Spring 5-2022

Development of Modeling and Simulation Platform for Path-Planning and Control of Autonomous Underwater Vehicles in Three-Dimensional Spaces

Sai Krishna Abhiram Kondapalli
Old Dominion University, skond002@odu.edu

DEVELOPMENT OF MODELING AND SIMULATION PLATFORM FOR PATH-PLANNING AND CONTROL OF AUTONOMOUS UNDERWATER VEHICLES IN THREE-DIMENSIONAL SPACES

by

Sai Krishna Abhiram Kondapalli
B.Tech. May 2017, Koneru Lakshmaiah University, India

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

ENGINEERING – MECHANICAL

MECHANICAL AND AEROSPACE ENGINEERING DEPARTMENT

OLD DOMINION UNIVERSITY
May 2022

Approved by:

Dr. Krishnanand Kaipa(Director)

Dr. Sebastian Bawab (Member)

Dr. Tian-Bing Xu (Member)

ABSTRACT

DEVELOPMENT OF MODELING AND SIMULATION PLATFORM FOR PATH-PLANNING AND CONTROL OF AUTONOMOUS UNDERWATER VEHICLES IN THREE-DIMENSIONAL SPACES

Sai Krishna Abhiram Kondapalli
Old Dominion University, 2022
Director: Dr. Krishnanand Kaipa

Autonomous underwater vehicles (AUVs) operating in deep sea and littoral environments have diverse applications including marine biology exploration, ocean environment monitoring, search for plane crash sites, inspection of ship-hulls and pipelines, underwater oil rig maintenance, border patrol, etc. Achieving autonomy in underwater vehicles relies on a tight integration between modules of sensing, navigation, decision-making, path-planning, trajectory tracking, and low-level control. This system integration task benefits from testing the related algorithms and techniques in a simulated environment before implementation in a physical test bed. This thesis reports on the development of a modeling and simulation platform that supports the design and testing of path planning and control algorithms in a synthetic AUV, representing a simulated version of a physical AUV. The approach allows integration between path-planners and closed-loop controllers that enable the synthetic AUV to track dynamically feasible trajectories in three-dimensional spaces. The dynamical behavior of the AUV is modeled using the equations of motion that incorporate the effects of external forces (e.g., buoyancy, gravity, hydrodynamic drag, centripetal force, Coriolis force, etc.), thrust forces, and inertial forces acting on the AUV. The equations of motion are translated into a state space formulation and the S-function feature of the Simulink and MATLAB scripts are used to evolve the state trajectories from initial conditions. A three-dimensional visualization of the resulting AUV motion is achieved by feeding the corresponding position and orientation states into an animation code. Experimental validation is carried out by performing

integrated waypoint planner (e.g., using the popular A* algorithm) and PD controller implementations that allow the traversal of the synthetic AUV in two-dimensional (XY, XZ, YZ) and three-dimensional spaces. An underwater pipe-line inspection task carried out by the AUV is demonstrated in a simulated environment. The simulation testbed holds a potential to support planner and controller design for implementation in physical AUVs, thereby allowing exploration of various research topics in the field.

This thesis is dedicated to my mother, my father, my sister, my brother-in-law, and my niece.

ACKNOWLEDGEMENTS

Thank you to the Old Dominion University Mechanical and Aerospace Engineering Department and faculty for providing me with the knowledge and skills needed to further my education. A special thank you to my advisor, Dr. Kaipa, for supporting and guiding me through my master's work and graduate education. Thank you to Dr. Bawab and Dr. Tian-Bing Xu for being a part of my thesis committee and devoting time and effort to this study. And thank you to family and friends who have supported and encouraged me to continue to pursue my goals.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

1.1. Motivation

Autonomous underwater vehicles (AUVs) operating in deep sea and littoral environments have diverse applications including marine biology exploration (Sagala and Bambang, 2011), ocean environment monitoring (Ramos et al., 2001), search for plane crash sites (Becker et al., 2012), inspection of ship-hulls (Kim, Ayoung, and Ryan Eustice, 2009) and pipelines (Liu et al., 2018), harbor patrolling (Allotta et al., 2017), etc. Depending on the mission, the distances traveled by the AUVs can drastically vary anywhere from 500 meters (e.g., in a ship hull inspection task) to thousands of kilometers (e.g., in an ocean environment monitoring task).

Achieving autonomy in underwater vehicles relies on a tight integration between modules of sensing, navigation, decision-making, path-planning, trajectory tracking, and low-level control. However, challenging conditions of the deep sea (e.g., limited communication and refueling/recharging resources, strong disturbances due to ocean currents, etc.) make it highly difficult to accomplish the tasks of feasibility testing, field trials, and deployment. Owing to these reasons, the AUV system integration effort can benefit from testing the related algorithms and techniques in a simulated environment before implementation in a physical test bed. For example, an AUV simulator allows preliminary testing of various autonomy modules, configurations, and mission scenarios without the fear of losing the physical AUV in the deep sea during the development phase. Difficulty in reaching the test sites and conducting experimental trials can also be avoided by using an AUV simulator.

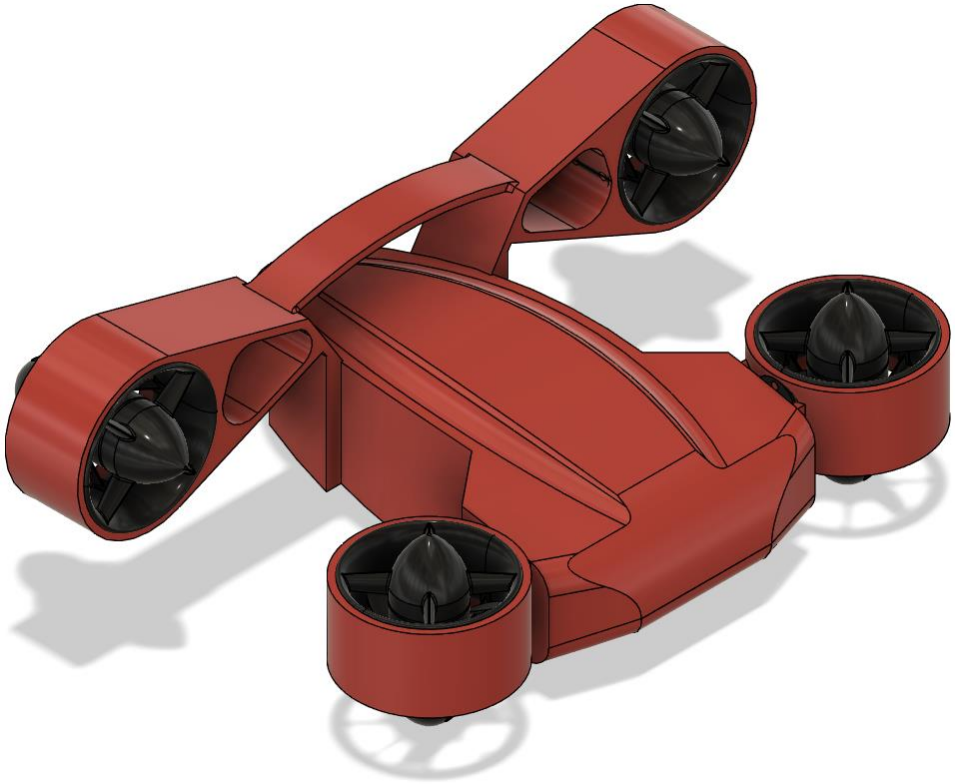

Figure 1: Assembly CAD model of the AUV prototype designed at the Collaborative Robotics & Adaptive Machines Laboratory, ODU. Four brushless DC motor-based thrusters are used to achieve surge, heave, and yaw motions in the AUV.

1.2. Goals and Contributions

This thesis reports on the development of a modeling and simulation platform that supports the design and testing of path-planning and control algorithms in a synthetic AUV, representing a simulated version of a physical AUV. The approach allows integration between path-planners and closed-loop controllers that enable the synthetic AUV to track dynamically feasible trajectories in three-dimensional spaces. The actuation model of the synthetic AUV assumes a thrust force in the surge and heave directions and a torque about the yaw direction. Sway motion is achieved by a combination of surge and yaw motions. The pitch and roll

motions are considered as underactuated degrees of freedom. A cylindrical coordinate system is used to match the actuation model of the AUV. This allows the AUV to move to any position in a three-dimensional space by yawing about the vertical axis, surging along the radial axis, and then heaving along the vertical axis.

The synthetic AUV is built in four stages:

1) The dynamical behavior of the AUV is modeled using the equations of motion that incorporate the effects of external forces (e.g., buoyancy, gravity, hydrodynamic drag, centripetal force, Coriolis force, etc.), thrust forces, and inertial forces acting on the AUV
2) The equations of motion are translated into a state space formulation using a twelve-dimensional state of the AUV defined by its three-dimensional position, orientation, linear velocity, and angular velocity variables
3) Numerical integration of the state derivative equations is performed, using the S-function feature of the Simulink and MATLAB scripts, in order to evolve the state trajectories from initial conditions,
4) A three-dimensional visualization of the resulting AUV motion is achieved by feeding the corresponding position and orientation states into an animation code. A CAD model of the physical AUV is designed and imported into the animation in order to create a realistic motion of the synthetic AUV in a three-dimensional space.

Experimental validation is carried by performing integrated waypoint planner (e.g., using the popular A^* algorithm) and PD controller implementations that allow the traversal of the synthetic AUV in two-dimensional (XY, XZ, YZ) and three-dimensional spaces. An underwater pipe-line inspection task carried out by the AUV is demonstrated in a simulated environment. The simulation testbed holds a potential to support planner and controller design

for implementation in physical AUVs, thereby allowing exploration of various research topics in the field.

1.3. Thesis Organization

The thesis is organized as follows: Chapter 2 presents a literature review related to the techniques presented in this work. Chapter 3 describes the development of the modeling and simulation platform that allows a synthetic AUV to execute path-planning and control algorithms. Chapter 4 presents the results from simulation experiments conducted to validate the simulation testbed. Chapter 5 presents the thesis conclusions and future work.

CHAPTER 2

RELATED WORK

Research on autonomous underwater vehicles began in the late 1980s (Hyland, 1989; Warren 1990). Initial work focused on obstacle avoidance algorithms (Hyland, 1989) and approaches like artificial potential fields (Warren, 1990) and A* algorithm (Carroll et al., 1992) for path planning of AUVs. However, achieving autonomy in unmanned vehicles does not only depend on obstacle avoidance and path-planning, but also a tight integration with other crucial modules like sensing, navigation, decision-making, trajectory tracking, and low-level control. While these modules are necessary for all autonomous vehicles, their implementation in AUVs becomes highly challenging due to factors like strong disturbances caused by ocean currents, operation in a three-dimensional environment, limited communication and refueling or recharging resources, etc.

This spectrum of challenges has attracted wide attention from researchers worldwide, attempting to address different aspects of achieving autonomy in underwater vehicles (Cheng et al., 2021; Panda et al, 2020; Li et al., 2018; Kinsey et al., 2006). This thesis is focused on the path-planning and control aspects of the overall problem. In recent years, there is a lot of research done on path-planning and control algorithms for AUVs. However, this work was mostly carried out in an isolated manner. That is, researchers focused either on path planning (follow paper) or controllers (follow paper). Relatively, few focused on integrating both modules (cite), which is similar to our focus in this thesis.

