minimum compressed air mass flow rate

% Delays
    wat_val_delay  = delays == 1;                       % water valve
    wat_pump_delay = delays == 1;                       % water pump
    val_time          = 2;                              % water valve delay [s]
    pump_delay        = 5;                              % water pump delay  [s]
    Pump_rate_fwd_max = 0.010;                          % max. water pump rate [m^3/s]
    pump_change       = (Pump_rate_fwd_max/pump_delay); % water pump change rate

% AUV Compression Due to Depth, Temperature
    AUV_compression = 0;

% Vertical Disturbance Forces
    Fdist = 000;                % disturbance force [N]

% Custom Trajectory
    ctrl_tgr     = 0;
    ctrl_tgr_sim = 1;

% HPDCB Controller
    neutral_correction = 1;     % neutral buoyancy correction

% xG Shifting Controller
    xGs       = 0;%-0.09;%-0.0128;        % center of gravity along the x-axis    [m]
    shift_cal = 0;

% Bowplane and Sternplane Control
    Fprop      = 0;             % fwd propulsion (body-fixed, x)    [N]
    kp_bow     = 0.1;           % proportional coefficient          [bowplane]
    kp_ste     = 8;             % proportional coefficient          [sternplane]
    angle_s    = 0*(pi/180);    % setpoint pitch angle              [rad]

% State Based PD Features
    u_out_sat = 1;

%% Empty Variables (unformated, necessary to prevent crashing)

[a,a_bound,a_hw1,a_hw2,a_pcd,A_wat_fwd_val_a,A_wat_fwd_val_a1,A_wat_fwd_val_ctrl,A_wat_fw
d_val_s,A_wat_val_actual,A_wat_val_rad,a_wp,a_wv1,a_wv1a,a_wv2,aAcurrent,aAprev,aBcurrent
,aBprev,abs_de_l,abs_w_l,aCcurrent,aCprev,aDcurrent,aDprev,Air_rate_aft,Air_rate_fwd,air_
vent_aft,air_vent_fwd,Av_ehmx,Av_lmx,Av_opn,AVc,b_bound,bal_adj1,Bz,c_bound,ca_cons,cas_a
ft,cas_fwd,CoG_aft_bal,CoG_aft_bal1,CoG_fwd_bal,CoG_fwd_bal1,control_status,control_statu
s_aft,d_bound,dB_AUV,db_chg_a,db_chg_l,db_deg,db_deg_s,db_deg_sq,db_degq,db_h_dbs,db_h_ds
s,db_h_mn,db_l_dbs,db_l_dss,db_l_mx,db_l_mx,deltaH,diff_ha,dP,ds_chg_a,ds_deg,ds_deg_s,ds
_deg_sq,ds_degq,ds_h_dss,ds_h_mn,ds_l_dss,ds_l_mx,dV_AUV,dz_high,e_bound,e_cap_e,e_cap_f,
eqq,error_d,error_rate,euu,eww,f_bound,f_hw1,f_hw2,f_pcd,f_pcda,f_wp,f_wv1,f_wv1a,f_wv2,f
Acurrent,fAprev,fBcurrent,fBprev,fCcurrent,fCprev,fDcurrent,fDprev,fe_check,ff_check,fina
l_cal,g_bound,G_z_eh,G_z_hmn,h_aft_d,h_air_aft1,h_air_fwd1,h_air_min_aft,h_air_min_aft1,h
_air_min_fwd,h_air_min_fwd1,h_bound,h_fwd_d,h_to_shift,h_wat_aft_init,h_wat_aft1,h_wat_fw
d_init,h_wat_fwd_save,h_wat_fwd1,h_wat_max_aft1,h_wat_max_fwd1,Hactual,hbal_l,hbal_la,Hch
eck,Hcurrent,Hprev,hwat_h,hwat_ha,hwat_l,hwat_la,Iy,Iyy,k_AUV,M,m_air_aft1,m_air_cas_aft,
m_air_cas_fwd,m_air_fwd_save,m_air_fwd1,m_air_vent_aft,m_air_vent_fwd,m_ref1,m_total,m_to
tal1,m_wat_aft,m_wat_aft1,m_wat_fwd,m_wat_fwd1,mair_m,mair_ma,mass,mass1,max_deg,max_loss
,mx_total,mx_total1,mz_total,mz_total1,neutral,not_neutral,p_comp_diff_aft,p_comp_diff_fw
d,p_diff_aft,p_diff_fwd,P_vent_aft,P_vent_fwd,Pa_elmx,Pa_elmxa,Pdiff_h,Pdiff_ha,Pdiff_l,P
diff_la,pr_cons,Pump_rate_aft,Pump_rate_fwd,Pump_rate_fwd_a,Pump_rate_fwd_s,Pv_low,rG,rG_
ref_i,rG_ref1,rG1,s_deg_s,save1,setpoint_depth,stoppage_reason,time,t_leave,u_cap_e,u_cap
_f,u_d_h,u_d_l,u_out1,ucp_ehu,ucp_elu,ucp_h,ucp_l,uout_h,uout_l,V_AUV_ref,v_cap,val_rad_a
ctual,val_rad_actual,val_rad_max,val_vel,vcap_ca,vcap_fe,vcap_ff,vcap_wp,vel_wat_fwd_fe1,
vel_wat_fwd_ff1,vel_wat_pump_aft,vel_wat_pump_aft_s,vel_wat_val_aft,vel_wat_val_fwd,vG_z_
e,vG_z_el,vG_z_h,vG_z_hmx,vG_z_l,vG_z_lmn,vG_z_lmx,vGz_minf,vGz_minr,vmn_ez,vmn_nez,vmx_e
z,vmx_nez,W,w_val_open,wat_pump_loss_shift,wat_val_vel,wps_aft,wps_fwd,Wz,xB,xG_bal_aft,x
G_bal_aft1,xG_bal_fwd,xG_bal_fwd1,xG_shift,z_low,z_ref,zB,zG_aft1,zG_bal_aft,zG_bal_aft1,
zG_bal_fwd,zG_bal_fwd1,zG_fwd1] = deal([]);

%% General System Parameters

% Ballast Tank Logic
    bal_tank = (PD_ctrl == 1) || (HPDCB_ctrl == 1) || (xgs_ctrl   == 1);

% Simulation Parameters
    h_wat_fwd_n   = sym('h_wat_fwd_n');
    final_cal     = 0;      % perform final calculations (0 = no, 1 = yes)
    t_last1       = 0;      % last time
    u_out         = 0;      % controller output
    t_prev        = 0;      % previous time
    t_remain      = 0;      % time remaining
    time_elapsed  = 1;      % elapsed time
    nbc           = 0;      % neutral buoyancy control
    tstart        = 0;      % simulation start time             [s]
    tfinal        = tmax;   % maximum simulation time           [s]
    stoppage      = 0;      % simulation stoppage count
    value         = 1;      % simulation stoppage variable

% Physical Parameters
    g       = 9.81;     % acceleration due to gravity    [m/s^2]
    rho_wat = 1025;     % (rho_wat) density of sea water [kg/m^3]
    R_air   = 286.9;    % individual gas constant of air [J/kg°K]
    T_air   = 293;      % temperature of air             [°K]

% General AUV Parameters
    [length_AUV, height] = deal(10.7, 1.27);    % length and height of AUV [m]
    a1         = length_AUV;    % length of AUV     [m]
    b1         = length_AUV;    % height of AUV     [m]

% Water Valve Open/Close Velocity
    w_val   = 0.10;                 % valve width       [m]
    val_vel = w_val/val_time;       % valve velocity    [m/s]

% Control Parameters
    wat_val_vel         = 0;        % water valve velocity
    [P_comp, D_comp] = deal(0);     % proportional & derivative component
    X                = 0;                       % inertial X position               [m]
    vG_z             = 0;                       % initial b-f z velocity            [m/s]
    theta            = 0;                       % initial pitch angle               [rad]
    depth            = initial_depth;           % initial depth (redundant)         [m]
    Z                = depth;                   % initial depth (redundant?)        [m]
    initial_position = [X; 0; Z];               % initial position of the AUV       [m]
    depth_error      = depth_s - initial_depth; % depth error                       [m]
    u_depth          = depth_error;             % depth error (redundant)           [m]
    u_depth_bal      = u_depth;                 % depth error (redundant)           [m]
    angle_p          = 0*(pi/180);              % initial pitch angle               [rad]
    initial_angle    = [0; angle_p; 0];         % Initial AUV Angles                [rad]
    pitch_error      = angle_s - angle_p;       % pitch error                       [rad]
    u_pitch  = angle_s - angle_p;               % pitch error                       [rad]
    db_rad   = +00*(pi/180);                    % bowplane angle                    [rad]
    ds_rad   = +00*(pi/180);                    % sternplane angle                  [rad]
    db_rad_s = sat_val(-u_depth,-25*pi/180,25*pi/180);  % setpoint bowplane angle   [rad]
    ds_rad_s = sat_val(-u_pitch,-25*pi/180,25*pi/180);  % setpoint sternplane angle [rad]
    t_last           = 0;                 % tracker for previous timestep           [s]
    [u,v,w] = deal(0);                    % body-fixed x-,y-, and z-axis velocity [m/s]
    [p,q,r] = deal(0);                    % angular yaw, pitch, and roll velocity [rad/s]
    [bow_fin_vel, ste_fin_vel] = deal(0); % bow- & sternplane angular vel.  [rad/s]
    fin_vel = pi/18;    % fin angular velocity [rad/s]

% HPDCB Parameters
    vd_f  =  0.005;  vd_e = -0.005;  % desired ballast fill & empty velocity    [m/s]
    vd_ff =  0.15;   vd_fr = -0.15;  % desired free-fall/free-rise velocity     [m/s]

% Controller Saturation
    max_fin = 25*(pi/180);          % maximum bow- and sternplane fin angle
    max_WB = rho_wat*g*(2*(w_bal_tank^2)*h_bal_tank);   % maximum weight   [N]
    min_WB = -max_WB;                                   % maximum buoyancy [N]

% Water Height
    [h_wat_fwd_pump, h_wat_aft_pump] = deal(0);   % water pump height    [m]
    [h_wat_fwd_ff, h_wat_aft_ff]     = deal(0);   % free-fill height     [m]
    [h_wat_fwd_fe, h_wat_aft_fe]     = deal(0);   % free-empty height    [m]

% Water Flow Rate
    Pump_rate_fwd_min  = 0;
    Pump_rate_aft_max = Pump_rate_fwd_max;
    Pump_rate_aft_min = Pump_rate_fwd_min;
    pump_change_rate  = 0; comp_change_rate  = 0;
    vel_wat_fwd = 0;                                    % initial water vel.      [m/s]
    vel_wat_pump_fwd_s = 0;  vel_wat_pump_fwd_prev = 0; % setpoint pump flow rate [m^3/s]
    [vel_wat_pump_fwd, vel_wat_pump_aft] = deal(0);   % water pump flow rate    [m/s]
    [vel_wat_fwd_ff, vel_wat_aft_ff]     = deal(0);   % free-fill flow rate     [m/s]
    [vel_wat_fwd_fe, vel_wat_aft_fe]     = deal(0);   % free-empty flow rate    [m/s]

% Air Mass Flow Rate
    air_vent_fwd        = -0.1;     % forward air venting mass flow rate
    ARr = round(10*depth_s/5000)/100;
    Air_rate_fwd_max = ARr*(ARr >= 0.01)*(ARr <= 0.10)...
    + 0.01*(ARr < 0.01) + 0.10*(ARr > 0.10);
    Air_rate_fwd_min   = 0;
    Air_rate_aft_max  = Air_rate_fwd_max;
    Air_rate_aft_min  = Air_rate_fwd_min;
    [m_air_cas_fwd_flow, m_air_cas_aft_flow]   = deal(0);   % comp. air flow rate  [kg/s]
    [m_air_vent_fwd_flow, m_air_vent_aft_flow] = deal(0);   % air vent flow rate   [kg/s]

% Water and Air Valves
    [A_wat_fwd_val_min, A_wat_fwd_val_max] = deal(0.001,0.01); % min/max valve area [m^2]
    [A_wat_aft_val_min, A_wat_aft_val_max] = deal(0.001,0.01); % min/max valve area [m^2]
    A_wat_fwd_val     = 0;                                     % initial valve area [m^2]
    pd_con            = 334888890;                             % Pdiff const.  [Pa*s/m^3]
```

```
% Initial Water Parameters
    h_fwd_bal   = h_bal_tank;                     % height of forward ballast tank    [m]
    h_wat_fwd   = h_fwd_bal - h_air_fwd;          % height of water                   [m]
    xGB_fwd     = +4;                             % xCoG/B of ballast tank            [m]
    w_fwd_bal   = w_bal_tank;                     % ballast tank width                [m]

% Water Pump Parameters
    Pump_rate_max     = Pump_rate_fwd_max;        % maximum water pump flow rate
    pump_wat_fwd      = 0;                        % initial water pump state (0 = off)
    pump_wat_aft      = 0;

% Initial Air Parameters
    av_fwd = -0.1;                                      % air venting flow rate [kg/s]
    P_air_fwd = 101325;                                 % pressure of air       [Pa]
    P_wat_fwd = rho_wat*g*initial_position(3);          % pressure of water     [Pa]
    V_air_fwd = h_air_fwd * w_fwd_bal^2;                % volume of air         [m^3]
    m_air_fwd = (P_air_fwd*V_air_fwd) / (R_air*T_air);  % mass of air           [kg]
    m_air_init_fwd = m_air_fwd;                         % initial air mass      [kg]

% Compressed Air Parameters
    [P_air_fwd_max, P_air_aft_max] = deal(100000000);  % maximum pressure threshold  [Pa]
    P_comp_fwd       = 100000000;                      % fwd air compressor pressure [Pa]
    m_comp_fwd       = 0;                              % mass of air added           [kg]
    Air_rate_max     = Air_rate_fwd_max;        % maximum air mass flow rate       [kg/s]
    mass_vent_fwd    = 0.00;                    % forward vent-to mass             [kg]
    comp_air_fwd     = 0;                       % initial compressed air state     (off)
    P_vent_to        = 100000;                  % desired vent-to pressure         [Pa]
    m_flow_comp_air  = 0;                       % compressed air mass flow rate    [kg/s]

% Saturate Forward Water Height
    V_air_fwd_min = m_air_fwd*R_air*T_air/P_air_fwd_max;    % mass of air       [kg]
    h_air_fwd_min = V_air_fwd_min/(w_fwd_bal^2);            % min air height    [m]
    h_wat_max_fwd = h_bal_tank - h_air_fwd_min;             % max water height  [m]
    h_wat_max_fwd = sat_val(h_wat_max_fwd,0,h_bal_tank);    % saturate

% Aft Ballast Tank Parameters
    A_wat_aft_val = 0;                          % initial water valve area         [m^2]
    av_aft = -0.1;                              % air venting mass flow rate       [kg/s]
    vel_wat_aft = 0;                            % initial ballast water velocity   [m/s]
    h_aft_bal   = h_bal_tank;                   % height of aft ballast tank       [m]
    h_wat_aft   = h_aft_bal - h_air_aft;        % height of water                  [m]
    xGB_aft     = -4;                           % xCoG/B of ballast tank           [m]
    w_aft_bal   = w_bal_tank;                   % ballast tank width               [m]

% Initial Air Parameters
    P_air_aft = 101325;                                 % pressure of air           [Pa]
    P_wat_aft = rho_wat*g*initial_position(3);          % pressure of water         [Pa]
    V_air_aft = h_air_aft * w_aft_bal^2;                % volume of air             [m^3]
    m_air_aft = (P_air_aft*V_air_aft) / (R_air*T_air);  % mass of air               [kg]
    m_air_init_aft = m_air_aft;                         % initial air mass          [kg]

% Compressed Air Parameters
    P_comp_aft    = 100000000;                  % aft air compressor pressure       [Pa]
    m_comp_aft    = 0;                          % mass of air added                 [kg]
    mass_vent_aft = 0.00;                       % aft vent-to mass                  [kg]
    comp_air_aft  = 0;                          % initial compressed air state      (off)

% Saturate Forward Water Height
    V_air_aft_min = m_air_aft*R_air*T_air/P_air_aft_max;    % mass of air       [kg]
    h_air_aft_min = V_air_aft_min/(w_aft_bal^2);            % min air height    [m]
    h_wat_max_aft = h_bal_tank - h_air_aft_min;             % max water height  [m]
    h_wat_max_aft = sat_val(h_wat_max_aft,0,h_bal_tank);    % saturate

% System Dynamics Parameters

% Reference Buoyancy
    bal_adj = 2*(rho_wat*0.5* h_bal_tank*w_fwd_bal^2);  % ballast adjustment        [kg]
    B_ref   = [0; 0; -8600*g];                  % reference buoyancy force          [N]
    rB_ref  = [0; 0; 0];                        % ref. vector between point O and B [m]
    rB      = rB_ref;                           % vector between point O and B      [m]
    B       = [0;0;B_ref(3)];                   % reference buoyancy of AUV         [N]