The challenging conditions of the deep sea make it highly difficult to accomplish the tasks of feasibility testing, field trials, and deployment. Owing to these reasons, the AUV system integration effort can benefit from testing the related algorithms and techniques in a simulated environment before implementation in a physical test bed. For example, an AUV

simulator allows preliminary testing of various autonomy modules, configurations, and mission scenarios without the fear of losing the physical AUV in the deep sea during the development phase. Difficulty in reaching the test sites and conducting experimental trials can also be avoided by using an AUV simulator.

Based on the above analysis of the challenges and prior work in the field of AUVs relevant to the goals and contributions of this thesis, we provide a literature survey of the following aspects:

- Path-Planning
- Control
- Dimension of the operating environment
- AUV Simulators

2.1 Path-Planning

The problem of AUV path planning deals with finding an optimal route from an initial location to a destination location in an underwater environment (Panda et al, 2020). Typically, the route is specified in terms of a course of waypoints that must be traversed by the AUV. Although the underwater environment is hostile and dynamic in nature, its effect on path-planning can be assumed to be predictable for most applications.

Many graph-searches based path-planning algorithms have been developed for such predictable environments. Among these, A* algorithm is one of the most popular methods applied for path-planning of AUVs (Likhachev et al., 2005). Other methods like D* (Carsten et al., 2006) and fast marching (FM) algorithm (Petres et al., 2005) have been employed to solve the same problem. Petres et al. (2005) provided FM-based path planning to deal with a dynamic environment. This method is accurate but also computationally expensive than A*. Later, FM* (heuristically guided FM) was developed (Petres et al. 2007) that preserves the

accuracy of the FM and efficiency of the A* algorithm. One limitation of heuristic grid-search based methods involves discrete-state transitions, which restrict the vehicle's motion to limited directions (Zadeh et al., 2018).

2.2 Control

Many control strategies have been developed to stabilize the AUV in the presence of unknown perturbations and modeling uncertainties (Guerrero et al., 2018). Control is required at three levels of autonomy including stabilization to a point, path following, and trajectory tracking (Encarnac$\tilde{a}$o and Pascoal, 2001). The AUV control system is traditionally based on a proportional–integral–derivative (PID) controller. PID systems can be divided into proportional–integral (PI) or proportional–derivative (PD) systems, depending on whether the vehicle is inertia sensitive (Chen et al., 2013).

Different control methods with adaptive properties have been developed to stabilize an underwater vehicle in the presence of external disturbances (Cui et al., 2016; Li and Lee, 2005; and Manzanilla et al., 2017). For the path following problem, a controller based on Lyapunov theory and Backstepping technique was designed by Lapierre et al. (2003). A robust controller based on sliding modes technique is presented by Elmokadem et al. (2016), which can ensure finite time convergence of the AUV to the desired path even in the presence of bounded perturbations. However, both works validated the proposed controller only via simulations.

Finally, many controllers have been developed to address the trajectory tracking problem. Sahu and Subudhi (2014) designed an adaptable controller capable of estimating parametric perturbations and uncertainties. Li et al. (2015) designed an adaptive fuzzy PID controller to follow straight lines that are commonly used in underwater reconnaissance missions. Guerrero et al., (2018) designed a robust algorithm based on the second order sliding mode technique with a self-adjusting gain proposed by Gonzalez et al. (2011), which is applied

to a Linear Time Invariant (LTI) system. The authors designed a trajectory tracking controller based on the Generalized Super-Twisting Algorithm (GSTA) with self-adjusting gains for a MIMO system. They demonstrated the robustness of the controller to external disturbances and parametric uncertainties through real-time experiments.

2.3 Workspace Dimension of the Operating Environment

The dynamics of underwater vehicles involve nonlinear and coupled hydrodynamic parameters, making it difficult to implement a controller for a six DOF AUV operating in a three-dimensional space. Therefore, many controller designs considered a simplified 3 DOF AUV model operating in a two-dimensional space. Typically, the roll DOF in most underwater vehicle is passively stable. Neglecting the stable roll DOF and considering the vehicle symmetry, the AUV model can be simplified into two decoupled subsystems in the longitudinal (vertical) and lateral (horizontal) planes (Subudhi et al., 2013; Healey and Lienard1993). The longitudinal model considers the AUV using the surge, heave, and pitch to move in two-dimensional vertical plane (Cristi et al., 1990; You & Chai, 1998; Silvestre 2000; Silvestre et al., 2009; Subudhi et al., 2013). Likewise, the lateral model considers the AUV using the surge and yaw to move in a two-dimensional horizontal plane.

Given that most path-planning algorithms were originally developed applied in autonomous ground vehicles operating in two-dimensional planar environments, they were later extended to path planning of simplified AUV models operating in longitudinal and lateral planes (Petres et al., 2007; Likhachev et al., 2005). However, path-planning algorithms for AUVs operating in three-dimensional spaces are beginning to appear (Yao & Zhao, 2018; Cao et al., 2009).

2.4 AUV Simulators

Many graphical simulators have been developed in the past (Cook et al., 2014; Ridao et al., 2004). Most of them used a kinematic model for the AUV. They can be broadly classified into offline simulators and online simulators. Offline simulators are mainly intended for design and preliminary testing using a synthetic AUV before implementation in a physical AUV. Also, there is no interaction between the simulator and the physical AUV. In some cases, the simulation load is so heavy that one second of simulation last for more than one second in the reality. In other cases, one second of simulation doesn't last for one second of the reality. In both cases, the temporal properties of the implemented algorithms are not considered (Ridao et al., 2004). Several MATLAB/SIMULINK based offline simulators developed in the past for different applications: Comparison of different control architectures for AUVs (Carreras et al., 2000), Simulation of a system for fault diagnosis and recovery applied to ROVs (Omerdic et al., 2003). There also exist more high-fidelity three-dimensional dynamic simulators for AUVs such as SUBSIM and DEEPWORKS (Matsebe et al., 2008). Online simulators allow real-time interaction with a physical AUV, ensuring time consistency between the simulated and the real time. Hence, the time properties of the simulated algorithm are taken into account within the simulation. NEPTUNE is a multi-vehicle, real-time, graphical simulator based on OpenGL that allows hardware in the loop simulations. Although there exist several commercial AUV simulators like UWSim, MORSE, and Gazebo, it may be more appropriate to build a simulator customized for individual research goals (Cook et al., 2014). A notable example is the development of an intelligent AUV simulator by Chen et al. (2013).

CHAPTER 3

DEVELOPMENT OF AUV MODELING AND SIMULATION PLATFORM

3.1 Modeling Approach

The approach used to develop the modeling and simulation of the AUV is first illustrated using the example of a simple one-dimensional Parachutist example. Imagine a parachutist starts falling from a certain height above ground. When the parachute is deployed, there will be two external forces acting upon the parachutist in the Z-dimension—gravitational force G $(= mg)$ acting downward and the drag D $(= cv)$ created by the parachute, acting upward.

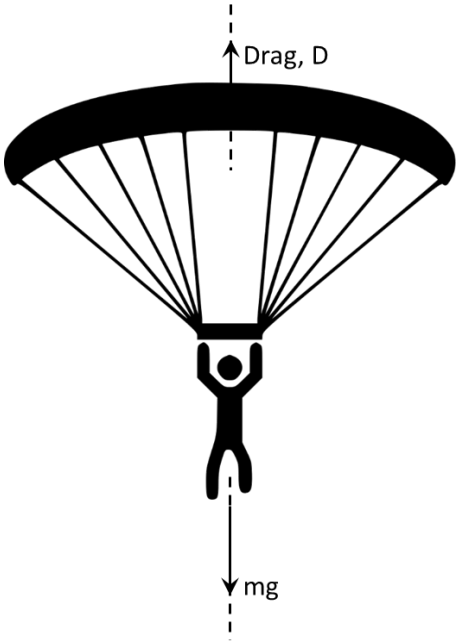

Figure 2: Free body diagram of a one-dimensional parachutist system used to illustrate the modeling approach

The equation of motion for the parachutist system can be written using the Newton's law of motion:

$$\sum F = ma \tag{1}$$

$$ma = mg - cv \tag{2}$$

The equation of motion can rewrite in the form of a differential equation as follows:

$$\frac{dv}{dt} = g - \left(\frac{c}{m}\right)v \tag{3}$$

As a next step, the equation of motion is translated into a state space formulation. For this purpose, the state of the parachutist system is defined by its position x and velocity v and representing them as state variables x_1 and x_2 respectively.

$$x_1 - \text{position } x$$

$$x_2 - \text{velocity } v$$

The state space formulation will then be derived as follows

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \left(\frac{c}{m}\right)x_2 + g \end{bmatrix} \tag{4}$$

A similar procedure can be used derive the state space formulation of the AUV.

3.2 Dynamic Model of an AUV

The AUV is assumed to have 6 degrees of freedom. The six-dimensional posture of the AUV in global coordinate frame is represented by $\eta = [x \quad y \quad z \quad \theta \quad \phi \quad \psi]^T$. The velocity of the AUV in local body frame is represented by $v = [u \quad v \quad w \quad p \quad q \quad r]^T$. The motion model of the AUV is represented as follows:

- Surge: translational motion along X-axis (North)
- Sway: translational motion along Y-axis (East)

- Heave: translational motion along Z-axis (Down)
- Roll: rotational motion about X-axis
- Pitch: rotational motion about Y-axis
- Yaw: rotational motion about Z-axis

The actuation forces and torques are represented as follows:

- The force along the X-axis is denoted by F_x
- The force along the Y-axis is denoted by F_y.
- The force along the Z-axis is denoted by F_z.
- The torque about X-axis is τ_x
- The torque about the Y-axis is τ_y
- The torque about the Z-axis is τ_z

$$\tau = \begin{bmatrix} F_x & F_y & F_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T \quad (5)$$

Based on the modeling approach as discussed in the previous section, first we need to consider the forces acting on the AUV with the help of a free body diagram.