% Reference Weight
    m_ref  = [0; 0; 8600];                      % reference  mass                   [kg]
    rG_ref = [xGs; 0; 0.25]; %0.075             % ref. vector between point O and G [m]
    xG     = rG_ref(1);                         % center of gravity (x-axis)        [m]
    zG     = rG_ref(3);                         % center of gravity (z-axis)        [m]

% Bowplane & Sternplane
    max_rad = 25*(pi/180);      % maximum bow & ste angle (+/-)  [rad]

% Dimensionless Stability Derivatives
    % Axial Force
        Xstar  = -0.0032870;        Xww    =  0.0898433;
        Xqq    =  0.0093828;        Xu_dot = -0.0005729;
        Xw = 0; Xwavw = 0; Xq  = 0; Xqavq = 0; Xqd = 0; Xwd = 0; Xwq = 0;

    % Normal Force
        Zw    = -0.0770456;         Zwavw  = -0.1203970;
        Zq     =  0.0006267;        Zqavq  = -0.0021804;
        Zw_dot = -0.01998;          Zq_dot = -0.00005726;
        Zstar = 0; Zww = 0; Zqq = 0;

    % Pitching Moment
        Mw     =  0.0180739;        Mwavw  =  0.0031395;
        Mq     = -0.0086821;        Mqavq  = -0.0015860;
        Mq_dot = -0.001196;
        Mstar = 0; Mww = 0; Mqq = 0; Mrr = 0; Mrp = 0; Mvr = 0; Mvv = 0;

% Theseus Control Derivatives
    Xdbdb   = -0.0035418;       Xdsds   = -0.0064658;
    Zdb     = -0.01605397;      Zds     = -0.01998134;
    Mdb     =  0.0062778;       Mdbavdb =  0.0000000;
    Mds     = -0.008312668;     Mdsavds =  0.0000000;
    Xdrdr = 0; Mdbdb = 0; Mdsds = 0; Mdrdr = 0;

% Simplifications
    RL2 = 0.5*rho_wat*length_AUV^2;
    RL3 = 0.5*rho_wat*length_AUV^3;
    RL4 = 0.5*rho_wat*length_AUV^4;
    RL5 = 0.5*rho_wat*length_AUV^5;

% Axial Force Hydrodynamic Coefficients
    [x1,x2,x3,x4,x5,x6] =...
    deal(RL4*Xqq,RL3*Xu_dot,RL2*Xstar,RL2*Xww,RL2*Xdsds,RL2*Xdbdb);
    % [1] q^2, [2] u_dot, [3] u^2, [4] w^2, [5] u^2*ds^2, [6] u^2*db^2

% Normal Force Hydrodynamic Coefficients
    [z1,z2,z3,z4,z5,z6,z7,z8] =...
    deal(RL4*Zq_dot,RL4*Zqavq,RL3*Zw_dot,RL3*Zq,RL2*Zw,RL2*Zwavw,RL2*Zds,RL2*Zdb);
    % [1] q_dot, [2] q*abs(q), [3] w_dot, [4] u*q, [5] u*w, [6] w*abs(w), [7] u^2*ds,
    % [8] u^2*db
% Pitching Moment Hydrodynamic Coefficients
    [m1,m2,m3,m4,m5,m6,m7,m8,m9] =...
    deal(RL5*Mq_dot,RL5*Mqavq,RL4*Mq,RL3*Mw,RL3*Mwavw,RL3*Mds,RL3*Mdb,RL3*Mdbavdb,...
         RL3*Mdsavds);
    % [1] q_dot, [2] q*abs(q), [3] u*q, [4] u*w, [5] w*abs(w), [6] u^2*ds, [7] u^2*db,
    % [8] u^2*db^2, [9] u^2*ds^2

%% Initial Conditions
    y0 = [X; Z;                            % inertial X & Z (depth) position     [m]
          theta;                           % pitch angle (theta)                 [°]
          0; 0;                            % body-fixed x & z velocity           [m]
          0;                               % body-fixed angular pitch velocity   [°]
          0; 0;                            % fwd & aft water height (water pump) [m]
          0; 0;                            % fwd & aft air mass                  [kg]
          db_rad; ds_rad;                  % bowplane & sternplane angle         [rad]
          0;                               % unused variable
          0; 0;                            % fwd & aft water height (free-fill)  [m]
          0; 0;                            % fwd & aft water height (free-empty) [m]
          h_wat_fwd; h_wat_aft;            % fwd & aft water height (initial)    [m]
          0;                               % water valve diameter (fwd and aft)  [m]
          0;                               % pump change                         [m^3/s]
          0;                               % air compressor change               [kg/s]
          0; 0;                            % fwd & aft air mass (compressed air) [kg]
          m_air_init_fwd; m_air_init_aft]; % initial fwd air mass                [kg]

    dydt((1:16)') = deal(0);    % integration array
    tout          = tstart;     % time array
    yout          = y0.';       % output array
    options       = odeset('Mass',@mass_matrix,'Events',@events,'MaxStep',sample_time);

%% Main Loop
while tstart < tfinal,

% Integrator
    [t,y,te,ye,ie] = ode45(@yp,[tstart tfinal],y0,options);

% Accumulate Output
    [nt,~]  = size(t);         % size of time (t) array
    [yot,~] = size(yout);      % # of timestep elemtnts in integration time output vector
    yt      = yot + nt - 1;    % adjust yot
    initial_conditions();      % new initial conditions

end

%% Final Calculations
    [yt,~] = size(yout);                        % total timestep elements
    final_cal = 0;                              % final calculations [on/off]
    if final_cal == 1, final_calculations();    % run full calculations
    else               final_quant(); end       % run only quantizations
    assignin('base','tout',tout);               % create tout
    assignin('base','yout',yout);               % create tout

%% ODE Function
    function dydt = yp(t,y)

        pre_controller();
        if ctrl_tgr_sim   == 1,         custom_triggers();  end;  % custom simulation
        if PD_ctrl        == 1,         PD_control();             % PD VBS depth
        elseif HPDCB_ctrl == 1,         HPDCB_control();    end;  % HPDCB VBS depth
        if (bow_ctrl || ste_ctrl) == 1, bow_ste_control();  end;  % bow & ste pi&de
        if xgs_ctrl       == 1,         xg_shift_control(); end;  % VBS xG shifting
        if bal_tank       == 1,         ballast_tank();     end;  % ballast tanks
        dydt_matrix();                                            % dynamics & hydro.
        display_data();                                           % display data
    end

%% Mass Matrix Function
    function M = mass_matrix(t,y)

    % Mass Matrix Components
        M      =  eye(26,26);
        %x2 = 0; z3 = 0; z1 = 0; m1 = 0;    % for when there are no hydrodynamics
        M(4,4) =  mass - x2;  M(4,6) =  mass*zG;
        M(5,5) =  mass - z3;  M(5,6) = -mass*xG - z1;
        M(6,4) =  mass*zG;    M(6,5) = -mass*xG;       M(6,6) =  Iy - m1;
    end

%% Events Function
    function [value,isterminal,direction] = events(t,y)

    % Load Current Parameters
        db_rad         = y(11);     ds_rad         = y(12);
        w_val_open     = y(20);     Pump_rate_fwd  = y(21);
        air_mass();                                 % air mass
        water_height();                             % water height
        h_water_max();                              % max water height
        A_wat_fwd_val_a = w_val*w_val_open;     % water valve area

    % Event Value
        value = [h_wat_fwd;               h_wat_aft;
                 h_bal_tank - h_wat_fwd;  h_bal_tank - h_wat_aft;
                 m_air_fwd;               m_air_aft;
                 max_fin - db_rad;        max_fin + db_rad;
                 max_fin - ds_rad;        max_fin + ds_rad;
                 A_wat_fwd_val_a;         A_wat_fwd_val_max - A_wat_fwd_val_a;
                 Pump_rate_fwd;           Pump_rate_fwd_max - Pump_rate_fwd];

    % If Out-of-Bounds
        isterminal = [+1;+1;+1;+1;+1;+1;+1;+1;+1;+1;+1;+1;+1;+1];  % stop the integration
        direction  = [-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1];  % negative direction
    end

end
```

[3] ballast_tank.m

```
%% Ballast Tank

% Quantization
    % AUV Depth
        if de_quant == 1, depth = round(depth); end

    % Water Pump Volume Flow Rate
        if wp_quant == 1, Pump_rate_fwd = round(Pump_rate_fwd/wpq)*wpq; end

    % Compressed Air Mass Flow Rate
        if ca_quant == 1, Air_rate_fwd = round(Air_rate_fwd/caq)*caq; end

%% Forward Ballast Tank
```

```
% Maximum Water Heights
    h_water_max();

% Water Height Conditions
    f_hw1  = h_wat_fwd >= 0.005;                    % water height low
    f_hw2  = h_wat_fwd <= (h_wat_max_fwd - 0.005);  % water height high

% Filling and Emptying Checks
    ff_check = (control_status == 1);   % free-filling
    fe_check = (control_status == 4);   % free-emptying
    f_wp     = (control_status == 2);   % water pump

% Water Valve Delay 1
    if wat_val_delay == 1,
        if final_cal == 1,
            w_val_open = yout(a,20);
        else
            w_val_open = y(20);                 % width of water valve that is opened
        end
        A_wat_fwd_val_a = w_val*w_val_open;     % actual water valve area
        A_wat_fwd_val_s = A_wat_fwd_val;        % setpoint water valve area
        A_wat_fwd_val   = A_wat_fwd_val_s;      % actual water valve area
        if (A_wat_fwd_val < 0.001) &&...        % soft min. valve open area sat.n for ff
           (ff_check == 0) && (fe_check == 0),
            A_wat_fwd_val = 0;
        end
        if (A_wat_fwd_val < 0.001) &&...        % soft minimum valve open area saturation
           (fe_check == 1) || (ff_check == 1),
            A_wat_fwd_val = 0.001;
        end
        if A_wat_fwd_val > 0.009,               % soft maximum valve open area saturation
            A_wat_fwd_val = 0.009;
        end
    end

% Water Pump Delay
    if wat_pump_delay == 1,
        Pump_rate_fwd_s = Pump_rate_fwd*f_wp;   % setpoint pump delay
        if final_cal == 1,
            Pump_rate_fwd_a = yout(a,21);       % actual pump delay
        else
            Pump_rate_fwd_a = y(21);            % actual pump delay
        end
        % Quantization
            Pump_rate_fwd_s = round(Pump_rate_fwd_s*1E3)/1E3;
            Pump_rate_fwd_a = round(Pump_rate_fwd_a*1E3)/1E3;
        % Pump Rate Change
            if Pump_rate_fwd_a < Pump_rate_fwd_s,
                pump_change_rate = +pump_change;
            elseif Pump_rate_fwd_a > Pump_rate_fwd_s,
                pump_change_rate = -pump_change;
            else
                pump_change_rate = 0;
            end
        % Soft Saturation
            if (Pump_rate_fwd_a < 0.001) &&...
               (pump_change_rate < 0),
                Pump_rate_fwd_a = 0;
                pump_change_rate = 0;
            elseif Pump_rate_fwd_a > 0.009 &&...
```

```
               (pump_change_rate > 0),
                Pump_rate_fwd_a = 0.009;
                pump_change_rate = 0;
            end
        Pump_rate_fwd = Pump_rate_fwd_a;
        % Final Saturation
        if wpfr_min == 1,
            if Pump_rate_fwd < 0.001,
                Pump_rate_fwd = 0.001;
            end
        end
    end

% Pressure Differences
    P_air_fwd       = (m_air_fwd*R_air*T_air) /...  % air pressure           [Pa]
                      (h_air_fwd*w_fwd_bal^2);
    P_wat_fwd       = rho_wat*g*depth;              % water pressure         [Pa]
    p_diff_fwd      = P_wat_fwd - P_air_fwd;        % pressure difference   [Pa]
    P_vent_fwd      = P_air_fwd - P_air_fwd_max;    % venting pressure       [Pa]

% Air Valve Conditions
    f_wv1  = A_wat_fwd_val >  0;            % water valve opened
    f_wv2  = A_wat_fwd_val == 0;            % water valve closed
    if wat_val_delay == 1,
        f_wv1 = A_wat_fwd_val_a >= 0.001;   % water valve opened
        f_wv2 = A_wat_fwd_val_a <  0.001;   % water valve closed
    end

% Free-filling & Free-emptying
    vel_wat_val_fwd = f_wv1*(p_diff_fwd / pd_con)*...       % free-fill/empty vel. [m/s]
                      (A_wat_fwd_val / A_wat_fwd_val_max) / w_fwd_bal^2;

    if (vel_wat_val_fwd >= 0),                              % free-filling
        vel_wat_fwd_ff = vel_wat_val_fwd;
        vel_wat_fwd_fe = 0;
    elseif (vel_wat_val_fwd < 0)                            % free-emptying
        vel_wat_fwd_ff = 0;
        vel_wat_fwd_fe = vel_wat_val_fwd;
    end

% Quantization 2
    if wv_quant == 1,
        vel_wat_fwd_ff = round(vel_wat_fwd_ff/wvq)*wvq;
        vel_wat_fwd_fe = round(vel_wat_fwd_fe/wvq)*wvq;
    end

% Water Pump Support
    if control_status == 2,
        vel_wat_pump_fwd = f_wv2*Pump_rate_fwd/(w_fwd_bal)^2;   % water pump vel. [m/s]
    else
        vel_wat_pump_fwd = 0;
    end

% Air Mass Flow Rate[kg/s]
    air_vent_fwd = av_fwd;      % air venting mass flow rate    [kg/s]

% Losses
    vel_wat_fwd_ff   = vel_wat_fwd_ff*(1-(wv_loss/100));    % free-filling loss
    vel_wat_fwd_fe   = vel_wat_fwd_fe*(1-(wv_loss/100));    % free-emptying loss
    vel_wat_pump_fwd = vel_wat_pump_fwd*(1-(wp_loss/100));  % water pump loss
    Air_rate_fwd     = Air_rate_fwd*(1-(ca_loss/100));      % compressed air losses
    air_vent_fwd     = air_vent_fwd*(1-(av_loss/100));      % air valve losses

% Water Velocity
    vel_wat_fwd = vel_wat_fwd_ff + vel_wat_fwd_fe + vel_wat_pump_fwd;

% Air Venting Support
    if control_status == 3,
        m_air_vent_fwd_flow = air_vent_fwd;
    else
        m_air_vent_fwd_flow = 0;
    end

% Compressed Air Support
    if control_status == 5,
        m_air_cas_fwd_flow = Air_rate_fwd;               % compressed air support
    else
        m_air_cas_fwd_flow = 0;
    end

% Water Valve Delay 2
    if wat_val_delay == 1,
        A_wat_fwd_val_s = A_wat_fwd_val;
        if     (A_wat_fwd_val_a < A_wat_fwd_val_s) &&...
               (A_wat_fwd_val_a < 0.009),
            wat_val_vel = +val_vel;                      % water valve velocity (opening)
        elseif (A_wat_fwd_val_a > A_wat_fwd_val_s) &&...
               (A_wat_fwd_val_a > 0.001),
            wat_val_vel = -val_vel;                      % water valve velocity (closing)
        else
            wat_val_vel = 0;                             % water valve vel. (stationary)
        end
    else
        wat_val_vel = 0;
    end

%% Aft Ballast Tank
    if xgs_ctrl == 1,
    % Water Height Conditions
        f_hw1  = h_wat_aft >= 0.005;                     % water height low
        f_hw2  = h_wat_aft <= (h_wat_max_aft - 0.005);  % water height high

    % Filling and Emptying Checks
        ff_check = (control_status_aft == 1);   % free-filling
        fe_check = (control_status_aft == 4);   % free-emptying
        f_wp     = (control_status_aft == 2);   % water pump

    % Pressure Differences
        P_air_aft       = (m_air_aft*R_air*T_air) /...  % air pressure          [Pa]
                          (h_air_aft*w_aft_bal^2);
        P_wat_aft       = rho_wat*g*depth;               % water pressure        [Pa]
        p_diff_aft      = P_wat_aft - P_air_aft;         % pressure difference   [Pa]
        P_vent_aft      = P_air_aft - P_air_aft_max;     % venting pressure      [Pa]

    % Air Valve Conditions
        f_wv1  = A_wat_aft_val >  0;            % water valve opened
        f_wv2  = A_wat_aft_val == 0;            % water valve closed

    % Free-filling & Free-emptying
        vel_wat_val_aft = f_wv1*(p_diff_aft / pd_con)*...    % free-fill/empty vel. [m/s]
                          (A_wat_aft_val / A_wat_aft_val_max) / w_aft_bal^2;
        if (vel_wat_val_aft >= 0),                  % free-filling
            vel_wat_aft_ff = vel_wat_val_aft;
            vel_wat_aft_fe = 0;
        elseif (vel_wat_val_aft < 0)                % free-emptying
            vel_wat_aft_ff = 0;
            vel_wat_aft_fe = vel_wat_val_aft;
        end

    % Water Pump Support
        if control_status_aft == 2,
            vel_wat_pump_aft = f_wv2*Pump_rate_aft/(w_aft_bal)^2; % water pump vel. [m/s]
        else
            vel_wat_pump_aft = 0;
        end

    % Air Mass Flow Rate[kg/s]
        air_vent_aft = av_aft;      % air venting mass flow rate    [kg/s]

    % Water Velocity
        vel_wat_aft = vel_wat_aft_ff + vel_wat_aft_fe + vel_wat_pump_aft;

    % Air Venting Support
        if control_status_aft == 3,
            m_air_vent_aft_flow = air_vent_aft;
        else
            m_air_vent_aft_flow = 0;
        end

    % Compressed Air Support
        if control_status_aft == 5,
            m_air_cas_aft_flow = Air_rate_aft;      % compressed air support
        else
            m_air_cas_aft_flow = 0;
        end

        wat_val_vel = 0;                            % no delay with xgs_ctrl

    else
        vel_wat_aft         = vel_wat_fwd;
        vel_wat_aft_ff      = vel_wat_fwd_ff;
        vel_wat_aft_fe      = vel_wat_fwd_fe;
        vel_wat_pump_aft    = vel_wat_pump_fwd;
        m_air_cas_aft_flow  = m_air_cas_fwd_flow;
        m_air_vent_aft_flow = m_air_vent_fwd_flow;
    end
```

[4] bow_ste_control.m

```
%% Bowplane Controller

% Setpoint Bowplane Angle [rad]
    db_rad_s = sat_val(-(kp_bow*depth_error),-max_rad,+max_rad);

% Quantization
    db_deg_sq = round(db_rad_s*180/pi);         % setpoint bowplane angle [deg]
    db_degq   = round(db_rad  *180/pi);         % current bowplane angle  [deg]

% Conditions
    db_h_mn  = db_degq > -(max_rad*180/pi - 1); % feedback low enough
    db_l_mx  = db_degq < +(max_rad*180/pi - 1); % feebback high enough
    db_h_dbs = db_degq > db_deg_sq;             % feedback higher than setpoint
    db_l_dbs = db_degq < db_deg_sq;             % feedback lower than setpoint

% Bowplane Angular Velocity [rad/s]
    bow_fin_vel = (-fin_vel*db_h_mn*db_h_dbs + fin_vel*db_l_mx*db_l_dbs)*bow_ctrl;

%% Sternplane Controller

% Setpoint Sternplane Angle [rad]
    ds_rad_s = sat_val(-(kp_ste*pitch_error),-max_rad,+max_rad);

% Quantization
    ds_deg_sq = round(ds_rad_s*180/pi);         % setpoint sternplane angle [deg]
    ds_degq   = round(ds_rad  *180/pi);         % current sternplane angle  [deg]

% Conditions
    ds_h_mn  = ds_degq > -(max_rad*180/pi - 1); % feedback low enough
    ds_l_mx  = ds_degq < +(max_rad*180/pi - 1); % feebback high enough
    ds_h_dss = ds_degq > ds_deg_sq;             % feedback higher than setpoint
    ds_l_dss = ds_degq < ds_deg_sq;             % feedback lower than setpoint

% Sternplane Angular Velocity [rad/s]
    ste_fin_vel = (-fin_vel*ds_h_mn*ds_h_dss + fin_vel*ds_l_mx*ds_l_dss)*ste_ctrl;
```

[5]   dydt_matrix.m

```
%% dydt Matrix

    air_mass();     % air mass
    water_height(); % water height components
    h_water_max();  % max water heights

%% Weight & Mass
    xG_bal_fwd = xGB_fwd;
    xG_bal_aft = xGB_aft;
    zG_bal_fwd = -0.5*h_wat_fwd + h_fwd_bal/2;
    zG_bal_aft = -0.5*h_wat_aft + h_aft_bal/2;
    m_wat_fwd  = rho_wat*(h_wat_fwd - 0.5*h_fwd_bal)*w_fwd_bal^2;
    m_wat_aft  = rho_wat*(h_wat_aft - 0.5*h_aft_bal)*w_aft_bal^2;
    mx_total   = m_wat_fwd*xG_bal_fwd + m_wat_aft*xG_bal_aft + m_ref(3)*rG_ref(1);
    mz_total   = m_wat_fwd*xG_bal_fwd + m_wat_aft*xG_bal_aft + m_ref(3)*rG_ref(3);
    mass       = m_wat_fwd + m_wat_aft + m_ref(3);  % mass
    xG = mx_total/mass;  zG = mz_total/mass;        % center of mass components
    xB = rB(1);     zB = rB(3);                     % center of buoyancy components
    Wz = mass*g;                                    % weight

%% Buoyancy & Density Change (due to depth, temperature, and salinity)
    Bz        = B(3);                       % reference buoyancy            [N]
    if AUV_compression == 1,
        depth     = y(2);                   % depth                         [m]
        rho_wat   = 0.003*depth + 1025;     % ocean water density           [kg/m^3]
        dP        = rho_wat*g*depth;        % change in pressure            [Pa]
        k_AUV     = 45.8E-11;               % compressibility of water      [Pa^-1]
        V_AUV_ref = Bz/(rho_wat*g);         % volume of the AUV             [m^3]
        dV_AUV    = -k_AUV*V_AUV_ref*dP;    % volume change due to depth    [m^3]
        dB_AUV    = rho_wat*g*dV_AUV;       % buoyancy change               [N]
        Bz        = Bz + dB_AUV;            % new buoyancy (temporary)      [N]
    end
```

```
%% Body-fixed Mass Moments of Inertia
    Iyy = mass/12*(length_AUV^2 + height^2);    % mass moment of inertial, yy     [m^4]
    Iy  = Iyy + mass*(xG^2 + zG^2);             % mass moment of inertial, y      [m^4]

%% Variable Defines
    X      = y(1);  Z      = y(2);   % inertial X & Z (depth) position  [m]
    theta  = y(3);                   % pitch angle                      [rad]
    u      = y(4);  w      = y(5);   % body-fixed x & z velocity        [m/s]
    q      = y(6);                   % angular pitch velocity           [rad/s]
    db_rad = y(11); ds_rad = y(12);  % bowplane & sternplane angle      [rad]

% Quantizing Depth
    if de_quant == 1, Z = round(Z); end
    depth = Z;

% Quantizing Pitch Angle
    if pa_quant == 1, theta = round(theta*180/pi)*pi/180; end

%% Component Matrix, f(t,y)
    euu =      RL4 * (Xqq*q^2 + Xqavq*q*abs(q))                                      +...
               RL3 * (Xwq*w*q + Xq*u*q)                                              +...
               RL2 * (Xstar*u^2 + Xww*w^2 + Xwavw*w^2 + Xw*u*w                       +...
                      Xdbdb*u^2*db_rad^2 + Xdsds*u^2*ds_rad^2)                       +...
                      Fprop                                                          +...
               - (Wz + Bz + Fdist)*sin(theta) - mass*(w*q - xG*q^2);
    eww =      RL4 * (Zqavq*q*abs(q) + Zqq*q^2)                                      +...
               RL3 * (Zq*u*q)                                                        +...
               RL2 * (Zstar*u^2 + Zw*u*w + Zwavw*w^2                                 +...
                      Zww*w^2 + Zdb*u^2*db_rad + Zds*u^2*ds_rad)                     +...
               (Wz + Bz + Fdist)*cos(theta) + mass*(u*q + zG*q^2);
    eqq =      RL5 * (Mqq*q^2 + Mqavq*q*abs(q))                                      +...
               RL4 * (Mq*u*q)                                                        +...
               RL3 * (Mstar*u^2 + Mw*u*w + Mwavw*w^2 + Mww*w^2                       +...
                      u^2*(Mdb*db_rad + Mdbdb*db_rad^2 + Mdbavdb*db_rad*abs(db_rad)) +...
                      u^2*(Mds*ds_rad + Mdsds*ds_rad^2 + Mdsavds*ds_rad*abs(ds_rad)))+...
            - (xG*Wz + xB*Bz + xB*Fdist)*cos(theta)...
            - (zG*Wz + zB*Bz + zB*Fdist)*sin(theta)...
            - mass*(zG*w*q + xG*u*q);

%% Controller Output
    % System Error
        depth_error = depth_s - Z;          % depth error   [m]
        pitch_error = angle_s - theta;      % pitch error   [rad]
        error_rate = -(-u*sin(theta) + w*cos(theta));   % rate-of-change of error

    % Controller Component Outputs
        P_comp = Kp*(depth_error);
        D_comp = Kd*error_rate;
        u_out  = sat_val(P_comp+D_comp,-1,1);   % controller output and saturation

%% dydt Component
    dydt =[+u*cos(theta) + w*sin(theta);    % 01: X_dot
           -u*sin(theta) + w*cos(theta);    % 02: Z_dot
           q;                               % 03: theta_dot
           euu;                             % 04: u_dot
           eww;                             % 05: w_dot
           eqq;                             % 06: q_dot
           vel_wat_pump_fwd;                % 07: water volume flow rate    (fwd)
           vel_wat_pump_aft;                % 08: water volume flow rate    (aft)
           m_air_cas_fwd_flow;              % 09: comp. air mass flow rate  (fwd)
           m_air_cas_aft_flow;              % 10: comp. air mass flow rate  (aft)
           bow_fin_vel;                     % 11: deflection fin velocity   (bowplane)
           ste_fin_vel;                     % 12: deflection fin velocity   (sternplane)
           0;                               % unused variable
           vel_wat_fwd_ff;                  % 14: water volume flow rate    (fwd)
           vel_wat_aft_ff;                  % 15: water volume flow rate    (aft)
           vel_wat_fwd_fe;                  % 16: water volume flow rate    (fwd)
           vel_wat_aft_fe;                  % 17: water volume flow rate    (aft)
           0;                               % 18: initial water height      (fwd)
           0;                               % 19: initial water height      (aft)
           wat_val_vel;                     % 20: water valve velocity      (fwd & aft)
           pump_change_rate;                % 21: water pump rate change    (fwd & aft)
           comp_change_rate;                % 22: compressed air rate       (fwd & aft)
           m_air_vent_fwd_flow;             % 23: air mass flow rate        (fwd)
           m_air_vent_aft_flow;             % 24: air mass flow rate        (aft)
           0;                               % 25: initial air mass          (fwd)
           0];                              % 26: initial air mass          (aft)

% Outputs
    vG_z = dydt(2);                         % inertial Z velocity           [m/s]
```

[6] final_calculations.m

```
%% Final Calculations

for a = 1 : yt,
    % Assign Values
        pr_cons        = 0; % pump rate constant
        depth          = yout(a,2);   depth_error    = depth_s - depth;
        theta          = yout(a,3);   u              = yout(a,4);
        w              = yout(a,5);
        h_wat_fwd_pump = yout(a,7);   h_wat_aft_pump = yout(a,8);
        db_rad         = yout(a,11);  ds_rad         = yout(a,12);

    % Quantizing Depth
        if de_quant == 1, depth = round(depth); end

    % Quantizing Pitch Angle
        if pa_quant == 1, theta = round(theta*180/pi)*pi/180; end

    h_wat_fwd = h_wat_fwd_pump + h_wat_fwd_ff + h_wat_fwd_fe + h_wat_fwd_init;
    h_wat_aft = h_wat_aft_pump + h_wat_aft_ff + h_wat_aft_fe + h_wat_aft_init;

    A_wat_fwd_val_a = pi*A_wat_val_rad^2;

    h_air_fwd = h_bal_tank - h_wat_fwd;                 % fwd air height [m]
    h_air_aft = h_bal_tank - h_wat_aft;                 % aft air height [m]
    m_air_fwd = yout(a,9)  + yout(a,23) + yout(a,25);   % fwd air mass [kg]
    m_air_aft = yout(a,10) + yout(a,24) + yout(a,26);   % aft air mass [kg]

    P_air_fwd = (m_air_fwd*R_air*T_air) / (h_air_fwd*w_fwd_bal^2); % fwd air press. [Pa]
    P_air_aft = (m_air_aft*R_air*T_air) / (h_air_aft*w_aft_bal^2); % aft air press. [Pa]

    % Max Water Heights
        h_water_max();

    % State-Based Control
        if HPDCB_control == 1,
            HPDCB_control();    % state-based controller
            ballast_tank;       % ballast tank
        end

    % Proportional Control
        if PD_ctrl == 1,
            PD_control();       % proportional controller
            ballast_tank;       % ballast tank
        end

    % Water Pump Delay
        if wat_pump_delay == 1, Pump_rate_fwd = Pump_rate_fwd_a; end

    % Controller Output
        P_comp     = Kp*(depth_error);
        error_rate = -(-u*sin(theta) + w*cos(theta));
        D_comp     = Kd*error_rate;
        u_out      = sat_val(P_comp+D_comp,-1,1);

    % Store to Array
        yout(a,27) = vel_wat_pump_fwd*w_fwd_bal^2;
        yout(a,28) = vel_wat_fwd;    yout(a,29) = vel_wat_aft;
        yout(a,31) = P_comp;         yout(a,33) = D_comp;
        yout(a,30) = u_out;          yout(a,34) = error_rate;
        yout(a,35) = h_wat_max_fwd;  yout(a,36) = h_wat_max_aft;
        yout(a,37) = wat_pump_loss;  yout(a,38) = vG_z;
        yout(a,40) = P_air_fwd;      yout(a,41) = P_air_aft;
        yout(a,39) = Pump_rate_fwd;

end
```

[7] final_quant.m

```
%% Final Quantization
    for a = 1 : yt,
        % Quantizing Depth
            depth = yout(a,2);      % inertial Z position (depth)
            if de_quant == 1, depth = round(depth); end

        % Quantizing Pitch Angle
            theta = yout(a,3);
            if pa_quant == 1, theta = round(theta*180/pi)*pi/180; end
        yout(a,2)  = depth;
        yout(a,3)  = theta;
    end
```

[8] h_water_max.m

```
%% Maximum Water Height

% Calculate Maximum Water Heights [m] (for design safety)
    h_air_fwd = h_bal_tank - h_wat_fwd;                                     % fwd air
    h_air_aft = h_bal_tank - h_wat_aft;                                     % aft air
    h_air_min_fwd = (m_air_fwd*R_air*T_air) / (P_air_fwd_max*w_fwd_bal^2);  % min fwd air
    h_air_min_aft = (m_air_aft*R_air*T_air) / (P_air_aft_max*w_aft_bal^2);  % max fwd air
    h_wat_max_fwd = sat_val(h_bal_tank - h_air_min_fwd,0,h_bal_tank);       % min fwd wat
    h_wat_max_aft = sat_val(h_bal_tank - h_air_min_aft,0,h_bal_tank);       % min aft wat
```

[9] HPDCB_control.m

```
%% State-Based with PD Controller [Depth]

% Default Defines
    nbc            = 0;
    control_status = 0;     % "do nothing" control status
    A_wat_fwd_val  = 0;     % water valve area               [m^2]
    Pump_rate_fwd  = 0;     % forward volume flow rate       [m^3/s]
    Air_rate_fwd   = 0;     % forward mass flow rate         [kg/s]
    pump_wat_fwd   = 0;     % forward pump state             (off)
    comp_air_fwd   = 0;     % forward air compressor state   (off)

% Controller Output
    u_depth_bal = (u_out)*(u_out <= +1)*(u_out >= -1) + (u_out > +1) - (u_out < -1);

% Venting Pressure Difference
    P_vent_fwd = P_air_fwd - P_air_fwd_max;     % forward   [Pa]
    P_vent_aft = P_air_aft - P_air_aft_max;     % aft       [Pa]

% Free-fill / Free-empty Capability
    v_cap = (P_wat_fwd - P_air_fwd)/(pd_con*w_fwd_bal^2);   % capable velocity      [m/s]

% Conditions
    Pa_elmx  = P_air_fwd        <= P_air_fwd_max;           % equal or lower
    Pv_low   = P_vent_fwd       <  0;                       % low
    Pdiff_h  = P_wat_fwd        >  P_air_fwd;               % + pressure difference
    Pdiff_l  = P_wat_fwd        <  P_air_fwd;               % - pressure difference
    f_pcd    = P_air_fwd        <= P_comp_fwd;
    uout_h   = u_out            >  0;                       % high
    uout_l   = u_out            <  0;                       % low
    hwat_l   = h_wat_fwd        <  h_wat_max_fwd - 0.005;   % low
    hwat_h   = h_wat_fwd        >  0.005;                   % high
    hbal_l   = h_wat_fwd        <  h_bal_tank - 0.005;
    vcap_ff  = v_cap            >= vd_f;                    % free-filling capable
    vcap_wp  = v_cap            <  vd_f;                    % water pump usage
    vcap_fe  = v_cap            <= vd_e;                    % free-filling capable
    vcap_ca  = v_cap            >  vd_e;                    % water pump usage
    vGz_minf = vG_z             <= +0.05;                   % minimum fall velocity
    vGz_minr = vG_z             >= -0.05;                   % minimum rise velocity
    abs_de_l = abs(depth_error) <= +2.0;                    % low absolute depth error
    mair_m   = m_air_fwd        >  0.535;
    AVc      = h_wat_max_fwd    <  0.170;                   % air venting control