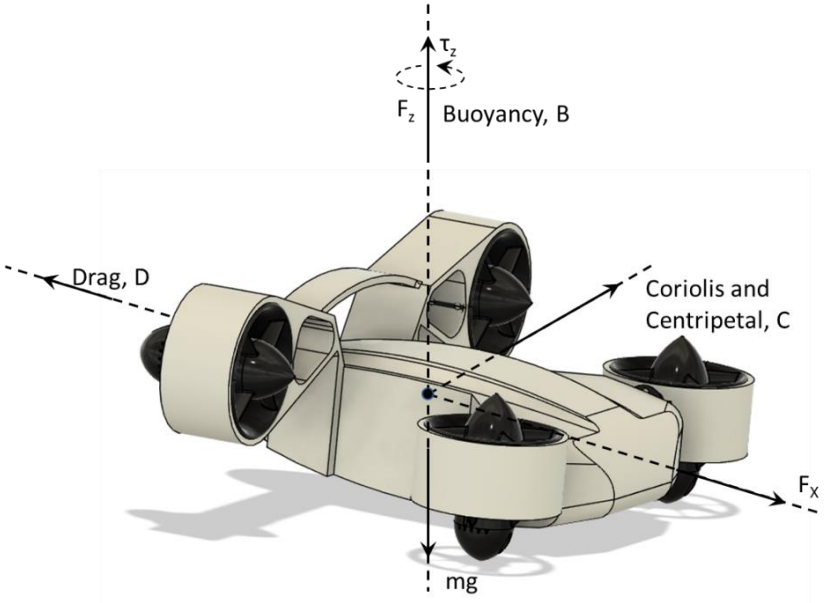

Figure 3: Free body diagram showing the external forces and actuation forces/torques acting on the AUV

The forces acting on the AUV include the inertial force, Coriolis and centripetal force, hydro-dynamic drag, gravitational force, and buoyancy force. The dynamic model of an AUV is derived from the equations of motion by considering all the forces mentioned in the free body diagram.

$$\tau = M\dot{v} + (C + D)v + g(\eta) \tag{6}$$

Where M is the inertia matrix, C is the Coriolis and centripetal matrix, D is the drag matrix, $g(\eta)$ is the gravitational and buoyancy matrix, and τ is the force/torque vector of the thruster input. Details on the procedure used to derive these matrices can be found in (Vervoot, 2008). The mass and inertia matrix consists of a rigid body mass M_{RB} and an added mass M_A, respectively.

$$M = M_A + M_{RB} \tag{7}$$

The rigid body mass, M_{RB} is defined by

$$\underline{M}_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_{xx} & -I_{xy} & -I_{xz} \\ mz_G & 0 & -mx_G & -I_{yx} & I_{yy} & -I_{yz} \\ -my_G & mx_G & 0 & -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$

The effect of the hydrodynamic added mass is modeled with the use of

$$\underline{M}_A = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix}$$

Coriolis and centripetal matrix C consist of a rigid body and an added mass term, respectively $C_{RB}(v)$ and $C_A(v)$,

$$C(v) = C_{RB}(v) + C_A(v) \tag{8}$$

The rigid body mass, $C_{RB}(v)$ is defined by

$$\underline{C}_{RB}(\nu) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & mw & -mv & 0 & I_{zz}r & -I_{yy}q \\ -mw & 0 & mu & -I_{zz}r & 0 & I_{xx}p \\ mv & -mu & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix}$$

Note that it is assumed that the B-frame is positioned at the center of gravity. The hydrodynamic added mass Coriolis-like matrix is given by

$$\underline{C}_A(\nu) = \begin{bmatrix} 0 & 0 & 0 & 0 & -\alpha_3(\nu) & \alpha_2(\nu) \\ 0 & 0 & 0 & \alpha_3(\nu) & 0 & -\alpha_1(\nu) \\ 0 & 0 & 0 & -\alpha_2(\nu) & \alpha_1(\nu) & 0 \\ 0 & -\alpha_3(\nu) & \alpha_2(\nu) & 0 & -\beta_3(\nu) & \beta_2(\nu) \\ \alpha_3(\nu) & 0 & -\alpha_1(\nu) & \beta_3(\nu) & 0 & -\beta_1(\nu) \\ -\alpha_2(\nu) & \alpha_1(\nu) & 0 & -\beta_2(\nu) & \beta_1(\nu) & 0 \end{bmatrix}$$

The hydrodynamic damping of underwater vehicles normally contains the drag force. The drag forces can be separated into a linear and a quadratic term, where $D_q(v)$ and $D_l(v)$ are the quadratic and the linear drag term, respectively.

$$\mathrm{D}(v) = D_q(v) + D_l(v) \tag{9}$$

$$D_l(v) = \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & 0 & 0 & K_q & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r \end{bmatrix}$$

The matrix notation of $D_q(v)$ quadratic drag matrix, is given by

$$D_q(v) = \begin{bmatrix} X_{u|u|}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{v|v|}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{w|w|}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{p|p|}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{q|q|}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{r|r|}|r| \end{bmatrix}$$

Where gravitational and buoyancy vector, $g(\eta)$, can be denoted in matrix form by

$$\underline{g}(\eta) = \begin{bmatrix} (W-B)\sin(\theta) \\ -(W-B)\cos(\theta)\sin(\phi) \\ -(W-B)\cos(\theta)\cos(\phi) \\ -(y_g W - y_b B)\cos(\theta)\cos(\phi) + (z_g W - z_b B)\cos(\theta)\sin(\phi) \\ (z_g W - z_b B)\sin(\theta) + (x_g W - x_b B)\cos(\theta)\cos(\phi) \\ -(x_g W - x_b B)\cos(\theta)\sin(\phi) - (y_g W - y_b B)\sin(\theta) \end{bmatrix}$$

The AUV's center of gravity is denoted by r_G, and the center of buoyancy is denoted by r_B.

$$r_G = [x_G \quad y_G \quad z_G]^T \tag{10}$$

$$r_B = [x_B \quad y_B \quad z_B]^T \tag{11}$$

3.3 State Space Formulation of the AUV

The equations of motion presented in the previous section are translated into a state space formulation using the procedure described for the parachutist system in section 3.1. The state of the AUV is defined by its 3D position, 3D orientation, 3D linear velocity, and 3D angular velocity. This results in a 12-state space model of the AUV as shown in the table below.

State Variables	**State**	**Symbol**
x_1	Position in X-axis, North	x
x_2	Position in Y-axis, East	y
x_3	Position in Z-axis, Down	z
x_4	Orientation about X-axis, Roll	θ
x_5	Orientation about Y-axis, Pitch	ϕ
x_6	Orientation about Z-axis, Yaw	ψ
x_7	Linear velocity in X-direction, North	u
x_8	Linear velocity in Y-direction, East	v
x_9	Linear velocity in Z-direction, Down	w
x_{10}	Angular velocity about X-axis, Roll rate	p
x_{11}	Angular velocity about Y-axis, Pitch rate	q
x_{12}	Angular velocity about Z-axis, Yaw rate	r

Table 1: State variables and the corresponding AUV's three-dimensional position, orientation, and velocity variables

The state space equations are represented in a matrix form as below:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix}_{12x1} = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ \left[M^{-1}\left(\tau - (C+D)v - g(\eta)\right) \right]_{6x1} \end{bmatrix}_{12x1} \quad (12)$$

The AUV parameters are as follows:

Parameter	m	ρ	∇	l_1	x_b	y_b	z_b	x_g	y_g	z_g
Value	161 kg	998 kg/m^3	151 L	400 mm	0 mm	0 mm	0 mm	0 mm	0 mm	0 mm

Table 2: Inertial parameters of the AUV

Parameter	x_u	y_v	z_w	N_r	x_{uu}	y_{vv}	z_{ww}	N_{rr}	I_{xx}	I_{yy}	I_{zz}
Value	60 Ns/m	60 Ns/m	100 Ns/m	20 Ns/m	90 Ns/m	90 Ns/m	150 Ns/m	15 Ns/m	15 Kgm^2	15 Kgm^2	15 Kgm^2

Table 3: Drag parameters and matrix variables

States	x_0	y_0	z_0	θ_0	ϕ_0	ψ_0	u_0	v_0	w_0	p_0	q_0	r_0
Value	0	0	-3	0	0	0	0	0	0	0	0	0

Table 4: Initial state of the AUV

3.4 Modeling in Simulink

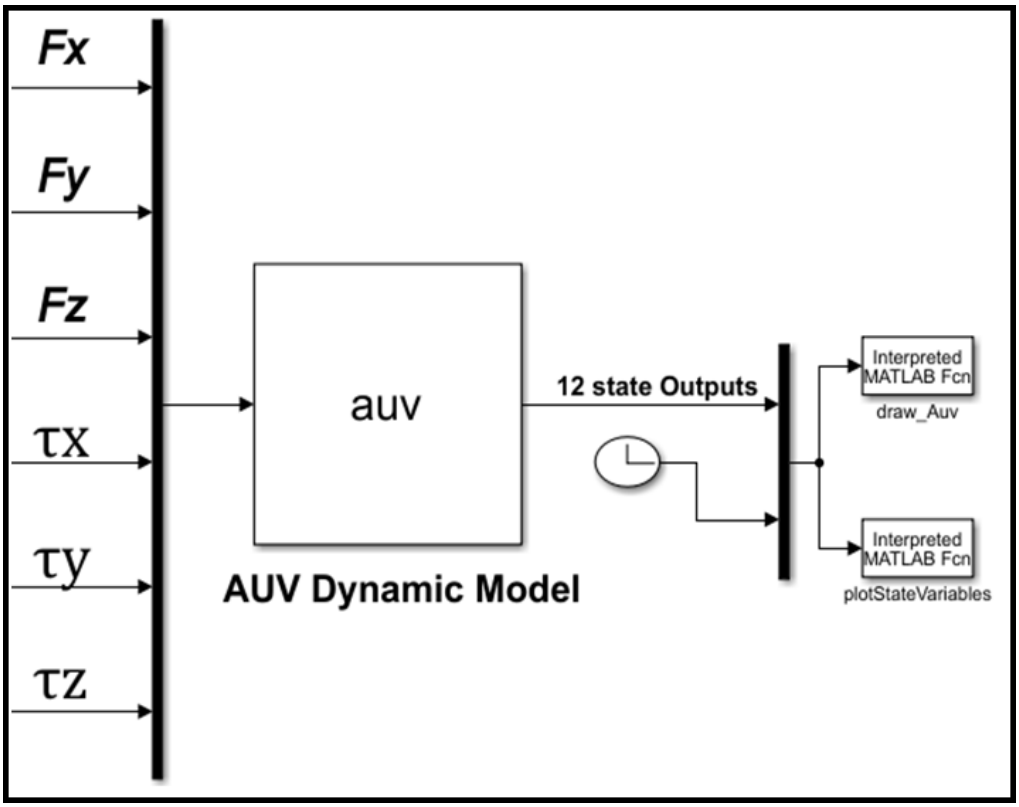

Figure 4: Simulink model of the synthetic AUV. The AUV dynamic model takes in six actuation forces and outputs 12 states that are fed into the animation block and plotting block.

The S-function block in Simulink, along with the corresponding S-function template in MATLAB, offers a convenient way to evolve the states of a dynamical system from initial conditions by iteratively integrating the state derivatives of the system. The S-function can be configured by listing the number of inputs, number of states, number of outputs, specifying the initial state, writing the equations for the state derivatives of the system, defining the outputs, and specifying the sampling time. The S-function block (**auv**) interacts with the MATLAB function **auv.m** (see Fig. 6) that codes the state space model of the AUV dynamics.

The position and orientation outputs from the S-function block are fed as inputs to an animation code to achieve a three-dimensional visualization of the resulting AUV motion. A CAD model of the physical AUV is designed and imported into the animation to create a realistic motion of the synthetic AUV in a three-dimensional space.

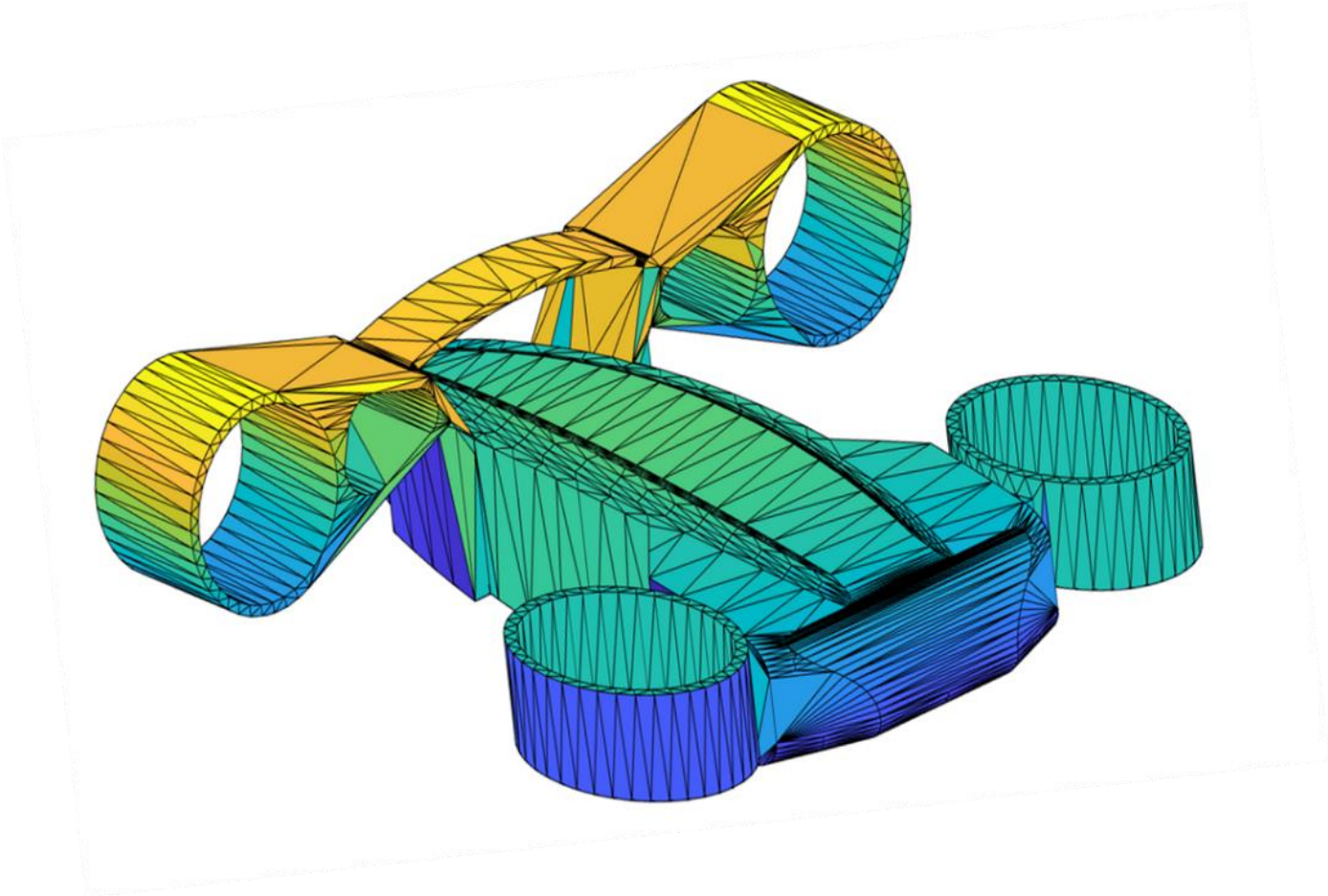

Figure 5: 3D rendering used for visualization of the AUV in the animation environment

```
function [sys,x0,str,ts,simStateCompliance] = auv(t,x,u,flag,P)
```

```
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes(P)

sizes = simsizes;
sizes.NumContStates  = 12;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 12;
sizes.NumInputs      = 6;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;   % at least one sample time is needed

sys = simsizes(sizes);

x0  =P.x0;
```

Initialization

```
function sys=mdlDerivatives(t,x,u,P)
 fx = u(1);%% Input forces
 fy = u(2);
 fz = u(3);
%F=u(1);
tauPhi=u(4);%% Input Torques
tauTheta=u(5);
tauPsi = u(6);
```

Inputs

```
function sys=mdlOutputs(t,x,u,P)

sys = x;
```

Outputs

```
xdot=zeros(12,1);
xdot(1:3,1)=R_v_b*[ub;vb;wb];
xdot(4:6,1)=[...
    1 sin(phi)*tan(theta) cos(phi)*tan(theta);...
    0 cos(phi) -sin(phi);...
    0 sin(phi)/cos(theta) cos(phi)/cos(theta)]*[p;q;r];
xdot(7:12,1)=M^(-1)*(-C*V_b-D*V_b-g_eta+tau);
sys = xdot;
```

State derivatives

Figure 6: Snippets from the MATLAB S-function (**auv.m**) showing the coding of different steps in numerical integration of the state derivatives that capture the AUV system dynamics

3.5 Underactuated AUV Model

The actuation model of the synthetic AUV assumes a thrust force in the surge and heave directions and a torque about the yaw direction. Sway motion is achieved by a combination of

surge and yaw motions. The pitch and roll motions are considered as underactuated degrees of freedom. A cylindrical coordinate system is used to match the actuation model of the AUV. This allows the AUV to move to any position in a three-dimensional space by yawing about the vertical axis, surging along the radial axis, and then heaving along the vertical axis.

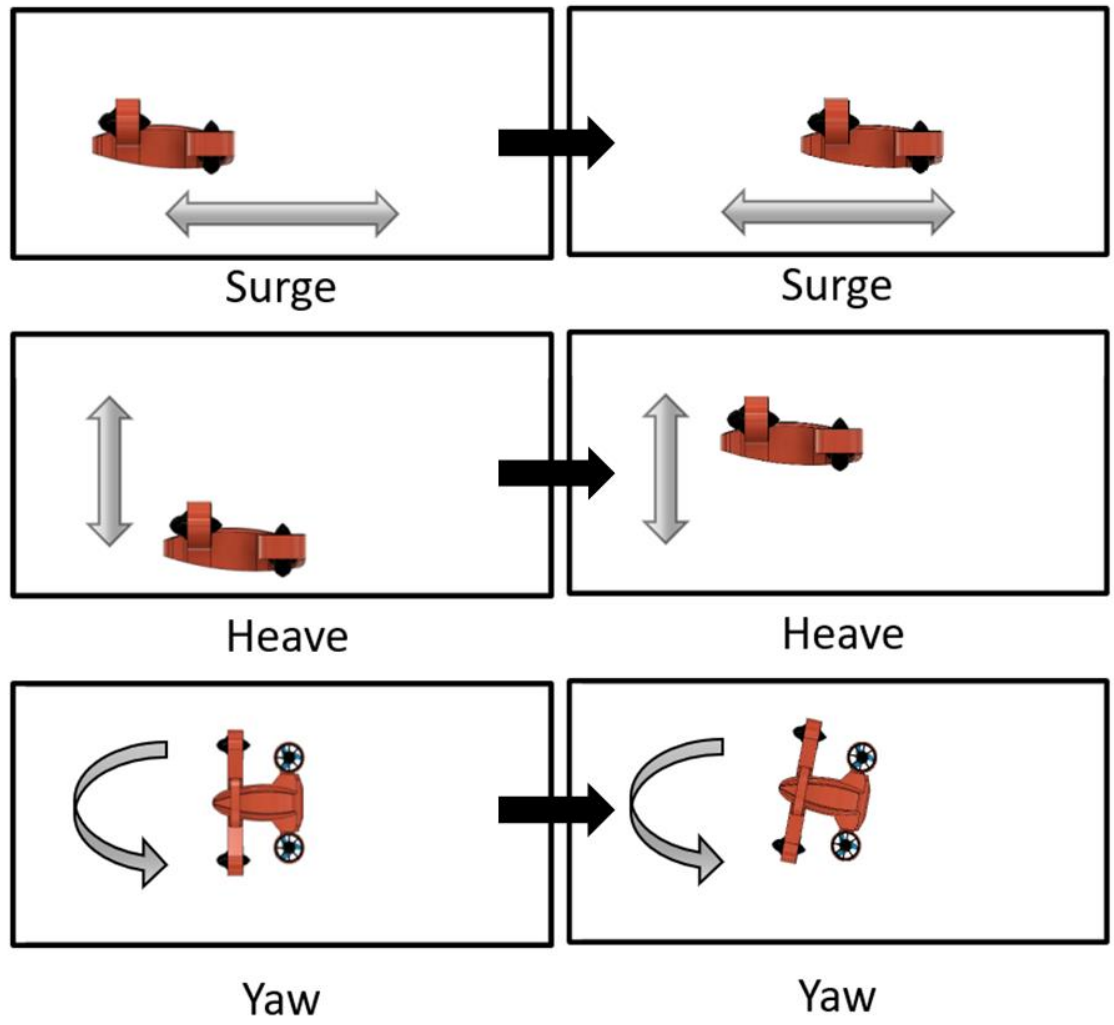

Figure 7: Illustration of the actuated degrees of freedom (Surge, Heave, and Yaw) of the underactuated AUV

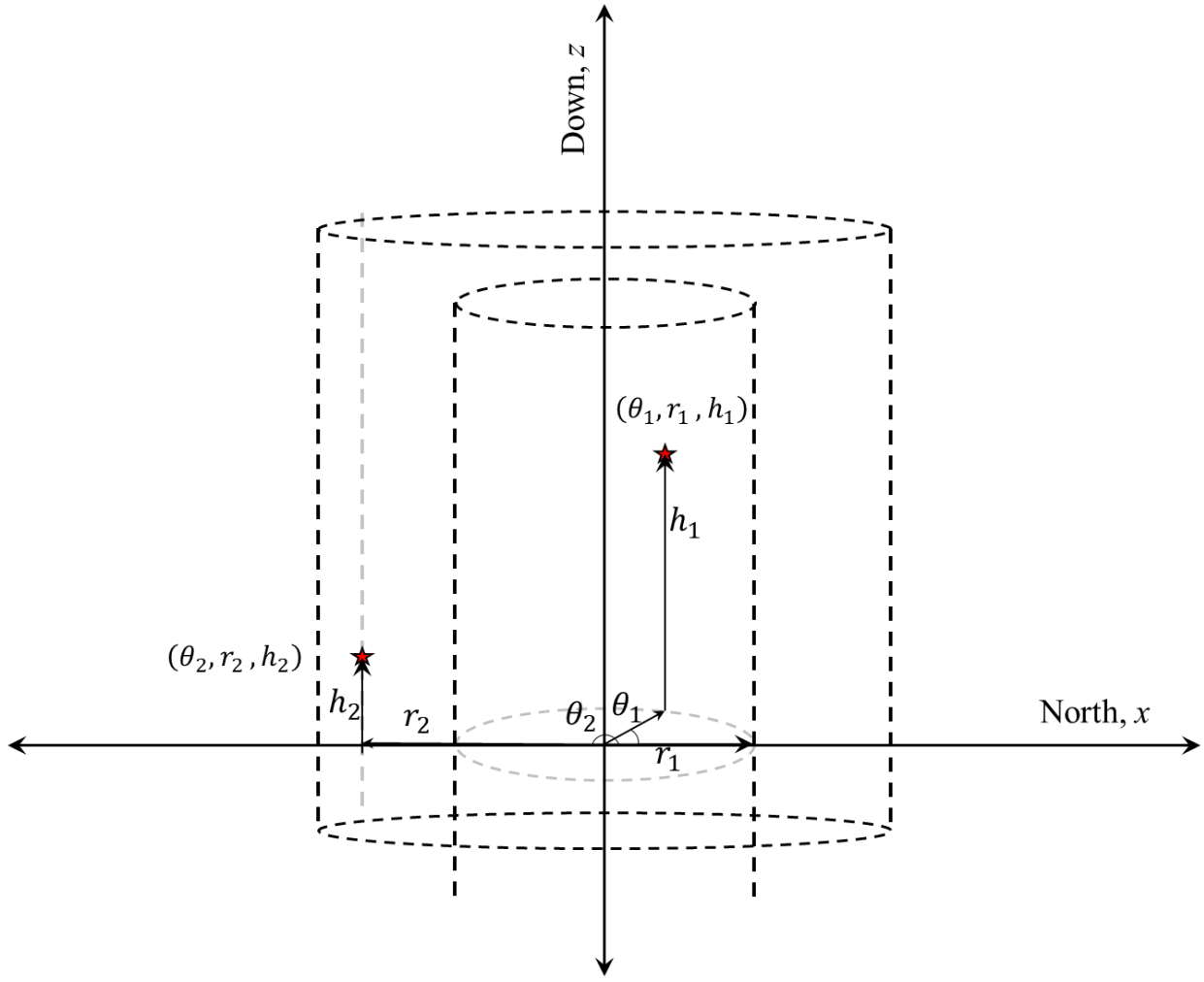

Figure 8: Cylindrical coordinate system used to match the actuation model of the AUV.