% Free-filling              (Status = 1)
    if uout_h && vcap_ff && hwat_l,
        A_wat_fwd_val  = A_wat_fwd_val_max*abs(vd_f/v_cap);     % open water valve
        control_status = 1;

% Water Pump Support        (Status = 2)
    elseif uout_h && Pa_elmx && vcap_wp && hwat_l,
        pr_cons        = abs(u_depth_bal);               % water pump rate constant
        Pump_rate_fwd  = Pump_rate_fwd_max*pr_cons ...
                       + Pump_rate_fwd_min*(1-pr_cons);  % [m^3/s]
        control_status = 2;

% Air Venting Support       (Status = 3)
    elseif uout_h && mair_m && hbal_l && AVc,
        control_status = 3;
    end
```

```
% Free-emptying             (Status = 4)
    if uout_l && vcap_fe && hwat_h,
        A_wat_fwd_val = A_wat_fwd_val_max*abs(vd_e/v_cap);  % open water valve
        control_status = 4;

% Compressed Air Support    (Status = 5)
    elseif uout_l && vcap_ca && Pa_elmx && f_pcd && hwat_h,
        ca_cons       = abs(u_depth_bal);               % compressed air constant
        Air_rate_fwd  = Air_rate_fwd_max*ca_cons...
                      + Air_rate_fwd_min*(1-ca_cons);   % [kg/s]
        control_status = 5;
    end

%% Make Neutrally Buoyant
    not_neutral = abs(h_wat_fwd - 0.15) >= 0.001;
    neutral     = (not_neutral == 0);

    if abs_de_l && (neutral_correction == 1),
        if (vGz_minf && vGz_minr) && (not_neutral),
            nbc = 1;                                      % neutral buoyancy correction
            Pump_rate_fwd = Pump_rate_fwd_max/10;         % pump vol. flow rate   [m^3/s]
            Air_rate_fwd  = Air_rate_fwd_max/10;          % comp. mass flow rate  [kg/s]
            pump_wat_fwd  = (h_wat_fwd - 0.15) < +0.001; % water pump state
            comp_air_fwd  = (h_wat_fwd - 0.15) > -0.001; % air compressor state
            if pump_wat_fwd == 1,
                control_status = 2;
            elseif comp_air_fwd == 1,
                control_status = 5;
            end
            if v_cap < 0,
                A_wat_fwd_val = A_wat_fwd_val_max*comp_air_fwd; % fwd valve area [m^2]
            else
                A_wat_fwd_val = 0;
            end
        end
    end

%% Dead Zone
    if abs_de_l && neutral,
        A_wat_fwd_val = 0;
        Pump_rate_fwd = 0; pump_wat_fwd  = 0;
        Air_rate_fwd  = 0; comp_air_fwd  = 0;
        control_status = 0;
    end

%% Water Valve Saturation
% Conditions
    Av_lmx   = A_wat_fwd_val <  A_wat_fwd_val_max;             % lower than max
    Av_opn   = A_wat_fwd_val >  0;                             % valve open
    Av_ehmx  = A_wat_fwd_val >= A_wat_fwd_val_max;             % equal or higher than max

% Saturation
    A_wat_fwd_val = A_wat_fwd_val    *(Av_opn && Av_lmx) +...  % valid open size
                    A_wat_fwd_val_max*(Av_ehmx);               % opened max or further

%% Aft Ballast Tank Values
    % since this controller does not control pitch, ballast tanks are operated equally
    % this could be adjusted to compensate where AUV starts xG shifted
    A_wat_aft_val = A_wat_fwd_val;
    Pump_rate_aft = Pump_rate_fwd;
    pump_wat_aft  = pump_wat_fwd;
    Air_rate_aft  = Air_rate_fwd;
    comp_air_aft  = comp_air_fwd;
```

[10] initial_conditions.m

```
% Assign Current Integrator Values
    X = y(nt,1); Z = y(nt,2); theta = y(nt,3);
    u = y(nt,4); w = y(nt,5); q = y(nt,6);
    h_wat_fwd_pump = y(nt,7); h_wat_aft_pump = y(nt,8);
    m_air_cas_fwd  = y(nt,9); m_air_cas_aft  = y(nt,10);
    db_rad = y(nt,11); ds_rad = y(nt,12);
    h_wat_fwd_ff   = y(nt,14); h_wat_aft_ff   = y(nt,15);
    h_wat_fwd_fe   = y(nt,16); h_wat_aft_fe   = y(nt,17);
    h_wat_fwd_init = y(nt,18); h_wat_aft_init = y(nt,19);
    w_val_open     = y(nt,20); Pump_rate_fwd  = y(nt,21);
    m_air_vent_fwd = y(nt,23); m_air_vent_aft = y(nt,24);
    m_air_init_fwd = y(nt,25); m_air_init_aft = y(nt,26);

% Air Mass
    m_air_fwd = m_air_cas_fwd + m_air_vent_fwd + m_air_init_fwd;
    m_air_aft = m_air_cas_aft + m_air_vent_aft + m_air_init_aft;

% Water Heights
    h_wat_fwd = h_wat_fwd_pump + h_wat_fwd_ff + h_wat_fwd_fe + h_wat_fwd_init;
    h_wat_aft = h_wat_aft_pump + h_wat_aft_ff + h_wat_aft_fe + h_wat_aft_init;

% Max Water Heights
    h_water_max();

% Water Valve Area
    A_wat_fwd_val_a = w_val*w_val_open;     % water valve area

% Bounds Crossed
    a_bound = -(h_wat_fwd <= 0) + (h_wat_fwd >= h_bal_tank);      % fwd water height bnds
    b_bound = -(h_wat_aft <= 0) + (h_wat_aft >= h_bal_tank);      % aft water height bnds
    c_bound = -(db_rad < -25*(pi/180)) + (db_rad > +25*(pi/180)); % bowplane bounds
    d_bound = -(ds_rad < -25*(pi/180)) + (ds_rad > +25*(pi/180)); % sternplane bounds
    e_bound =  (m_air_fwd <= 0);                                  % fwd air mass bound
    f_bound =  (m_air_aft <= 0);                                  % aft air mass bound
    g_bound = -(A_wat_fwd_val_a < 0) +...                         % water valve vel. bnds
               (A_wat_fwd_val_a > A_wat_fwd_val_max);
    h_bound = -(Pump_rate_fwd < 0) +...                           % water pump rate bnds
               (Pump_rate_fwd > Pump_rate_fwd_max);

%% Set Initial Conditions

% Forward Air Mass
    m_air_fwd = (m_air_fwd)*(e_bound == 0) + (0.002)*(e_bound == 1);

% Aft Air Mass
    m_air_aft = (m_air_aft)*(f_bound == 0) + (0.002)*(f_bound == 1);

% Max Water Heights
    h_water_max();  % used again to update h_wat_max in case air mass is changed

% Forward Water Height
    h_wat_fwd = (h_wat_fwd)*(a_bound == 0) +...
                (h_bal_tank - 0.005)*(a_bound == 1) +...
                (0.005)*(a_bound == -1);
```

```
% Aft Water Height
    h_wat_aft = (h_wat_aft)*(b_bound == 0) +...
                (h_bal_tank - 0.005)*(b_bound == 1) +...
                (0.005)*(b_bound == -1);

% Height Soft Saturation
    if h_wat_fwd <= 0, h_wat_fwd = 0.005; end
    if h_wat_aft <= 0, h_wat_aft = 0.005; end

% Water Valve Radius
    A_wat_fwd_val_a = (0.001)*(g_bound == -1) +...
                      (A_wat_fwd_val_a)*(g_bound == 0);
    w_val_open      = A_wat_fwd_val_a/w_val;

% Water Pump Rate
    Pump_rate_fwd = (0.001)*(h_bound == -1) +...
                    (Pump_rate_fwd)*(h_bound == 0) +...
                    (0.009)*(h_bound == 1);
    y(nt,21) = Pump_rate_fwd;    % overwrite previous entry

% Bowplane Fin Deflection (max represents positive and negative)
    db_rad   = max_rad*c_bound +...
               db_rad*(c_bound == 0);
    y(nt,11) = db_rad;          % overwrite previous entry

% Sternplane Fin Deflection
% max represents positive and negative
    ds_rad   = max_rad*d_bound +...
               ds_rad*(d_bound == 0);
    y(nt,12) = ds_rad;          % overwrite previous entry

% Determine Initial Values
    fAcurrent = h_wat_fwd_pump;  aAcurrent = h_wat_aft_pump;
    fBcurrent = h_wat_fwd_ff;    aBcurrent = h_wat_aft_ff;
    fCcurrent = h_wat_fwd_fe;    aCcurrent = h_wat_aft_fe;
    fDcurrent = h_wat_fwd_init;  aDcurrent = h_wat_aft_init;

    if (a_bound) == 1, resource_usage_wat_fwd(); end; % fwd water
    if (b_bound) == 1, resource_usage_wat_aft(); end; % aft water
    if (e_bound) == 1, resource_usage_air_fwd(); end; % fwd air
    if (f_bound) == 1, resource_usage_air_aft(); end; % aft air

% Create New Initial y
    y0 = [X;Z;theta;
          u;w;q;
          fAcurrent; aAcurrent;
          m_air_cas_fwd; m_air_cas_aft;
          db_rad; ds_rad; 0;
          fBcurrent; aBcurrent; fCcurrent; aCcurrent; fDcurrent; aDcurrent;
          w_val_open;
          Pump_rate_fwd; 0;
          m_air_vent_fwd; m_air_vent_aft; m_air_init_fwd; m_air_init_aft];

% Number of Stoppages
    stoppage = stoppage + 1;

% Set New Initial and Max Timesteps
    options = odeset(options);
    tstart  = t(nt);                            % new start time
```

```
% Accumulate Output
    tout       = [tout; t(2:nt)];               % timestep array
    yout       = [yout; y(2:nt,:)];             % integration output vector
```

[11] PD_Control.m

```
%% PD Controller

% Controller Output
    u_depth_bal = sat_val(u_out,-1,1);

% Default Defines
    control_status = 0;     % "do nothing" control status
    A_wat_fwd_val  = 0;     % water valve area              [m^2]
    Pump_rate_fwd  = 0;     % forward volume flow rate      [m^3/s]
    Air_rate_fwd   = 0;     % forward mass flow rate        [kg/s]
    pump_wat_fwd   = 0;     % forward pump state            (off)
    comp_air_fwd   = 0;     % forward air compressor state  (off)

% Conditions
    Pa_elmx  = P_air_fwd <= P_air_fwd_max;              % equal or lower
    f_pcd    = P_air_fwd <= P_comp_fwd;
    Pdiff_h  = P_wat_fwd >  P_air_fwd;                  % positive pressure difference
    Pdiff_l  = P_wat_fwd <  P_air_fwd;                  % negative pressure difference
    uout_h   = u_out     >  0;                          % high
    uout_l   = u_out     <  0;                          % low
    hwat_l   = h_wat_fwd <  h_wat_max_fwd - 0.005;      % low
    hbal_l   = h_wat_fwd <  h_bal_tank - 0.005;
    hwat_h   = h_wat_fwd >  0.005;                      % high
    mair_m   = m_air_fwd >  0.535;

% Water Pump Support        (Status = 2)
    if uout_h && Pa_elmx && hwat_l,
        pr_cons        = abs(u_depth_bal);               % water pump rate constant
        Pump_rate_fwd  = Pump_rate_fwd_max*pr_cons ...
                       + Pump_rate_fwd_min*(1-pr_cons);  % [m^3/s]
        control_status = 2;
% Air Venting Support       (Status = 3)
    elseif uout_h && mair_m && hbal_l;
        control_status = 3;
    end

% Compressed Air Support    (Status = 5)
    if uout_l && Pa_elmx && f_pcd && hwat_h,
        ca_cons       = abs(u_depth_bal);               % compressed air constant
        Air_rate_fwd  = Air_rate_fwd_max*ca_cons...
                      + Air_rate_fwd_min*(1-ca_cons);   % [kg/s]
        control_status = 5;
% Free-emptying;            (Status = 4)
        if uout_l && Pdiff_l && hwat_h,
            A_wat_fwd_val = A_wat_fwd_val_max;
        end
    elseif uout_l && Pdiff_l && hwat_h,
        A_wat_fwd_val = A_wat_fwd_val_max;
        control_status = 4;
    end

%% Aft Ballast Tank Parameters
    Pump_rate_aft = Pump_rate_fwd;
    Air_rate_aft  = Air_rate_fwd;
    pump_wat_aft  = pump_wat_fwd;
    comp_air_aft  = comp_air_fwd;
    A_wat_aft_val = A_wat_fwd_val;
```

[12] pre_controller.m

```
%% Pre-Controller

% Pump and Air Rates
    if xgs_ctrl == 0,
        Pump_rate_fwd_min = 0;  % water pump maximum volume flow rate   [m^3/s]
        Air_rate_fwd_min  = 0;  % compressed air maximum mass flow rate [kg/s]
        Pump_rate_fwd     = 0;  % forward volume flow rate              [m^3/s]
        Air_rate_fwd      = 0;  % forward mass flow rate                [kg/s]
        pump_wat_fwd      = 0;  % forward pump state                    (off)
        comp_air_fwd      = 0;  % forward air compressor state          (off)
    end

    if wpfr_min == 1, Pump_rate_fwd_min = 0.001; end
    if cafr_min == 1, Air_rate_fwd_min  = 0.001; end

% Quantization

    if de_quant == 1, depth = round(depth); end

% Water Valve
    if xgs_ctrl == 0,
        A_wat_fwd_val = 0;      % default forward water valve [m]
    end
```

[13] resource_usage_air_aft.m

```
%% Air Mass Stoppage Correction

% Notes
    % will crash if stopped on the 1st timestep (which should never occur)
    % currently only valid for symmetrical systems (xG shifting is NOT symmetrical)

    Aprev    = y(nt-1,10);          % previous fwd air mass cas portion
    Bprev    = y(nt-1,24);          % previous fwd air mass venting portion
    Hactual  = 0.002;
    Hprev    = Aprev + Bprev + m_air_init_aft;
    deltaH   = Hactual - Hprev;
    if m_air_aft <= 0,
        m_air_vent_aft = Bprev + deltaH;
        y(nt,24) = m_air_vent_aft;
    end
```

[14] resource_usage_air_fwd.m

```
%% Air Mass Stoppage Correction

% Notes
    % will crash if stopped on the 1st timestep (which should never occur)
    % currently only valid for symmetrical systems (xG shifting is NOT symmetrical)

    Aprev    = y(nt-1,9);               % previous fwd air mass cas portion
    Bprev    = y(nt-1,23);              % previous fwd air mass venting portion
    Hactual  = 0.002;
    Hprev    = Aprev + Bprev + m_air_init_fwd;
    deltaH   = Hactual - Hprev;
    if m_air_fwd <= 0,
        m_air_vent_fwd = Bprev + deltaH;
        y(nt,23) = m_air_vent_fwd;
    end
```

[15] resource_usage_wat_aft.m

```
% Aft Ballast Tank

    aAprev    = y(nt-1,8);      % previous water pump height portion
    aBprev    = y(nt-1,15);     % previous free-filling height portion
    aCprev    = y(nt-1,17);     % previous free-emptying height portion
    aDprev    = y(nt-1,19);     % previous initial height portion
    aAcurrent = y(nt,8);        % current water pump height portion
    aBcurrent = y(nt,15);       % current free-filling height portion
    aCcurrent = y(nt,17);       % current free-emptying height portion
    aDcurrent = y(nt,19);        % current initial height portion

    stoppage_reason = 1*(aAprev ~= aAcurrent) +...
                      2*(aBprev ~= aBcurrent) +...
                      3*(aCprev ~= aCcurrent);

    if (stoppage_reason == 1) || (stoppage_reason == 2),
        Hactual = h_bal_tank - 0.005;
    elseif stoppage_reason == 3,
        Hactual = 0.005;
    else                                % due to air mass increase
        Hactual = h_wat_aft;
    end

    Hprev    = aAprev + aBprev + aCprev + aDprev;
    deltaH   = Hactual - Hprev;

    if stoppage_reason == 1,            % water pump stoppage
        aAcurrent = aAprev + deltaH;
        y(nt,8)  = aAcurrent;

    elseif stoppage_reason == 2,        % free-filling stoppage
        aBcurrent = aBprev + deltaH;
        y(nt,15) = aBcurrent;

    elseif stoppage_reason == 3,        % free-emptying stoppage
        aCcurrent = aCprev + deltaH;
        y(nt,17) = aCcurrent;
    else                                % due to air mass increase
        aCcurrent = aCprev + deltaH;
        y(nt,17) = aCcurrent;
    end
```

[16] resource_usage_wat_fwd.m