3.6 Closed-Loop Controller Implementation

A basic Proportional-Derivative (PD) controller is implemented that enables the AUV to move and settle at any desired set position. The controller gains are tuned to achieve a critically damped response in the AUV. One problem was observed during the controller implementation—the AUV started wobbling a brief period after it reached the desired position specified by the controller. This wobbly behavior was attributed to torque disturbances about

roll and pitch axis caused due to offset between the centers of buoyancy and gravity. This offset was removed to resolve this unstable behavior.

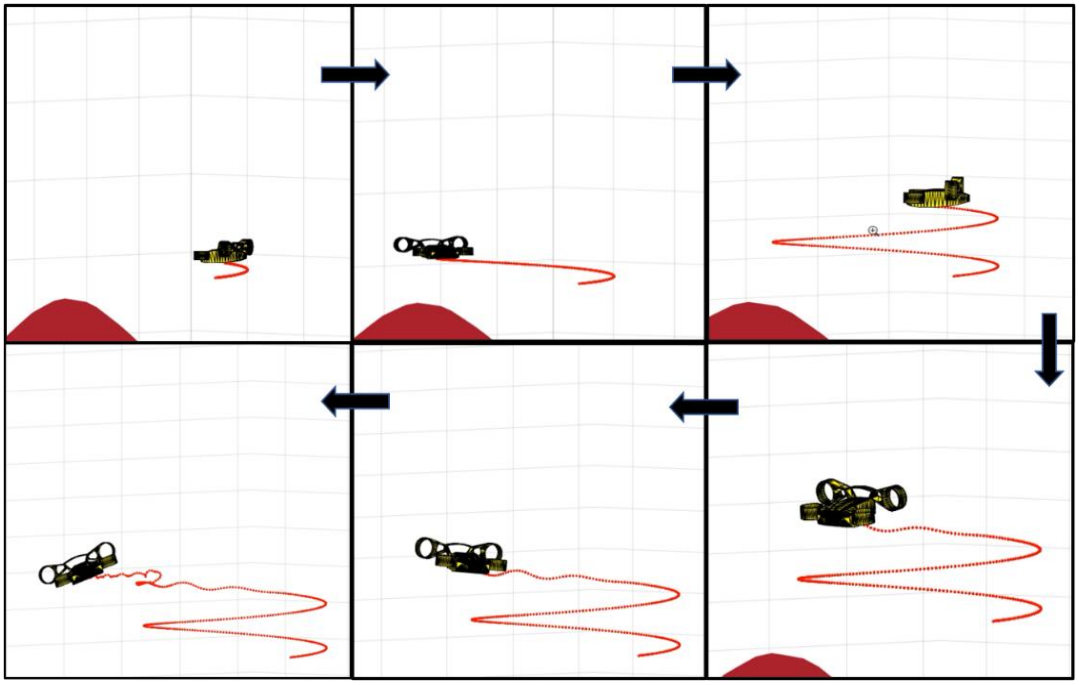

Figure 9: Unstable wobbly behavior of the AUV when it reaches the set point attributed to external torque disturbances about roll and pitch axis caused due to offset between the centers of buoyancy and gravity.

3.7 Augmentation with a Waypoint Planner

The AUV controller is augmented with a waypoint planner to allow the AUV to traverse arbitrary paths in two- and three- dimensional spaces. For this purpose, a memory block in Simulink was used that stored a global variable used to represent the index of the current waypoint being tracked by the AUV controller. As the AUV reached within a threshold proximity of this current waypoint, the index in the memory block was incremented, thereby updating the controller's setpoint to the next waypoint.

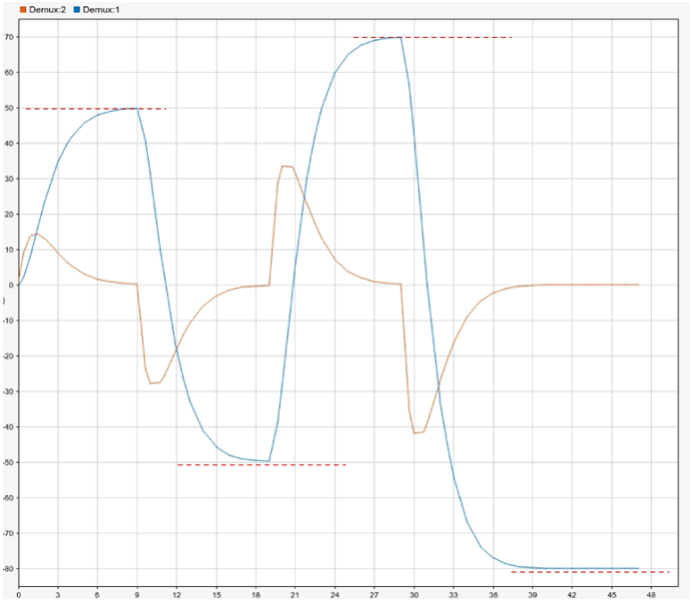

Figure 10: Waypoint planning of the one-dimensional parachutist system and tracking using a critically damped controller

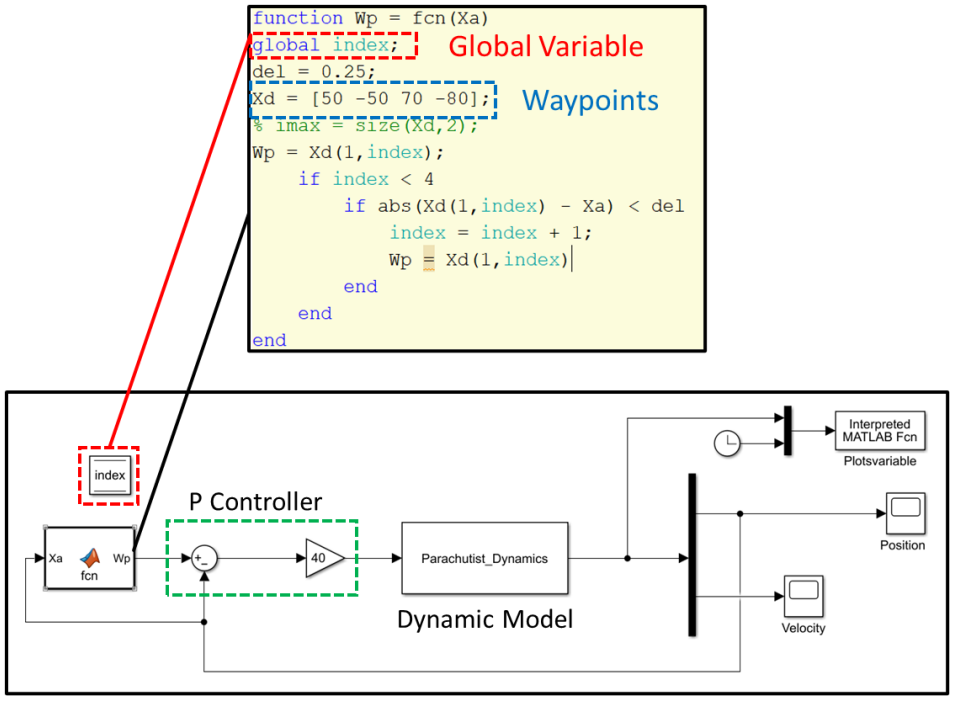

Figure 11: MATLAB & Simulink interface for parachutist system. A global variable is implemented in Simulink using a data storage memory block that enables the updation of the waypoint index whenever the system reaches the current waypoint within a user-defined threshold.

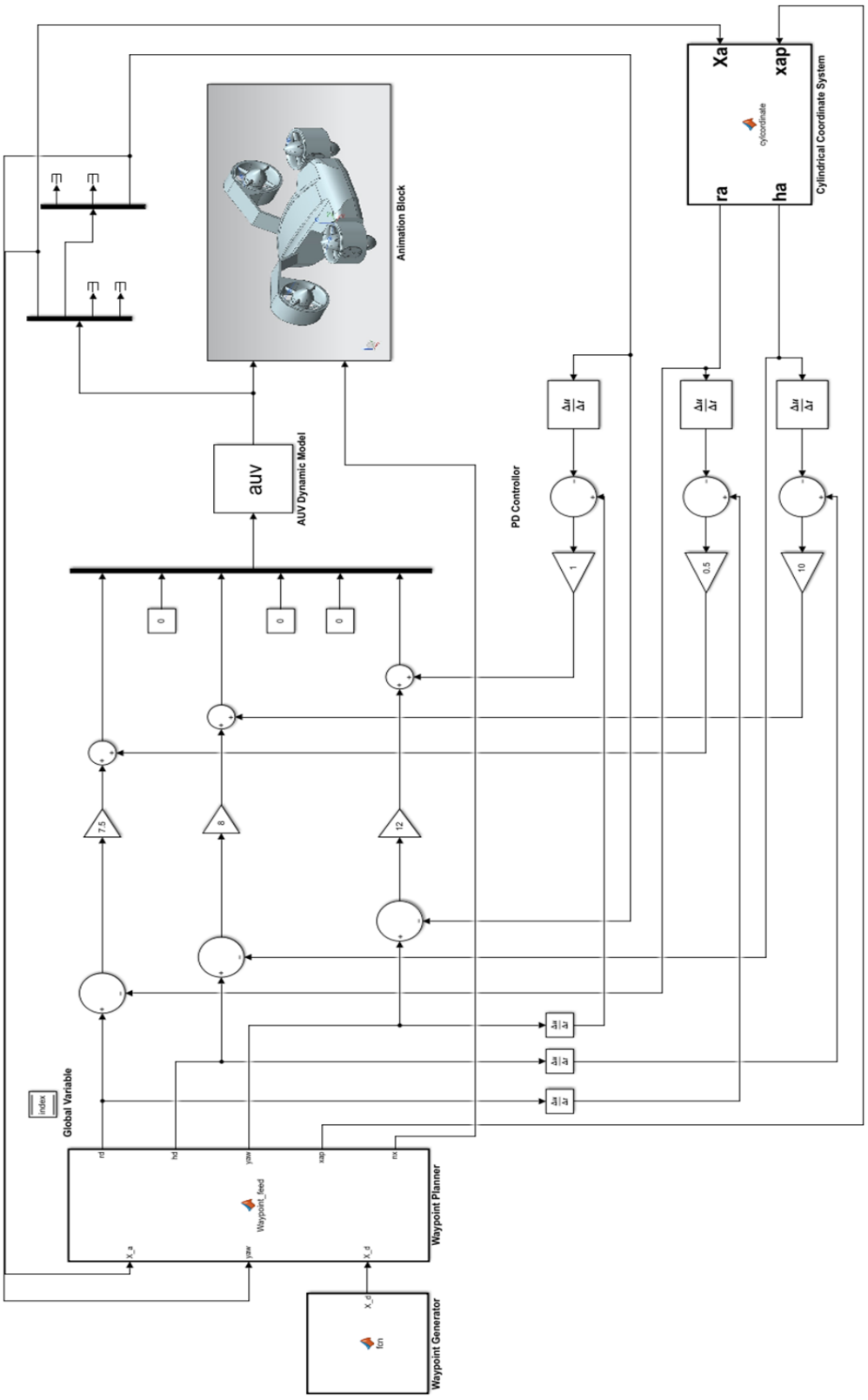

Figure 12: A full-blown Simulink Model of the AUV with integration between waypoint planner, controller, and animation modules

CHAPTER 4

EXPERIMENTAL VALIDATION

The simulator uses an underactuated AUV. Therefore, we want to analyze how the synthetic AUV will achieve the mission goals with only three actuated motions (i.e., Surge, Heave and Yaw). XYZ axes are named as north, east and down. The simulator has been tested in various user-defined environments to evaluate its capabilities and limitations:

1. Planar motion analysis (in XZ, XY, and YZ)
2. Waypoint-controller parametric analysis
3. A* path planner implementation
4. AUV velocity model

4.1 Planar Motion Analysis

4.1.1 XZ Plane:

The simulator has been tested first in XZ plane. Here, we will observe how a synthetic underactuated AUV can track a complex trajectory like a sine wave with multiple waypoints, by using just Heave and Surge simultaneously. As AUV is operated in XZ plane, the orientation about Z-axis is constrained.

Snapshots from an animation while AUV is tracking its desired trajectory is shown in the Figure 13. The behavior of the synthetic AUV with respect to the tracked waypoints, its position and velocity are shown in Figures 14 to 19.

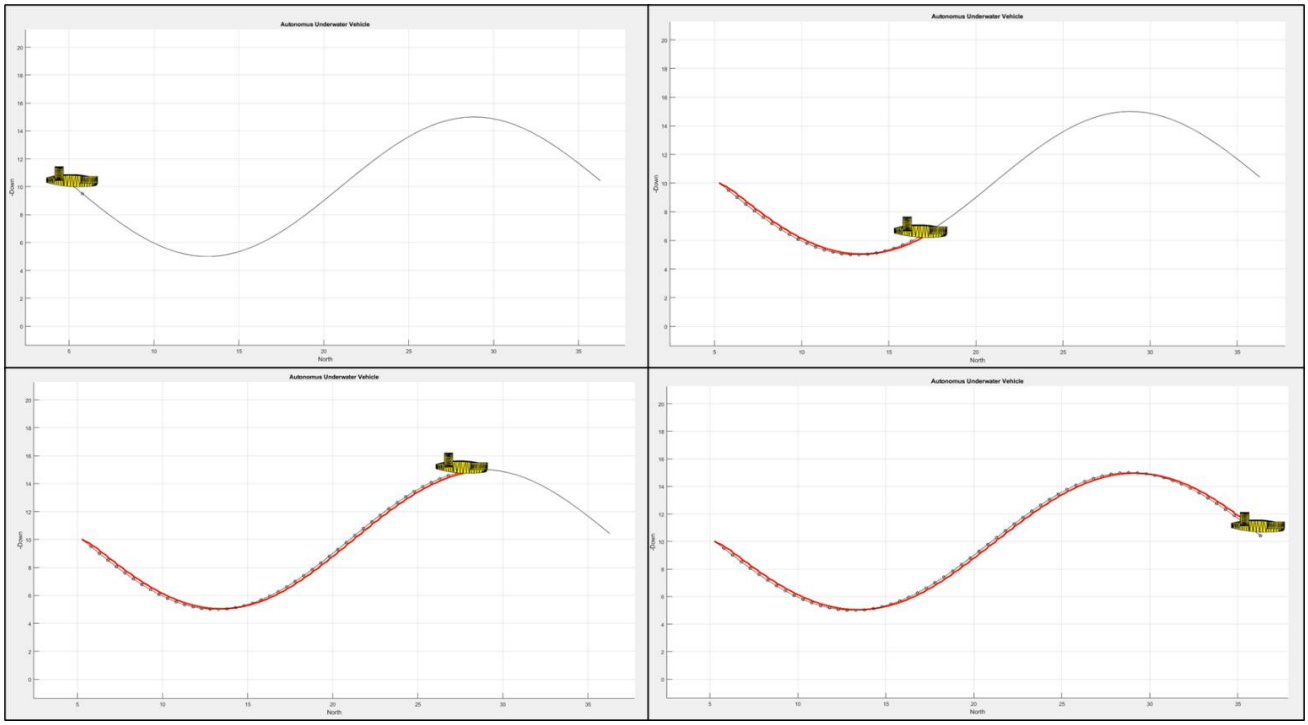

Figure 13: Snapshots from the AUV simulator visualization showing the AUV tracking a trajectory in XZ plane using surge and heave as actuation motions

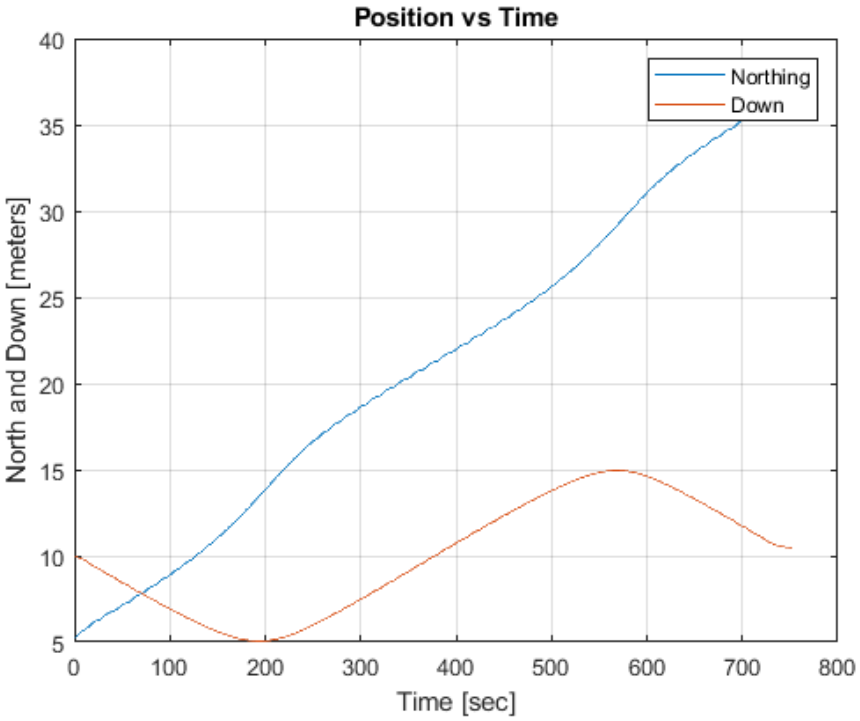

Figure 14: Plot of AUV's surge (along Northing axis) and heave (along Down axis) as a function of time

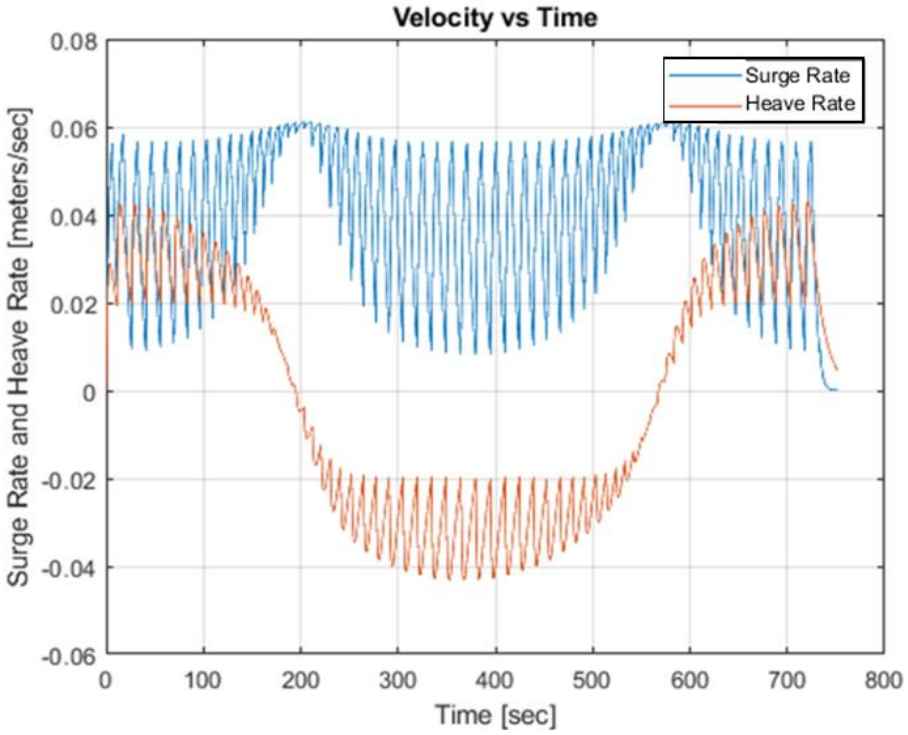

Figure 15: Plot of surge rate and heave rate of the AUV as a function of time

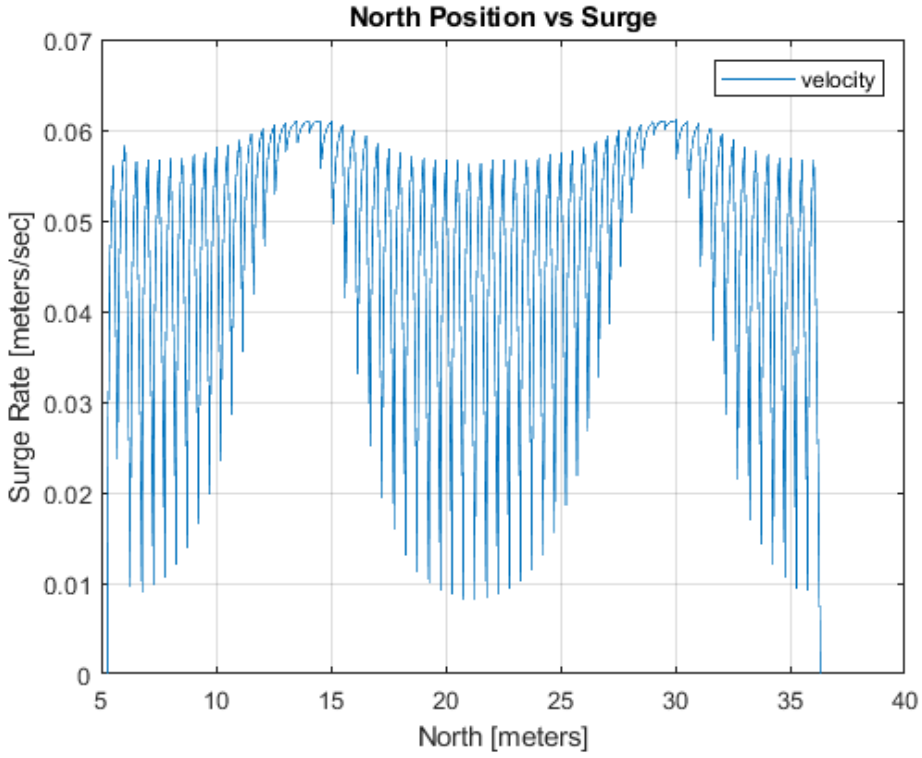

Figure 16: Plot of surge rate of the AUV as a function of position in north (X-axis)