```
% Forward Ballast Tank

    fAprev    = y(nt-1,7);      % previous water pump height portion
    fBprev    = y(nt-1,14);     % previous free-filling height portion
    fCprev    = y(nt-1,16);     % previous free-emptying height portion
    fDprev    = y(nt-1,18);     % previous initial height portion
    fAcurrent = y(nt,7);        % current water pump height portion
    fBcurrent = y(nt,14);       % current free-filling height portion
    fCcurrent = y(nt,16);       % current free-emptying height portion
    fDcurrent = y(nt,18);       % current initial height portion

    stoppage_reason = 1*(fAprev ~= fAcurrent) +...
                      2*(fBprev ~= fBcurrent) +...
                      3*(fCprev ~= fCcurrent);
    if (stoppage_reason == 1) || (stoppage_reason == 2),
        Hactual = h_bal_tank - 0.005;
    elseif stoppage_reason == 3,
        Hactual = 0.005;
    else                                % due to air mass increase
        Hactual = h_wat_fwd;
    end

    Hprev    = fAprev + fBprev + fCprev + fDprev;
    deltaH   = Hactual - Hprev;

    if stoppage_reason == 1,            % water pump stoppage
        fAcurrent = fAprev + deltaH;
        y(nt,7)  = fAcurrent;

    elseif stoppage_reason == 2,        % free-filling stoppage
        fBcurrent = fBprev + deltaH;
        y(nt,14) = fBcurrent;

    elseif stoppage_reason == 3,        % free-emptying stoppage
        fCcurrent = fCprev + deltaH;
        y(nt,16) = fCcurrent;
    else                                % due to air mass increase
        fCcurrent = fCprev + deltaH;
        y(nt,16) = fCcurrent;
    end
```

[17] sat_val.m

```
%% Saturation Function
    function saturated = sat_val(saturated,min_val,max_val)
        if saturated > max_val,     % max bound
            saturated = max_val;
        elseif saturated < min_val, % min bounds
            saturated = min_val;
        end
    end
```

[18] water_height.m

```
% Water Height
    h_wat_fwd_pump = y(7);
    h_wat_aft_pump = y(8);
    h_wat_fwd_ff   = y(14);
    h_wat_aft_ff   = y(15);
    h_wat_fwd_fe   = y(16);
    h_wat_aft_fe   = y(17);
    h_wat_fwd_init = y(18);
    h_wat_aft_init = y(19);
    h_wat_fwd = h_wat_fwd_pump + h_wat_fwd_ff + h_wat_fwd_fe + h_wat_fwd_init;
    h_wat_aft = h_wat_aft_pump + h_wat_aft_ff + h_wat_aft_fe + h_wat_aft_init;
```

[19] xg_shift_control.m

```
%% Ballast xG Shifting Controller

% Default Defines
    control_status     = 0; % "do nothing" fwd control status
    control_status_aft = 0; % "do nothing" aft control status
    A_wat_fwd_val      = 0; % water valve area              [m^2]
    A_wat_aft_val      = 0; % water valve area              [m^2]
    Pump_rate_fwd      = 0; % forward volume flow rate      [m^3/s]
    Pump_rate_aft      = 0; % forward volume flow rate      [m^3/s]
    Air_rate_fwd       = 0; % forward mass flow rate        [kg/s]
    Air_rate_aft       = 0; % aft mass flow rate        [kg/s]
    pump_wat_fwd       = 0; % forward pump state            (off)
    comp_air_fwd       = 0; % forward air compressor state  (off)
    pump_wat_aft       = 0; % forward pump state            (off)
    comp_air_aft       = 0; % forward air compressor state  (off)

% Air Mass
    air_mass();

% Water Height Components
    water_height();

% Conditions
    % Forward
        Pa_elmx  = P_air_fwd <= P_air_fwd_max;              % equal or lower
        f_pcd    = P_air_fwd <= P_comp_fwd;
        Pdiff_h  = P_wat_fwd >  P_air_fwd;                  % positive pressure diff.
        Pdiff_l  = P_wat_fwd <  P_air_fwd;                  % negative pressure diff.
        hwat_l   = h_wat_fwd <  h_wat_max_fwd - 0.005;      % low
        hbal_l   = h_wat_fwd <  h_bal_tank - 0.005;
        hwat_h   = h_wat_fwd >  0.005;                      % high
        mair_m   = m_air_fwd >  0.535;
    % Aft
        Pa_elmxa = P_air_aft <= P_air_aft_max;              % equal or lower
        f_pcda   = P_air_aft <= P_comp_aft;
        Pdiff_ha = P_wat_aft >  P_air_aft;                  % positive pressure diff.
        Pdiff_la = P_wat_aft <  P_air_aft;                  % negative pressure diff.
        hwat_la  = h_wat_aft <  h_wat_max_aft - 0.005;      % low
        hbal_la  = h_wat_aft <  h_bal_tank - 0.005;
        hwat_ha  = h_wat_aft >  0.005;                      % high
        mair_ma  = m_air_aft >  0.535;

% Shift Parameters
    rG_ref_i = [0.00; 0; 0.075];            % [m] initial reference CoG
    rG_ref1  = [xGs; 0; 0.075];             % [m] caused by mod. rG_ref from rG_ref_i
    xG_shift = rG_ref_i(1) - rG_ref1(1);    % desired xG shift (+ = towards fwd)
% Calculate Necessary Shift
    if (shift_cal == 0),
        % Symbolic Variable (requires Symbolic Math Toolbox)
            h_wat_fwd_n = sym('h_wat_fwd_n');

        % Weight & Mass
            xG_bal_fwd1 = xGB_fwd;
            xG_bal_aft1 = xGB_aft;
            zG_bal_fwd1 = -0.5*h_wat_fwd_n + 0.15;
            zG_bal_aft1 = -0.5*h_wat_aft1 + 0.15;
            m_wat_fwd1  = rho_wat*(h_wat_fwd_n - 0.5*h_fwd_bal)*w_fwd_bal^2;
            m_wat_aft1  = rho_wat*(h_wat_aft   - 0.5*h_aft_bal)*w_aft_bal^2;
            mx_total1   = m_wat_fwd1*xG_bal_fwd1 + m_wat_aft1*xG_bal_aft1 +...
                          m_ref(3)*rG_ref1(1);
            mz_total1   = m_wat_fwd1*xG_bal_fwd1 + m_wat_aft1*xG_bal_aft1 +...
                          m_ref(3)*rG_ref1(3);
            mass1       = m_wat_fwd1 + m_wat_aft1 + m_ref(3);           % mass
            rG1         = [mx_total1; 0; mz_total1] / mass1;            % center of mass
        % Desired h_wat_fwd
            h_wat_fwd_n = double(solve(simplify(rG1(1))));

        % Shift Logic
            if w_fwd_bal <= 0.3,
            % Small Ballast Tank
                if xG_shift > 0,    % fwd filled and/or aft emptied
                    h_fwd_d = h_wat_fwd_n;
                    if h_wat_fwd_n > h_wat_max_fwd,
                        h_fwd_d = h_wat_max_fwd;
                        h_aft_d = h_wat_aft - (h_wat_fwd_n - h_wat_max_fwd);
                        if h_aft_d < 0,
                            h_aft_d = 0;
                        end
                    end
                end
                if xG_shift < 0,    % fwd filled and/or aft emptied
                    h_aft_d = h_wat_fwd_n;
                    if h_wat_fwd_n < 0,
                        h_aft_d = 0;
                        h_fwd_d = h_wat_fwd - h_wat_fwd_n;
                        if h_fwd_d > h_bal_tank,
                            h_fwd_d = h_bal_tank;
                        end
                    end
                end
            % Large Ballast Tank
            else
                h_to_shift = h_wat_fwd_n - h_bal_tank/2;
                if abs(h_to_shift) > h_wat_max_fwd,
                    h_to_shift = sign(h_to_shift)*h_wat_max_fwd;
                end
                h_fwd_d = h_bal_tank/2 + h_to_shift/2;
                h_aft_d = h_bal_tank/2 - h_to_shift/2;
            end
        shift_cal = 1;  % used to stop shifting calculations on ever timestep
    end

% Ballast Controller
    if abs(xG_shift) >= 0.001,
    % Quantization
        h_wat_fwd1 = round(h_wat_fwd*1e3)/1e3;
        h_wat_aft1 = round(h_wat_aft*1e3)/1e3;
        h_fwd_d    = round(h_fwd_d*1e3)/1e3;
        h_aft_d    = round(h_aft_d*1e3)/1e3;

    % Desired Ballast Action
        pump_wat_fwd  = (h_fwd_d - h_wat_fwd1) > +0.001;    % fill with fwd water pump
        comp_air_fwd  = (h_fwd_d - h_wat_fwd1) < -0.001;    % empty with fwd air comp
        pump_wat_aft  = (h_aft_d - h_wat_aft1) > +0.001;    % fill with aft water pump
        comp_air_aft  = (h_aft_d - h_wat_aft1) < -0.001;    % empty with aft air comp

    % Forward Ballast Tank
        % Water Pump Support        (Status = 2)
            if pump_wat_fwd && Pa_elmx && hwat_l,
                Pump_rate_fwd  = Pump_rate_fwd_max;
                control_status = 2;
        % Air Venting Support       (Status = 3)
            elseif mair_m && hbal_l;
               control_status = 3;
            end
    % Compressed Air Support    (Status = 5)
        if comp_air_fwd && Pa_elmx && f_pcd && hwat_h,
            if w_fwd_bal <= 0.3,
                Air_rate_fwd = Air_rate_fwd_max;
            else
                Air_rate_fwd = Air_rate_fwd_max*10;
            end
            control_status = 5;
    % Free-emptying;        (Status = 4)
            if Pdiff_l && hwat_h,
                A_wat_fwd_val = A_wat_fwd_val_max;
            end
        elseif comp_air_fwd && Pdiff_l && hwat_h,
            A_wat_fwd_val = A_wat_fwd_val_max;
            control_status = 4;
        end

% Aft Ballast Tank
    % Water Pump Support        (Status = 2)
        if pump_wat_aft && Pa_elmxa && hwat_la,
            Pump_rate_aft  = Pump_rate_aft_max;
            control_status_aft = 2;
    % Air Venting Support       (Status = 3)
        elseif mair_ma && hbal_la;
            control_status_aft = 3;
        end

    % Compressed Air Support    (Status = 5)
        if comp_air_aft && Pa_elmxa && f_pcda && hwat_ha,
            if w_aft_bal <= 0.3,
                Air_rate_aft = Air_rate_aft_max;
            else
                Air_rate_aft = Air_rate_aft_max*10;
            end
            control_status_aft = 5;
    % Free-emptying;            (Status = 4)
            if Pdiff_la && hwat_ha,
                A_wat_aft_val = A_wat_aft_val_max;
            end
        elseif comp_air_aft && Pdiff_la && hwat_ha,
            A_wat_aft_val = A_wat_aft_val_max;
            control_status_aft = 4;
        end

% Water Valve Saturation
    A_wat_fwd_val = sat_val(A_wat_fwd_val,0,A_wat_fwd_val_max);
    A_wat_aft_val = sat_val(A_wat_aft_val,0,A_wat_aft_val_max);
end
```

Appendix C AUV and Component Validation Results

AUV Dynamics & Hydrodynamics

The dynamics and hydrodynamics of the AUV are an important aspect of the simulator design. It is important that they each work properly in order to obtain accurate results. There are four major areas that have been tested to determine the validation of the AUV's dynamics and hydrodynamics. These areas include: status quo, descent and ascent, orientation, bowplane and sternplane deflection fins, and hydrodynamics refer to the *List of Abbreviations and Symbols Used* for additional parameter definitions).

Status Quo

Given an AUV with no propulsion force, external forces, and with a weight equal to its buoyancy, it is expected that the AUV will not have any motion in any direction. Figure 50 plots the AUV depth for this situation. It can be seen that there is no depth change as expected.

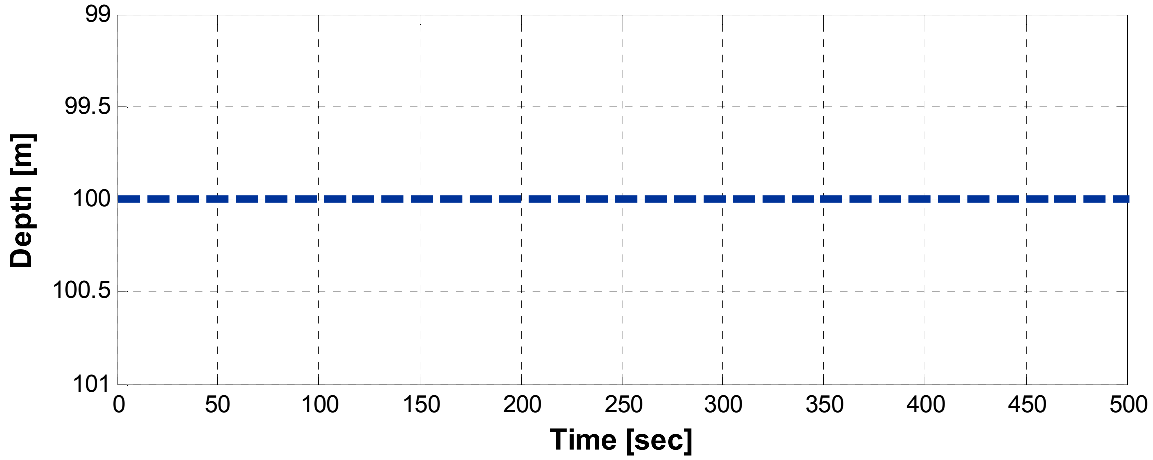

Figure 50 – Validation – Status Quo – Depth (no control response, "small" ballast tanks)

Descent and Ascent

Unpowered, an AUV will begin descending in the inertial frame of reference when it has weight greater than its buoyancy. There are other factors that can affect this motion, such as external environmental conditions or forces. For the following tests, the AUV is unpowered and begins at rest. For the descent test, the weight of the AUV has been set to 85347 N (8,700 kg) and its buoyancy at 84366 N (equivalent to 8,600 kg). It is expected that the AUV will descend due to having a weight that is 981 N (100 kg) higher than its buoyancy. For the ascent test, the weight of the AUV has been set to 84366 N (8,600 kg) and its buoyancy 85347 N (equivalent to 8,700 kg). It is expected that the AUV will ascend due to having a weight that is 981 N (100 kg) lower than its buoyancy. Figure 51 plots the depth of the AUV in response to both weight and buoyancy magnitude increases. Increasing the weight of the AUV, while keeping the buoyancy unchanged, resulted in a combined positive net weight and buoyancy $(net = Weight + Buoyancy)$. When the AUV has a combined positive net weight and buoyancy of 100 N, as represented by the solid navy blue curve, the AUV began and continued to decend. Increasing the buoyancy of the AUV, while keeping the weight unchanged, resulted in a combined negative net weight and buoyancy $(net = Weight + Buoyancy)$. When the AUV has a combined negative net weight and buoyancy of –100 N, as represented by the solid red curve, the AUV began and continued to ascend. This behavious is as expected.

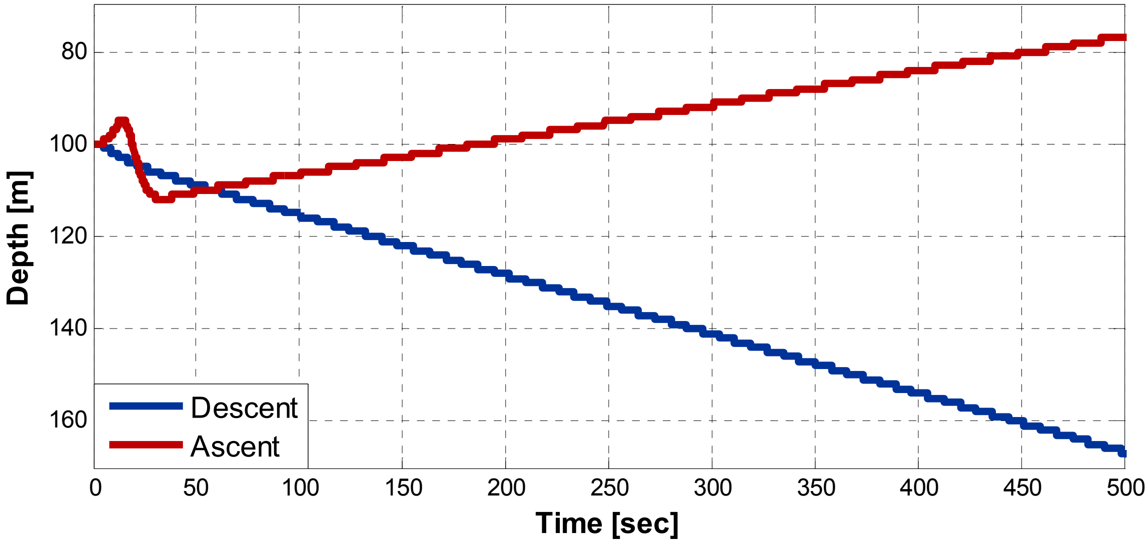

Figure 51 – Validation – Descent and Ascent – Depth (100 N [navy blue] and –100 N (red) combined net weight and buoyancy, “small” ballast tanks)

It should be noted that when ascending, at the beginning, the AUV appears to initially ascend, then briefly descend, and then ascend again. Figure 52 plots the corresponding pitch angle of the AUV in response to the same weight and buoyancy magnitude increases as used in Figure 51. The hydrodynamics cause the AUV to rotate with a large negative angle of approximately –45° before righting itself at approximately –10°. When descending, the angle of the AUV remains approximately constant at about –5°. The hydrodynamic terms in the $\dot{q}$ term grows in relative magnitude initially much faster than the hydrodynamic terms in the $\dot{w}$ equation and are dominant. Once the hydrodynamic terms in the $\dot{w}$ once again (as with the minor initial rise) are dominant in magnitude, the AUV will begin to ascend while the pitch angle becomes steady at approximately –10°.

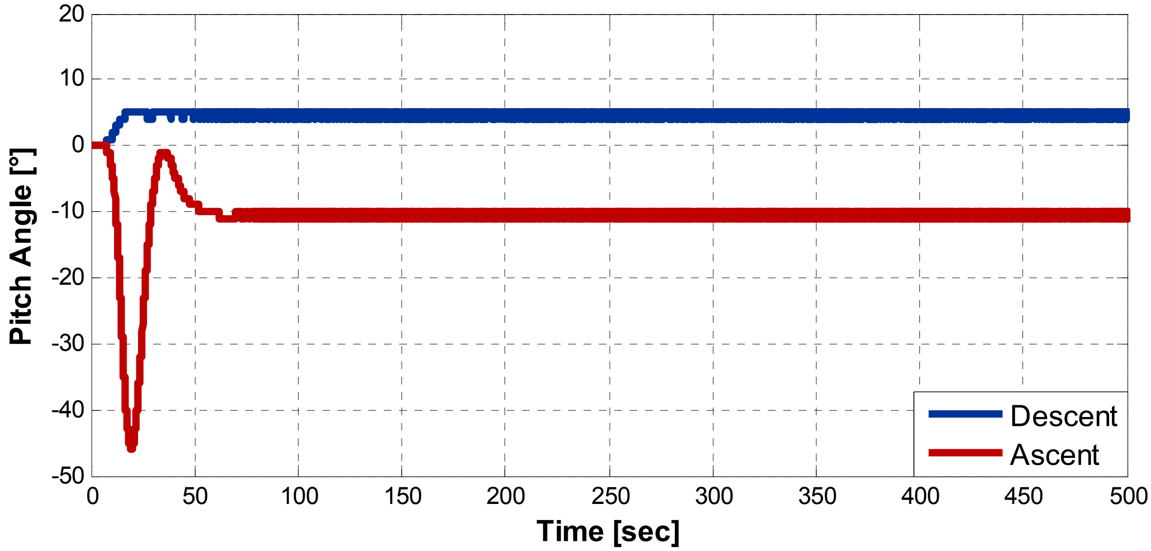

Figure 52 – Validation – Descent and Ascent – Pitch Angle (100 N [navy blue] and –100 N (red) combined net weight and buoyancy, "small" ballast tanks)

Orientation (Pitch Angle)

The orientation of the AUV that will be tested is the pitch angle. Stated simply, it is expected that having a center of gravity that has been shifted forward from the center of buoyancy along the x-axis will cause the AUV to rotation with a positive pitch angle with respect to the direction of orientation. Figure 53 plots the pitch angle of the AUV in response to different ballast tank configurations. The dashed blue curve represents a forward shift as a result of filling the forward ballast tank 100% and emptying the aft ballast tank 100%. The dashed and dotted red curve represents an aft shift as a result of emptying the forward ballast tank 100% and filling the forward ballast tank 100%. For this simulation, the center of gravity is 0.25 m below the origin (z_G). As can be observed in Figure 53 with the forward shift $(x_G = 0.1)$, the AUV initially rotates negatively. With with an aft shift $(x_G = -0.1)$, the AUV initially rotates positively. A positive rotation signifies that the forward portion of the AUV is rotating down which is expected when the AUV is forward-heavy, while the aft portion rotates upwards. Therefore, the AUV is performing rotational behaviour as expected.

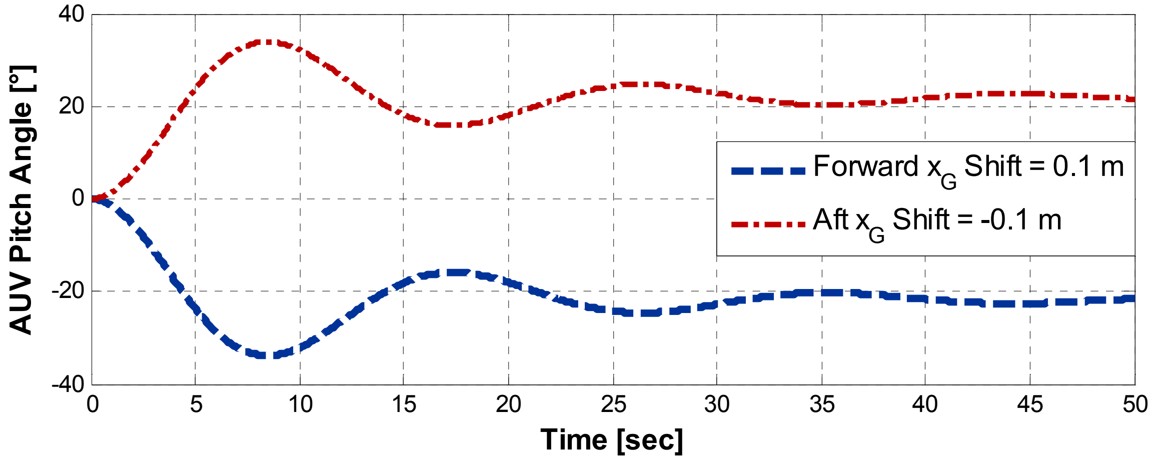

Figure 53 – Validation – Pitch Angle ("small" ballast tanks completely filled and emptied, forward filled 100% / aft emptied 100% [navy blue], forward emptied 100% / aft filled 100% [red])

Figure 54 shows the initial and equilibrium configuration for the AUV. Parameters C_B and C_G represent the center of buoyancy and gravity, respectively (recall: weight W is positive and buoyancy B is negative). The equilibrium pitch angle $\left(\theta_{eq}\right)$ is the angle at which the AUV will remain, after being damped by the hydrodynamics of the AUV, as a result of the offset between C_B and C_G. The equilibrium pitch angle is calculated by geometry as follows:

$$\theta_{eq} = tan^{-1}\left(\frac{x_G}{z_G}\right) = tan^{-1}\left(\frac{0.10}{0.25}\right) = 21.8° \qquad (79)$$

where x_G and z_G represent the distance from the origin to the x-axis and z-axis component of the center of gravity, respectively. As can be observed in Figure 53, with a forward shift $(x_G = 0.1)$, the calculated equilibrium angle of –21.8° is approximately equivalent to the simulated result.

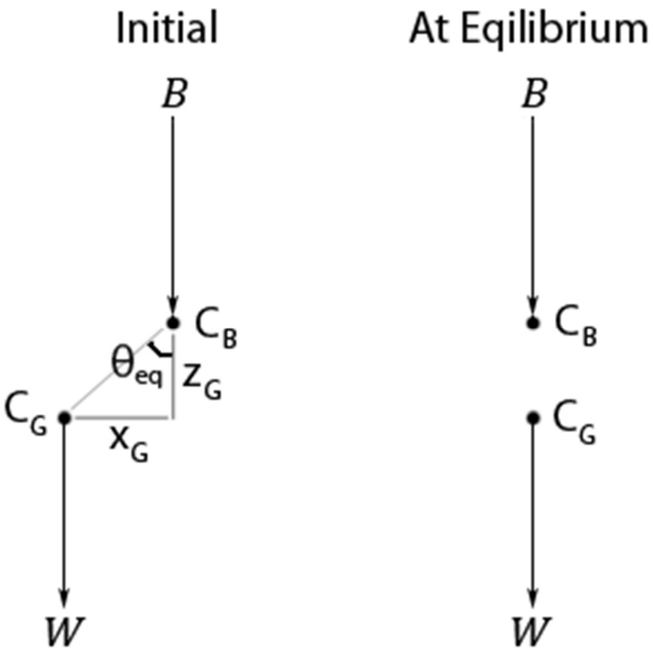

Figure 54 – Initial and Equilibrium Pitch Angle

Bowplane Fins

Tests of the bowplane fins should emphasise depth control, as a result of bowplane angle, when propelled forward. With a negative bowplane angle it is expected that the AUV will descend, while having a positive bowplane angle the AUV should ascend. Figure 55 plots the depth of the AUV in response to both positive and negative bowplane angles. The solid navy blue curve represents a positive bowplane angle and the solid red curve represents a negative bowplane angle. As can be seen, the AUV descends with a negative bowplane angle and ascends with a positive bowplane angle as expected.

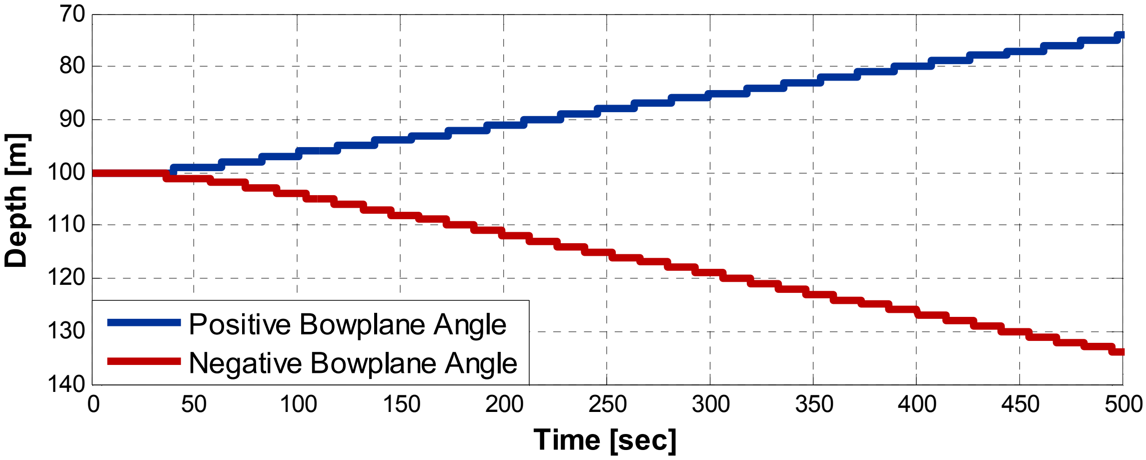

Figure 55 – Validation – AUV Depth – Bowplane (only) Control