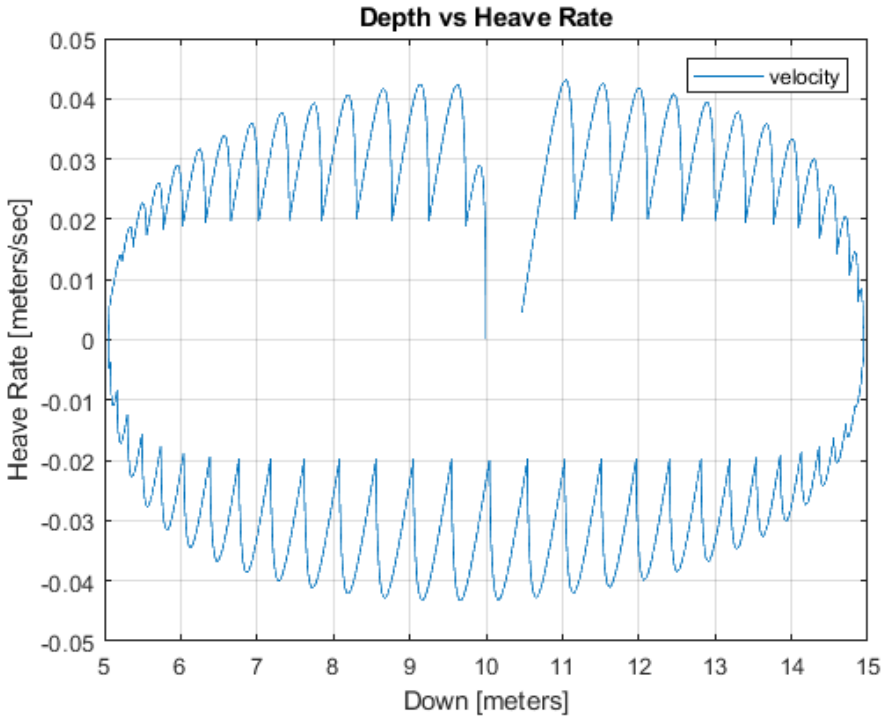

Figure 17: Plot of heave rate of the AUV as a function of position in down (Z-axis)

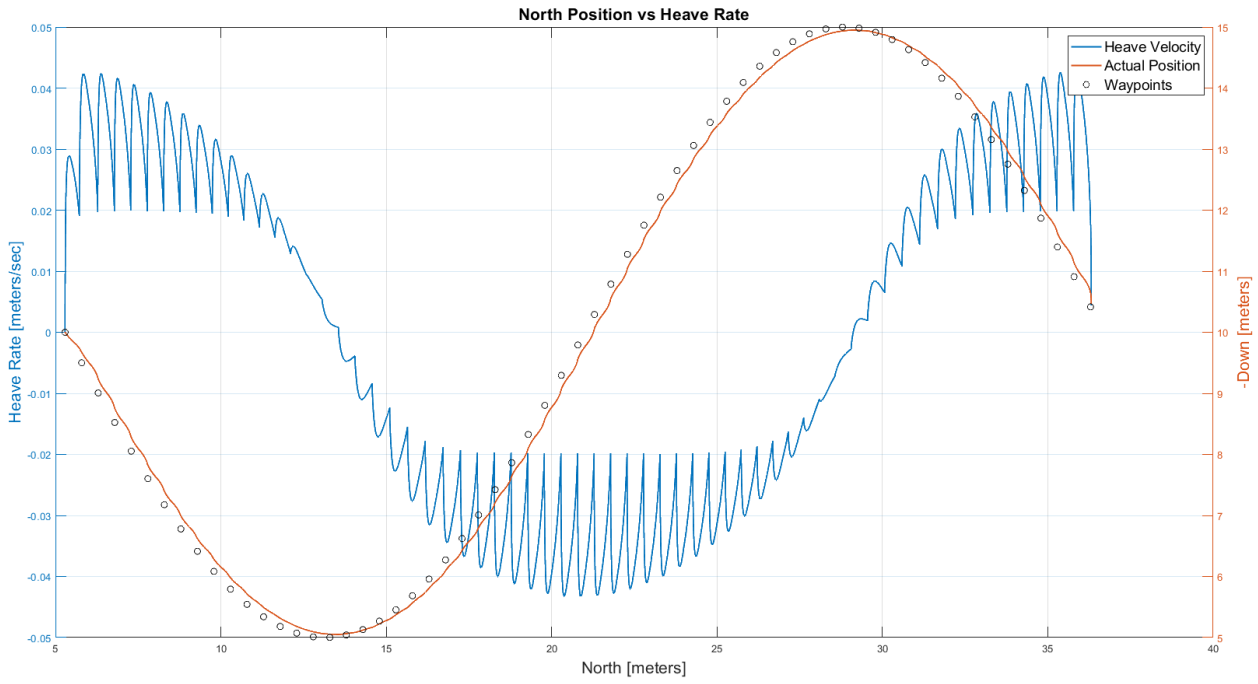

Figure 18: Analysis of the AUV behavior with heave rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

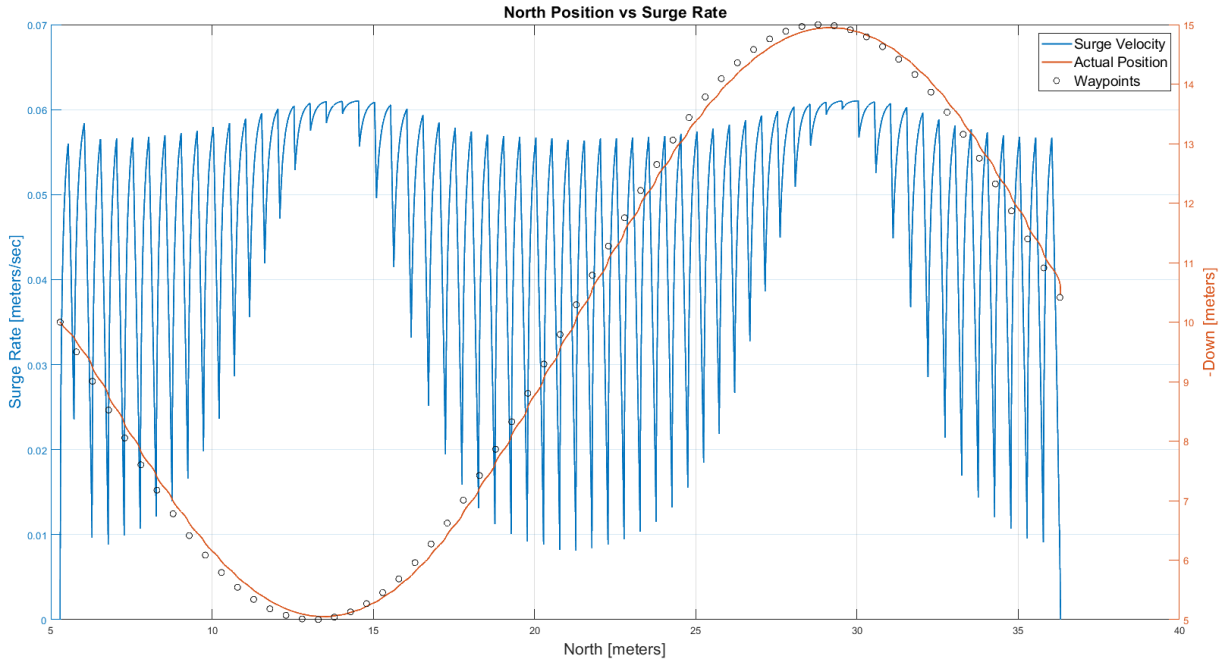

Figure 19: Analysis of the AUV behavior with surge rate as a function of waypoints by using two different scales in the vertical axis and plotted the actual path traversed by the AUV

4.1.2 XY Plane:

The simulator has been tested first in XY plane. Here, we will observe how a synthetic underactuated AUV can track a complex trajectory like a sine wave with multiple waypoints, by using just Surge and Yaw. As AUV is operated in XY plane, the motion (Heave) in Z-axis is constrained.

Snapshots from an animation while AUV is tracking its desired trajectory is shown in the Figure 20. The behavior of the synthetic AUV with respect to the tracked waypoints, its position and velocity are shown in Figures 21 to 26.

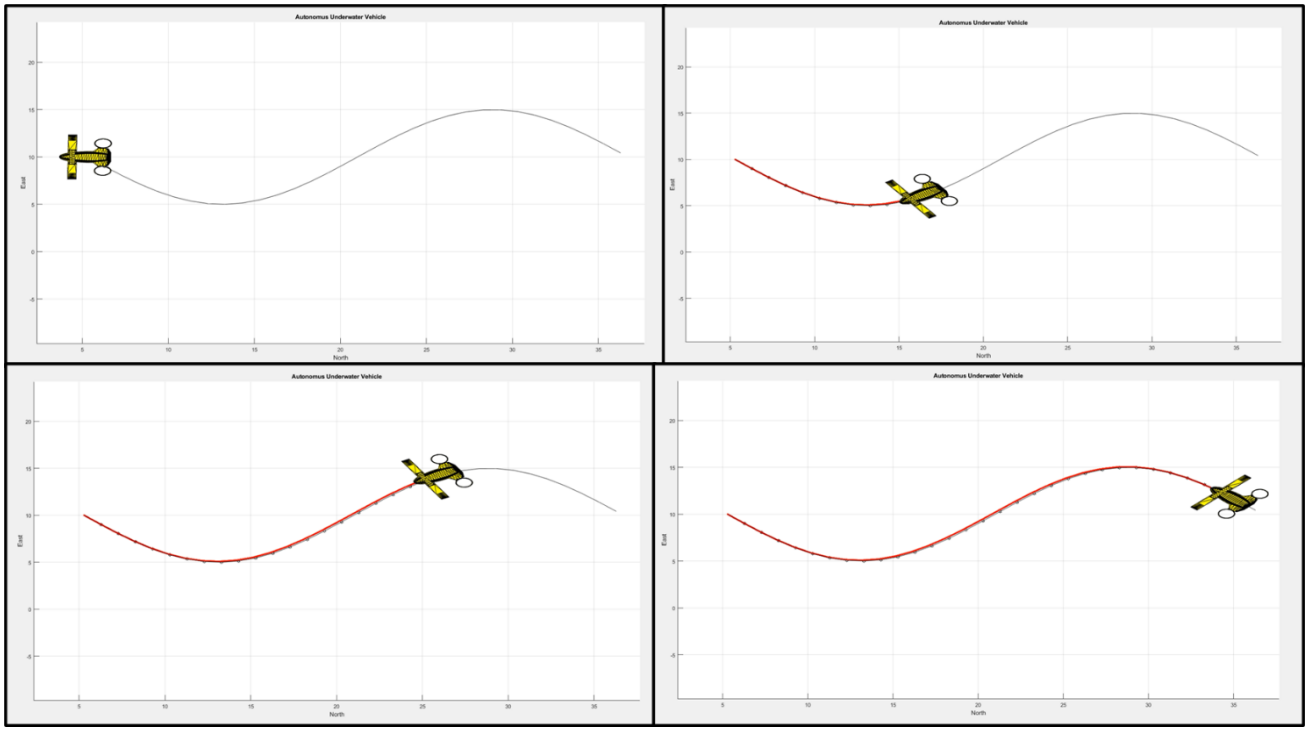

Figure 20: Snapshots from the AUV simulator visualization showing the AUV tracking a trajectory on XY plane using surge and yaw as actuation motions