Tests of the sternplane fins should emphasise pitch control, as a result of sternplane angle, when propelled forward. With a negative sternplane angle, it is expected that the pitch angle of the AUV will move positively. With a positive sternplane angle, it is expected that the pitch angle of the AUV will move negatively. Figure 56 plots the resulting pitch angle of the AUV in response to both positive and negative sternpalne angles. The solid navy blue curve represents a positive sternplane angle and the solid red curve represents a negative sternplane angle. As can be seen, the pitch angle of the AUV is negative with a positive sternplane angle and the pitch angle of the AUV is positive with a negative sternplane angle as expected.

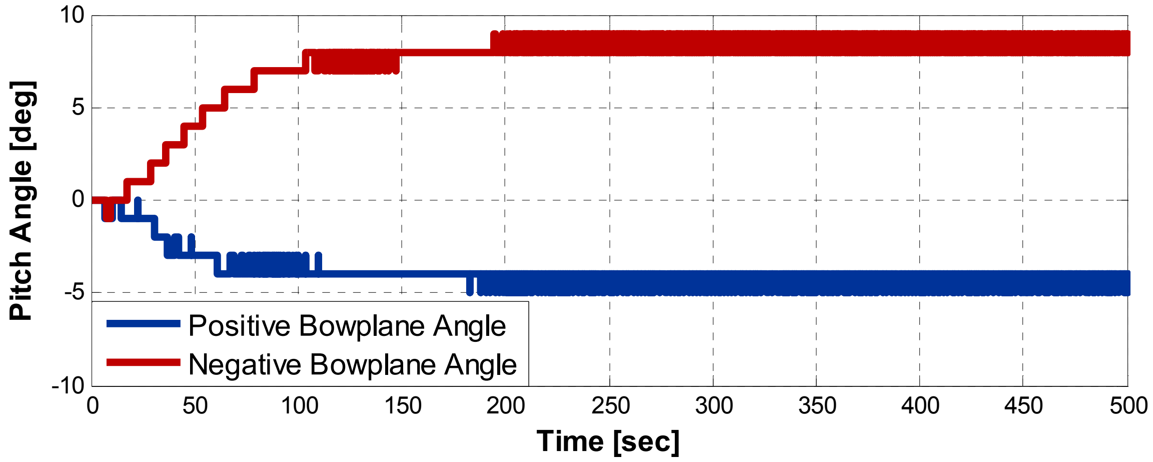

Figure 56 – Validation – AUV Pitch Angle – Sternplane (only) Control

Hydrodynamics

It is expected that, with the addition of hydrodynamics, the AUV will have a dynamic drag force that can cause it to slow down, rotate, ascend, or descend depending on what components of the hydrodynamics are dominating. Figure 57 plots the depth of the AUV in response to a positive combined net weight and buoyancy for simulations with and without hydrodynamics added. The positive net weight and buoyancy is equal to 100 kg ($F = W + B$ =100 kg = 981 N). The dashed and dotted red curve represents an AUV descending without hydrodynamics. The dashed navy blue curve represents the AUV descending with hydrodynamics. As can be observed, once hydrodynamics are introduced, the AUV descends at a lower rate of approximately 10 m over 50 sec (with hydrodynamics) versus approximately 60 m over 50 sec.

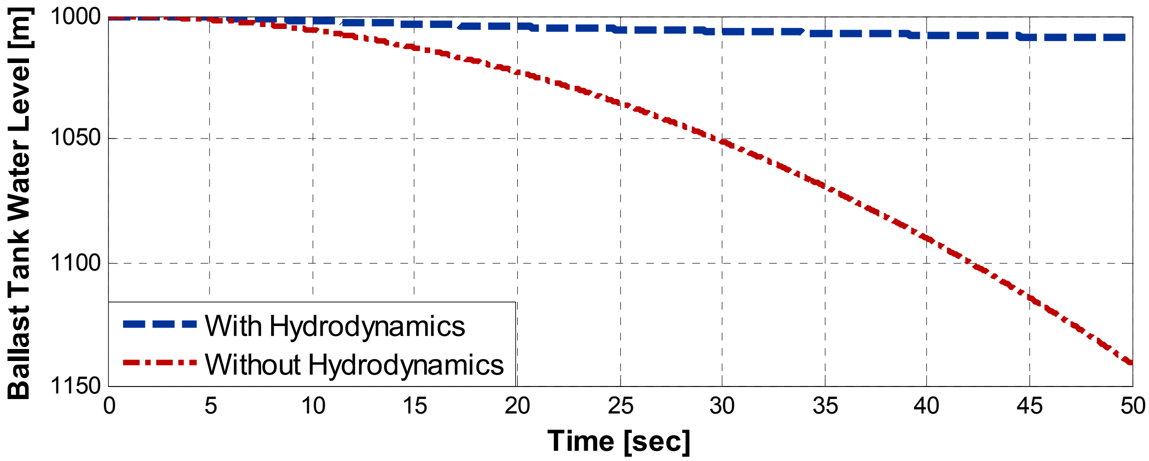

Figure 57 – Validation – Hydrodynamics – Descending ($F = W + B$ =100 kg)

Based on motion with constant acceleration, the final displacement of the AUV without hydrodynamics can be calculated by the following:

$$a_{AUV} = \frac{F_{Z,AUV}}{mass} = \frac{981\ N}{8700\ kg} = 0.113\ m/s^2 \tag{80}$$

$$d_f = d_i + d_t = 1{,}000 + \frac{1}{2}a \cdot t = 1{,}000 + (0.5)(0.113)(50)^2 = 1{,}140.9\ m \tag{81}$$

where a_{AUV} represents the acceleration of the AUV, $F_{Z,AUV}$ is the net weight and buoyancy force of the AUV, $mass$ represents the mass of the AUV, d_f represents the final displacement, d_i represents the initial displacement, and d_t represents the travelled displacement. As can be observed in Figure 57, descending without hydrodynamics, the calculated final displacement of 1,140.9 m is equivalent to the simulated result.

Ballast Tank

There are four major aspects of the ballast tank system that must be validated: free-filling, free-emptying, water pump support, and compressed air support. Each of these will be tested to show that the contents of the ballast tanks are changing as expected.

Free-Emptying

When undergoing free-emptying, the pressure of the air within the ballast tank will be greater than that of the external ocean water. This situation will cause the water in the ballast tank to flow out of the ballast tank at a rate that is approximately proportional to the pressure difference. Figure 58 and Figure 60 plot the height of water in the ballast tank during a free-emptying ballast tank operation, represented by the dashed navy blue curve. Figure 58 and Figure 60 also plot the maximum water level in the ballast tank due to the current ballast tank air pressure, represented by the dashed and dotted red curve. Figure 59 and Figure 61 plot the pressure of air in the ballast tank and corresponding ocean pressure during a free-emptying ballast tank operation. In Figure 58 and Figure 59, it can be observed, that, for as long as the pressure inside the ballast tank remains greater than that of the ocean ($water\ pressure$), the water height continues to lower. However, as observed in Figure 60 and Figure 61, once the pressure of the ocean becomes equal to that of the air in the ballast tank, the water decrease ends as the ballast tank system has become neutrally buoyant. The initial pressure of air in Figure 59 is 5 MPa versus that of 2 MPa in Figure 61. The higher air pressure of 5 MPa, from Figure 59, allows for greater free-emptying potential than 2 MPa due to the pressure difference between the ocean and ballast tank air remaining negatively longer under identical circumstances, thus prolonging the condition required for free-emptying. Free-emptying support occurs as expected.

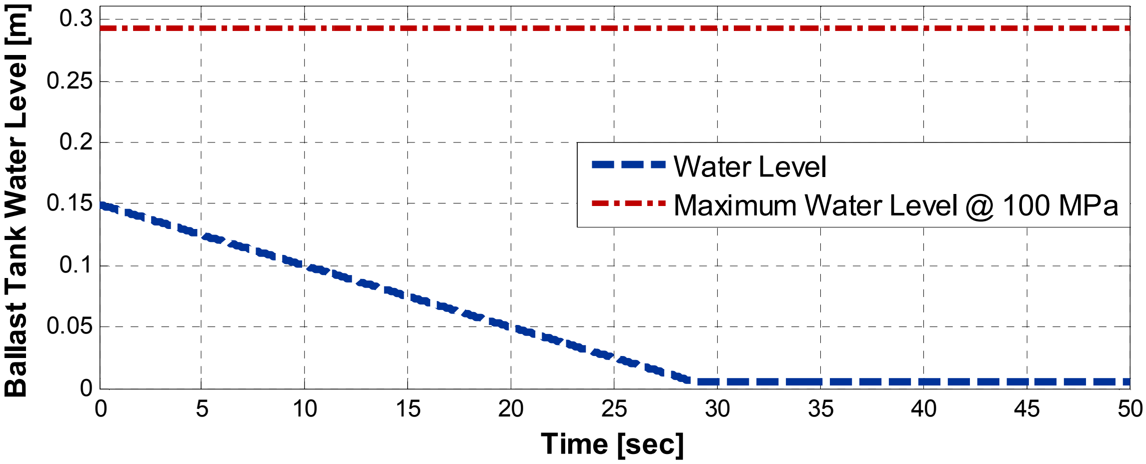

Figure 58 – Validation – Free-emptying – Ballast Tank Water Level ($depth_{AUV}$ = 1,000 m, $P_{air,initial}$ = 5 MPa)

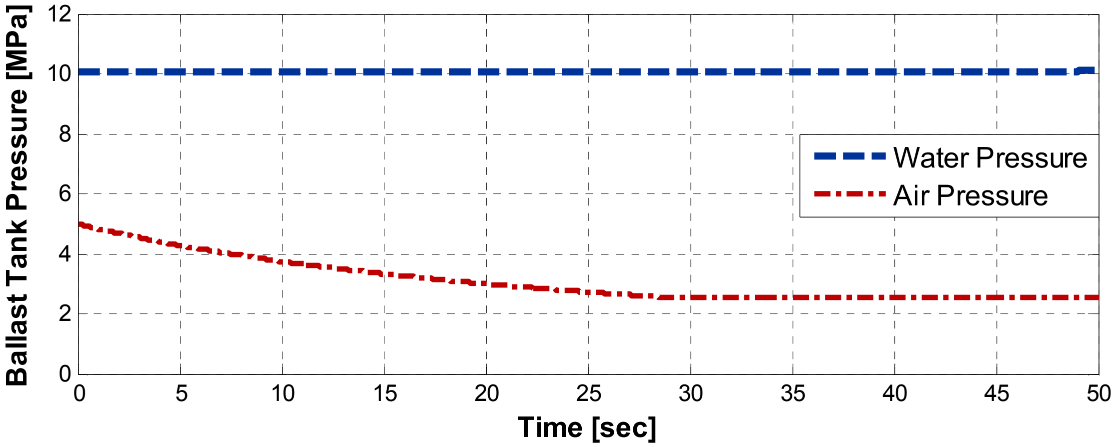

Figure 59 – Validation – Free-emptying – Ballast Tank Pressure ($depth_{AUV}$ = 1,000 m, $P_{air,initial}$ = 5 MPa)

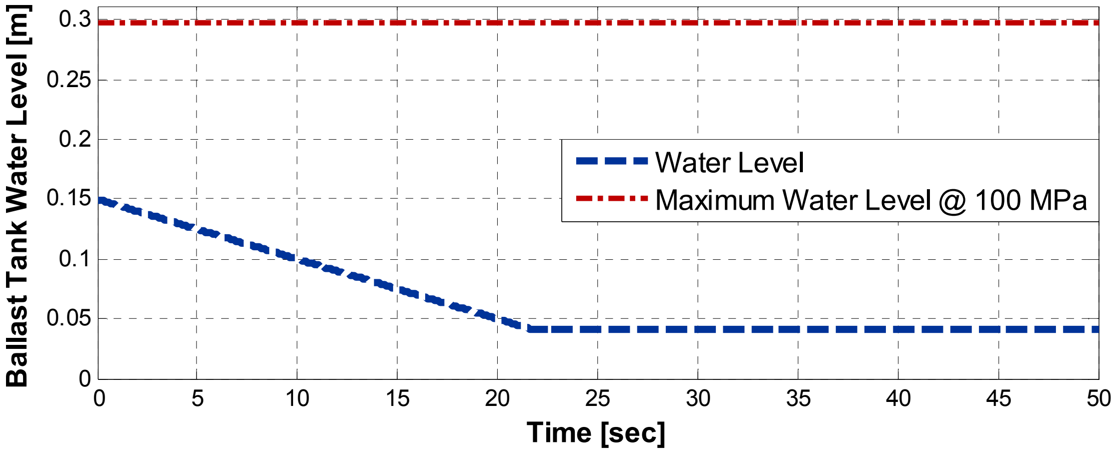

Figure 60 – Validation – Free-emptying – Ballast Tank Water Level ($depth_{AUV}$ = 100 m, $P_{air,initial}$ = 2 MPa)

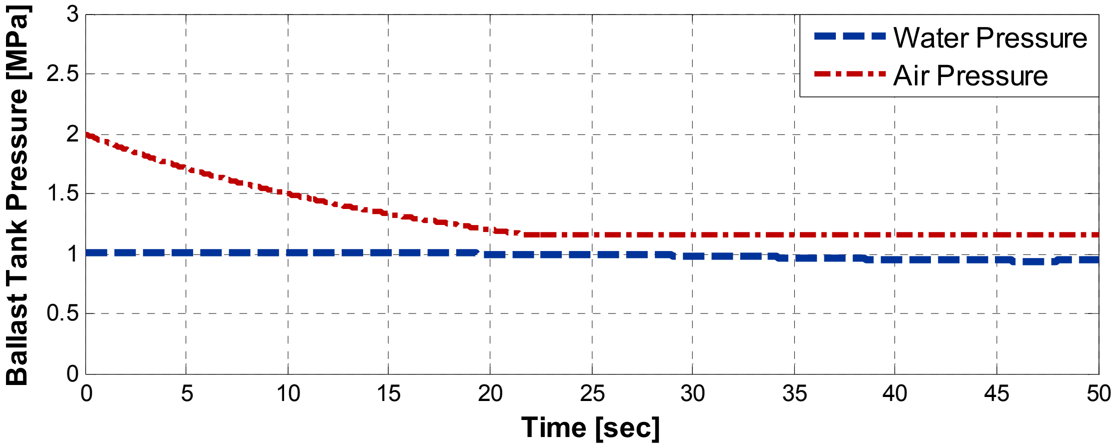

Figure 61 – Validation – Free-emptying – Ballast Tank Pressure ($depth_{AUV}$ = 100 m, $P_{air,initial}$ = 2 MPa)

Free-Filling

Similar to free-emptying, free-filling will occur when the air pressure within the ballast tank is less than that of the external ocean water. This situation will cause the water in the ballast tank to flow into the ballast tank at a rate that is approximately proportional to the pressure difference. Figure 62 and Figure 64 plot the height of water in the ballast tank during a free-filling ballast tank operation, represented by the dashed navy blue curve. Figure 62 and Figure 64 also plot the maximum water level in the ballast tank due to the current ballast tank air pressure, represented by the dashed and dotted red curve. Figure 63 and Figure 65 plot the pressure of air in the ballast tank and corresponding ocean pressure during a free-filling ballast tank operation. In Figure 62 and Figure 63, it can be observed, that, for as long as the air pressure inside the ballast tank remains lower than that of the ocean ($water\ pressure$), the water height continues to raise. However, as observed in Figure 64 and Figure 65, once the pressure of air in the ballast tank becomes equal to that of the water in the ocean, the water increase ends as the ballast tank system has become neutrally buoyant. The initial pressure of air in Figure 62 is 0.1 MPa versus that of 5 MPa in Figure 63. The lower air pressure of 0.1 MPa, from Figure 62, allows for greater free-filling potential than 5 MPa due to the pressure difference between the ocean and ballast tank air remaining positive longer under identical circumstances, thus prolonging the condition required for free-filling. Free-filling support occurs as expected.

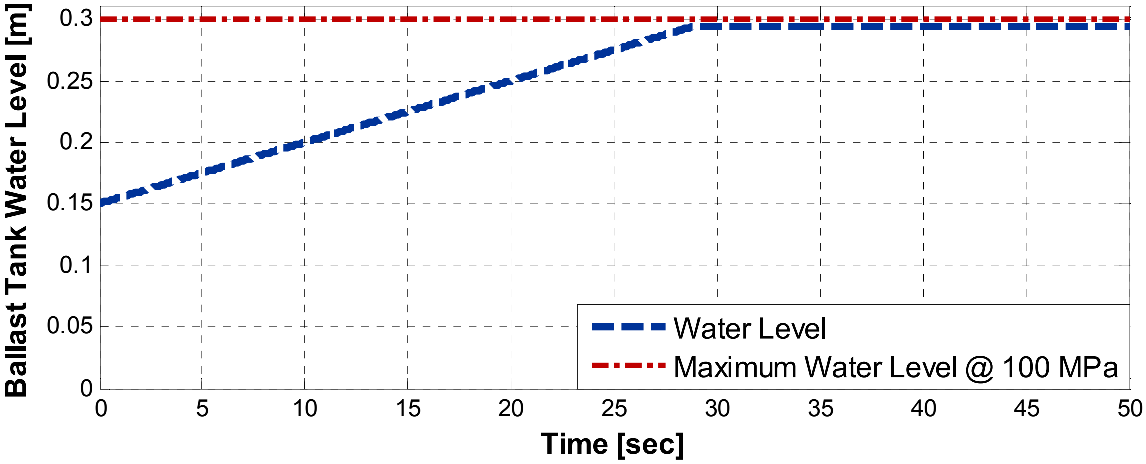

Figure 62 – Validation – Free-filling – Ballast Tank Water Level ($depth_{AUV}$ = 1,000 m, $P_{air,initial}$ = 0.1 MPa)

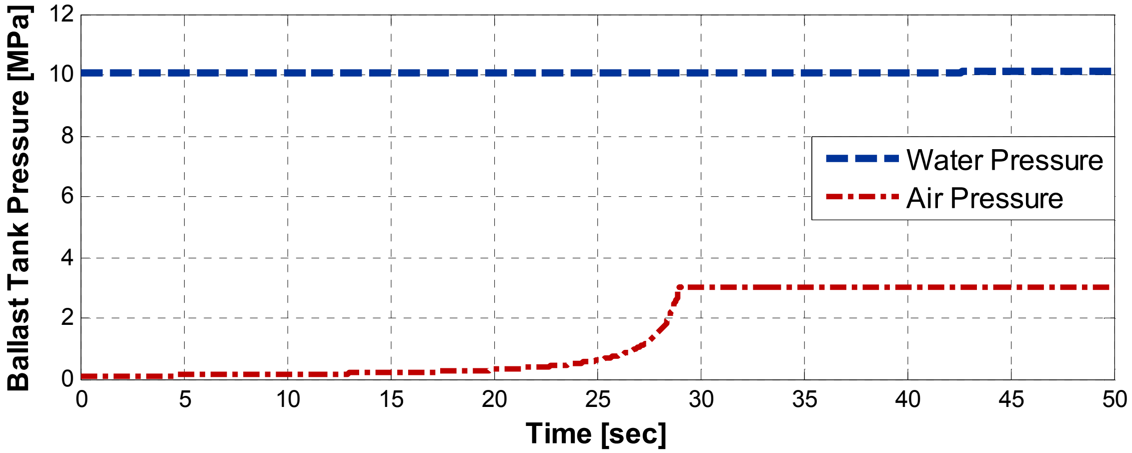

Figure 63 – Validation – Free-filling – Ballast Tank Pressure ($depth_{AUV}$ = 1,000 m, $P_{air,initial}$ = 0.1 MPa)

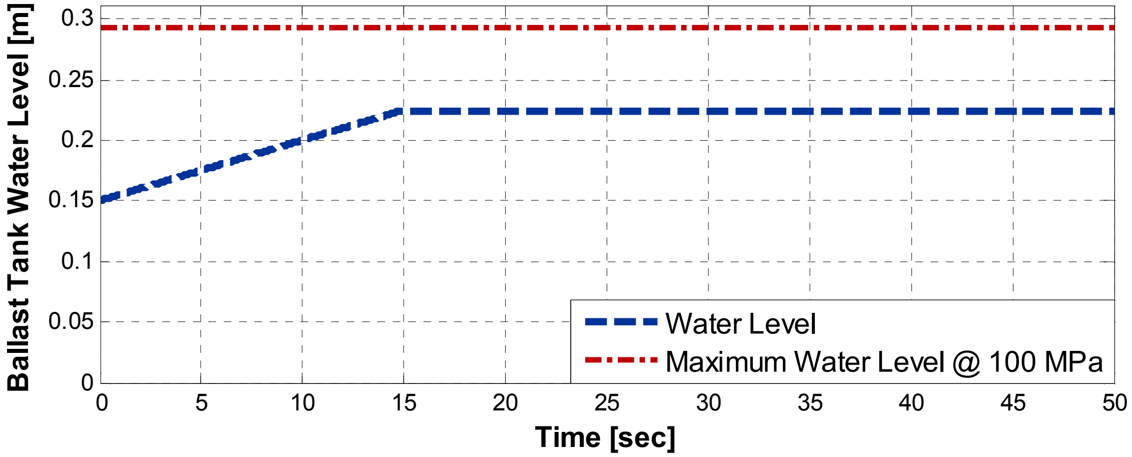

Figure 64 – Validation – Free-filling – Ballast Tank Water Level ($depth_{AUV}$ = 1,000 m, $P_{air,initial}$ = 5 MPa)

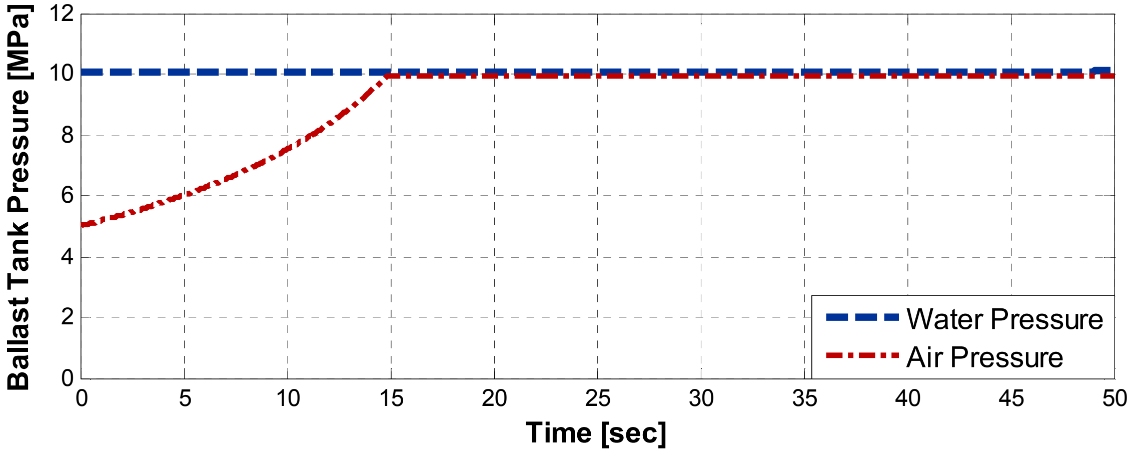

Figure 65 – Validation – Free-filling – Ballast Tank Pressure ($depth_{AUV}$ = 1,000 m, $P_{air,initial}$ = 5 MPa)

In Figure 64 and Figure 65, once the air pressure became approximately equivalent to the water pressure, the water level could no longer rise. The following calculations determine the height at which the ballast tank water will stop at due to this equalization of pressures:

$$P_{air} \cdot V_{air} = m_{air} \cdot R_{air} \cdot T_{air} \tag{82}$$

$$m_{air} = \frac{P_{air,initial} \cdot V_{air}}{R_{air} \cdot T_{air}} = \frac{(5{,}000{,}000)(0.15 \cdot 0.30 \cdot 0.30)}{(286.9)(293)} = 0.803\, kg \tag{83}$$

$$P_{air,final} = P_{wat} = 10\, MPa \tag{84}$$

$$V_{air} = \frac{m_{air} \cdot R_{air} \cdot T_{air}}{P_{wat}} = \frac{(0.803)(286.9)(293)}{10{,}000{,}000} = 0.0068\, m^3 \tag{85}$$

$$h_{air,min} = \frac{V_{air}}{{w_{bal}}^2} = \frac{0.0068}{0.30^2} = 0.076\, m \tag{86}$$

$$h_{wat,max} = h_{bal\,tank} - h_{air,min} = 0.30 - 0.076 = 22.4\, m \tag{87}$$

where V_{air} represents the volume of the ballast tank, $P_{air,initial}$ represents the initial air pressure in the ballast tank, $P_{air,final}$ represents the final air pressure in the ballast tank, $h_{bal\,tank}$ represents the height of the ballast tank, $h_{air,min}$ represents the minimum air height in the ballast tank, and $h_{wat,max}$ represents the maximum water height in the ballast tank. As can be observed in Figure 64, the ballast tank water level increases to the calculated value of approximately 22.4, at which point the air pressure and water pressure within the ballast tank is approximately equivalent.

Water Pump Support

When water pump support is activated, the ballast tank water valve will be closed and ocean water will be pumped into the ballast tank. Figure 66 plots the height of water in the ballast tank during a water pump ballast tank operation, represented by the dashed navy blue curve. Figure 66 also plots the maximum water level in the ballast tank due to the current ballast tank air pressure, represented by the dashed and dotted red curve. Figure 67 plots the pressure of air in the ballast tank and corresponding ocean pressure during a water pump support ballast tank operation. It is expected that the water will continue to increase regardless of the internal air pressure, until the air becomes compressed considerably. At this point, the compressed air has a pressure that is equal to the set maximum pressure of the ballast tank and the water pump is no longer able to continue pumping water into the ballast tank. Figure 66 and Figure 67 show the height of the water in the ballast tank increasing to approximately 0.3 m while the air pressure in the ballast tank increases to approximately 85 MPa. The maximum air pressure of 100 MPa could not be achieved due to the soft saturation within the ballast tank which stops water pump support when the height of the water in the ballast tank is within 0.005 m of the maximum water height (approximately 1.7% of the ballast tank height of 0.3 m). Water pump support occurs as expected.

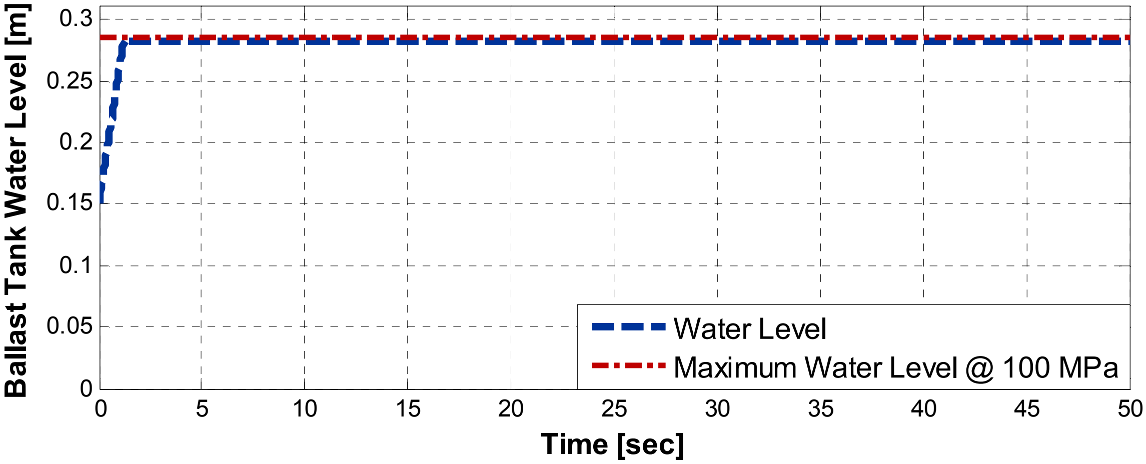

Figure 66 – Validation – Water Pump Support – Ballast Tank Water Level ($P_{air,initial}$ = 10 MPa)

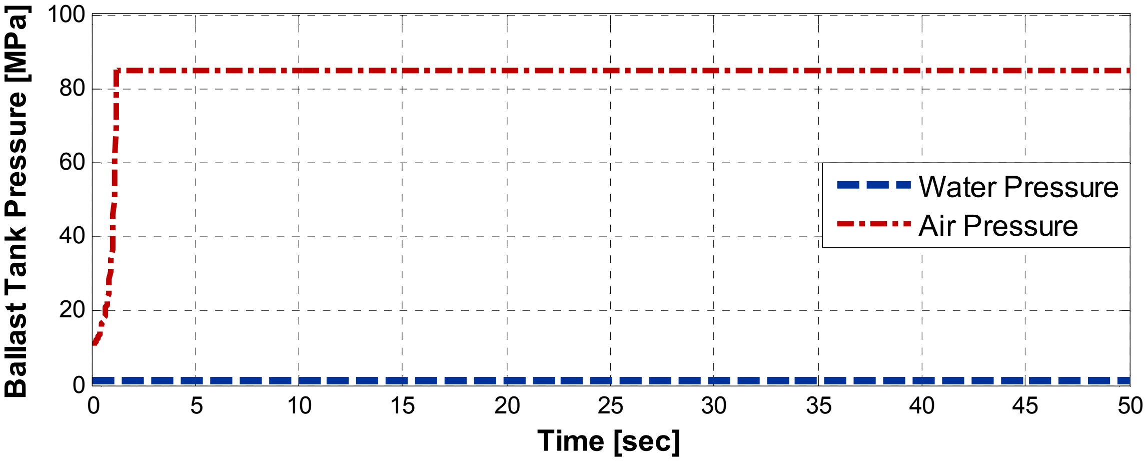

Figure 67 – Validation – Water Pump Support – Ballast Tank Pressure ($P_{air,initial}$ = 10 MPa)

Compressed Air Support

When compressed air support is activated, the ballast tank water valve can be either opened or closed. However, if opened, there is potential for undesired free-filling. When the air mass in the ballast tank is increased, it is expected that the internal air pressure will also increase eventually forcing water out of the ballast tank. Figure 68 plots the height of water in the ballast tank during a compressed air cylinder ballast tank operation, represented by the dashed navy blue curve. Figure 68 also plots the maximum water level in the ballast tank due to the current ballast tank air pressure, represented by the dashed and dotted red curve. Figure 69 plots the pressure of air in the ballast tank and corresponding ocean pressure during a compressed air support ballast tank operation. It is expected that the air mass, and thus air pressure, will continue to increase until free-emptying occurs and completes, or until the maximum air pressure is met, whichever happens first. Once free-emptying can occur, due to a negative pressure difference, it can continue until the water has been decreased to the setpoint amount (in this case, completely removed). Figure 69 shows the pressure of air in the ballast tank increasing until it is slightly higher than the maximum amount of 1 MPa (which is higher than the 1 MPa of the ocean water at 1,000 m). Once the air pressure is higher than the ocean pressure, free-emptying occurs as can be seen in the decreasing ballast tank water level in Figure 68. Compressed air support occurs as expected.

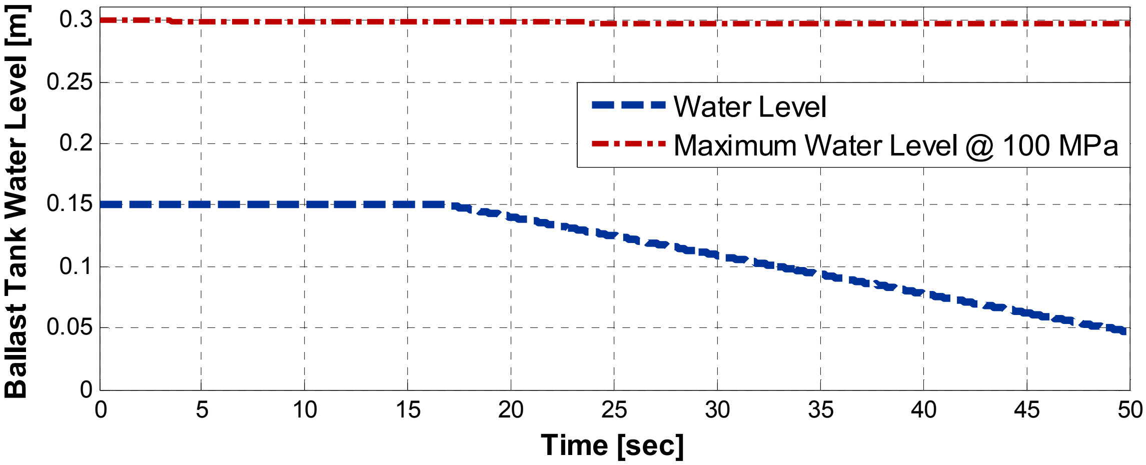

Figure 68 – Validation – Compressed Air Support – Ballast Tank Water Level ($depth_{AUV}$ = 100 m, $P_{air,initial}$ = 0.1 MPa)

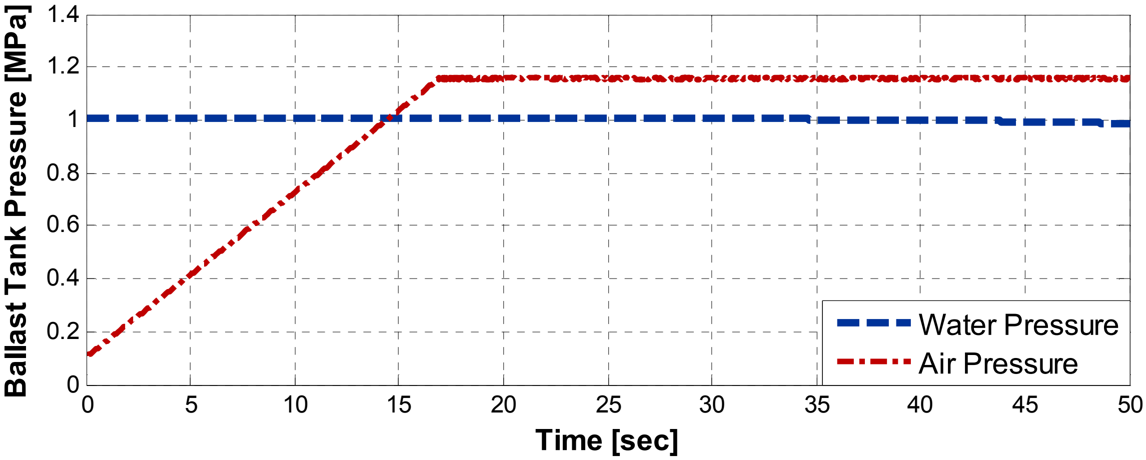

Figure 69 – Validation – Compressed Air Support – Ballast Tank Pressure ($depth_{AUV}$ = 100 m, $P_{air,initial}$ = 0.1 MPa)

Air Mass Venting

When air venting is activated, the ballast tank air valve is opened to allow some of the air mass to exit the ballast tank in order to decrease the pressure of air in the ballast tank. Figure 70 plots the height of water in the ballast tank during an air venting ballast tank operation, represented by the dashed navy blue curve. Figure 70 also plots the maximum water level in the ballast tank due to the current ballast tank air pressure, represented by the dashed and dotted red curve. Figure 71 plots the pressure of air in the ballast tank and corresponding ocean pressure during an air venting support ballast tank operation. It is expected that when the water height of the ballast tank wants to increase, but cannot due to the air pressure (and thus air mass) being above 100 MPa, the air valve will open allowing the air mass to decrease. Figure 70 shows the maximum height and increasing water level in the ballast tank while Figure 71 shows the air mass decreasing and then increasing continuously. As the air mass decreases, this allows the maximum water height to lower and the ballast tank water level to increase. As the ballast tank water level increases, the air pressure in the ballast tank increases. Therefore, the process will repeat as more air is vented to allow for the maximum water height to decrease and thus more water will be brought in. Air venting support occurs as expected.

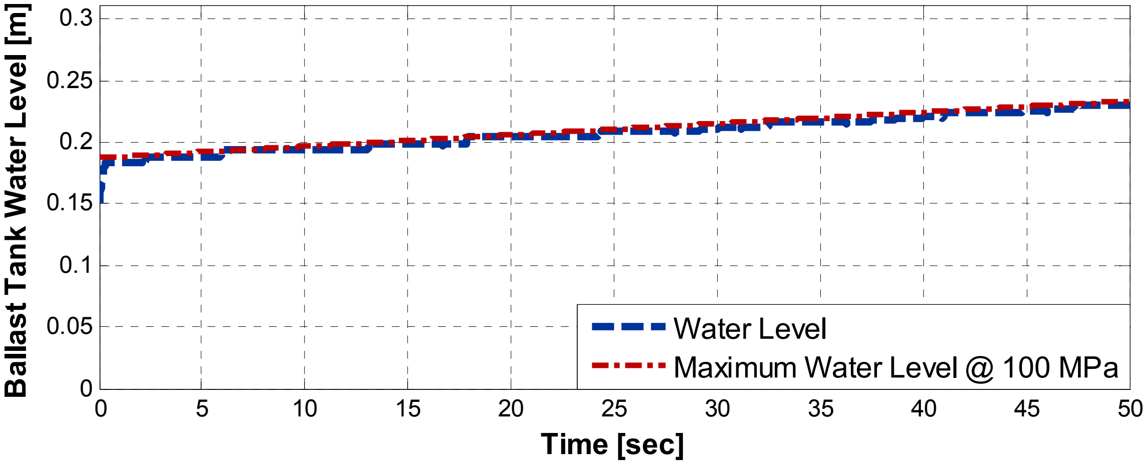

Figure 70 – Validation – Air Venting Support – Ballast Tank Water Level ($P_{air,initial}$ = 75 MPa)

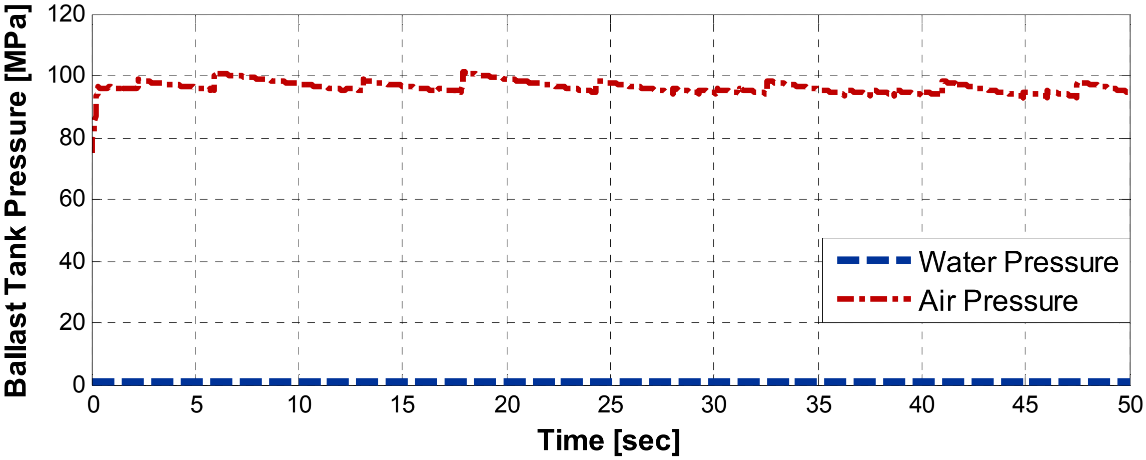

Figure 71 – Validation – Air Venting Support – Ballast Tank Water Level ($P_{air,initial}$ = 75 MPa)