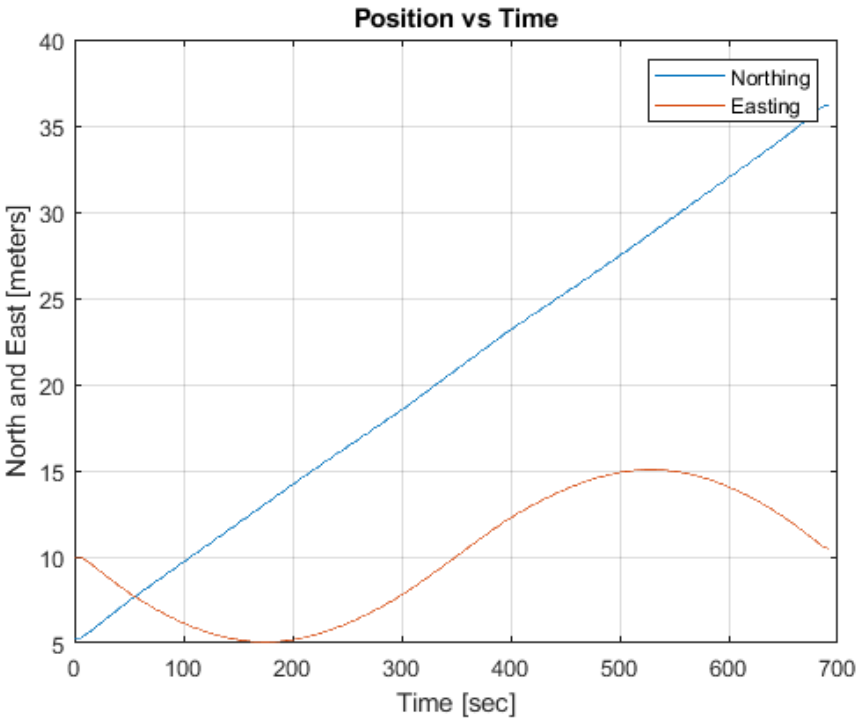

Figure 21: Plot of AUV's surge and yaw as a function of time

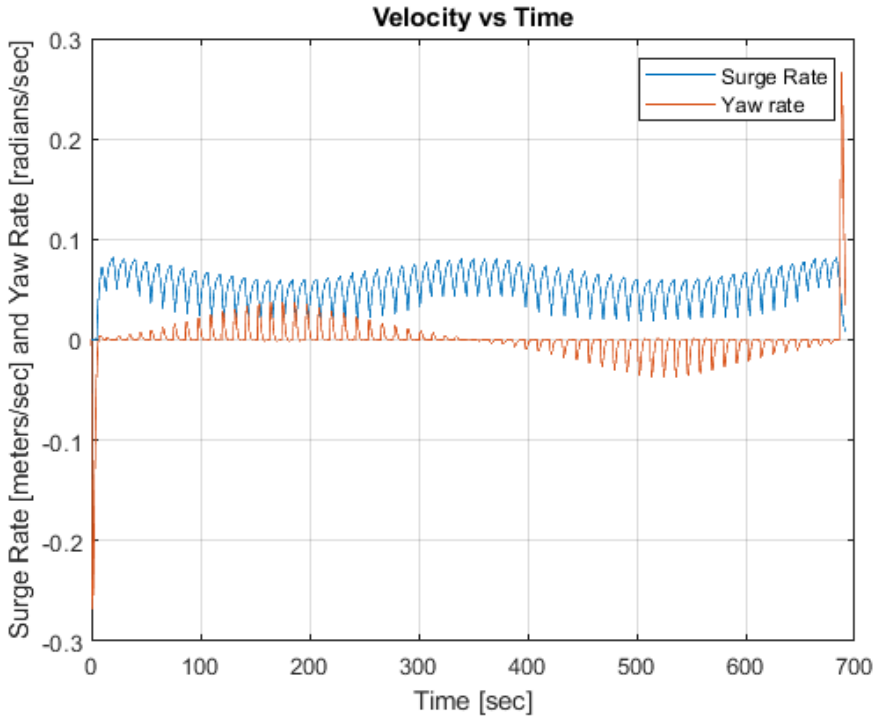

Figure 22: Plot of surge rate and yaw rate of the AUV as a function of time

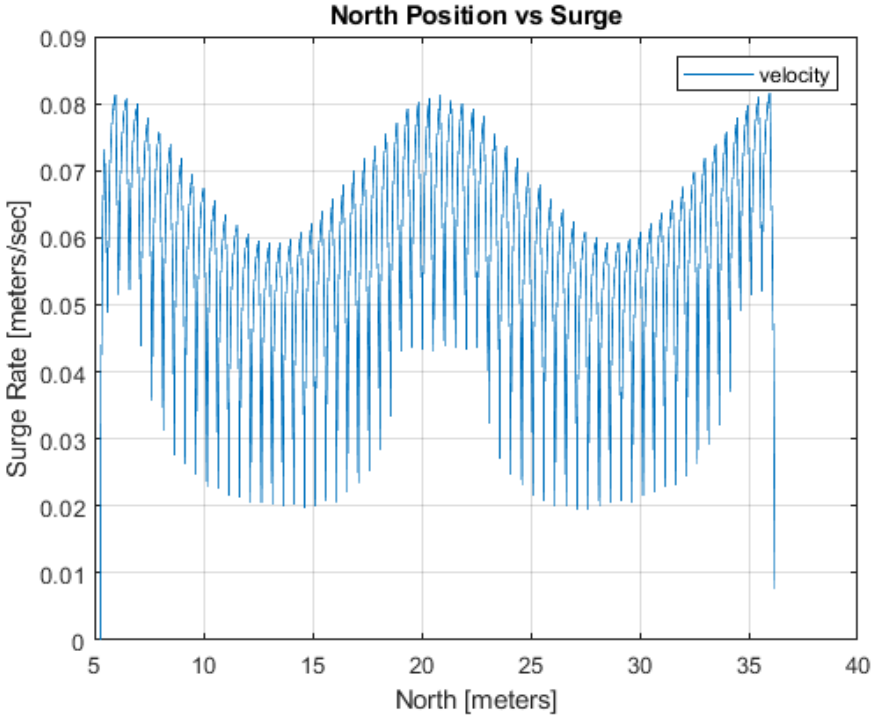

Figure 23 : Plot of surge rate of the AUV as a function of position in north (X-axis)

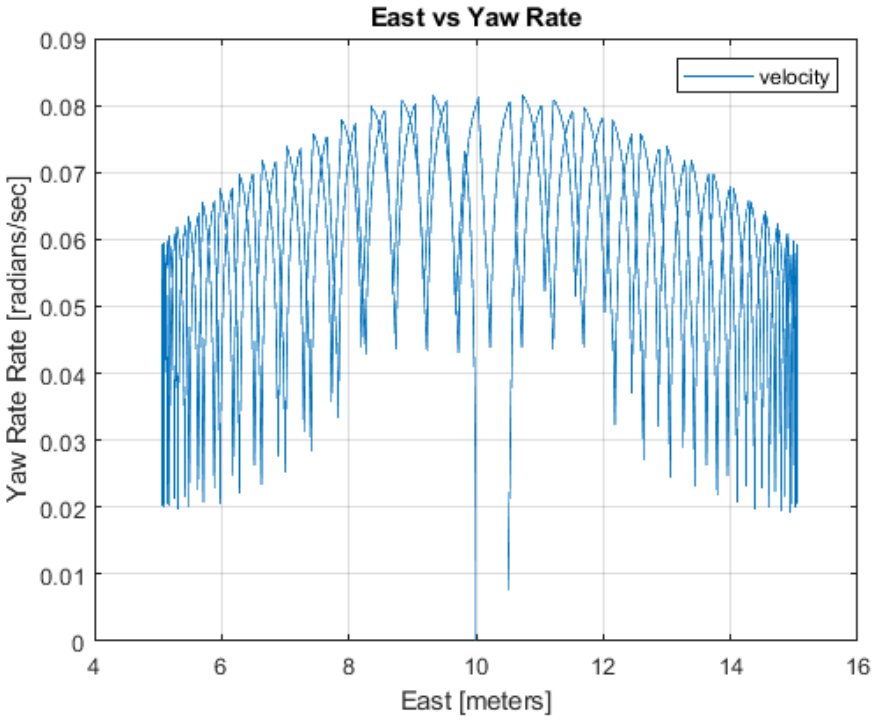

Figure 24 : Plot of yaw rate of the AUV as a function of position in east (Y-axis)

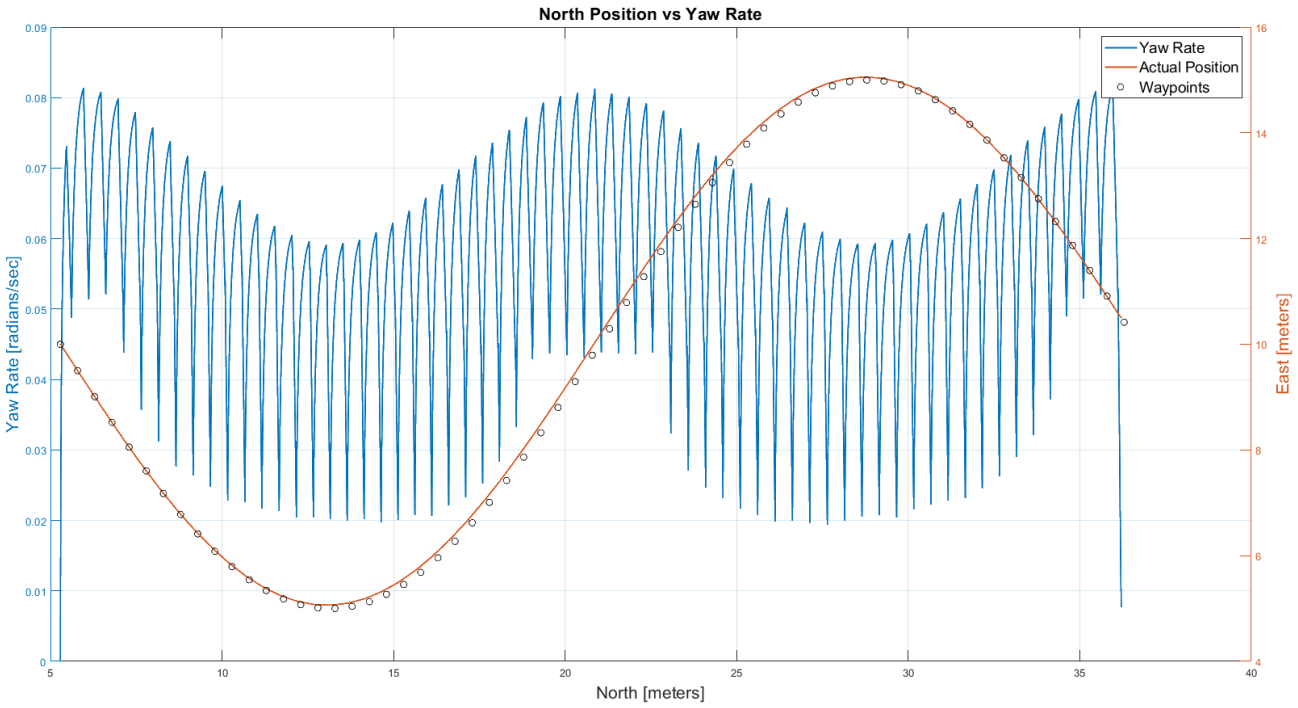

Figure 25: Analysis of the AUV behavior with yaw rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

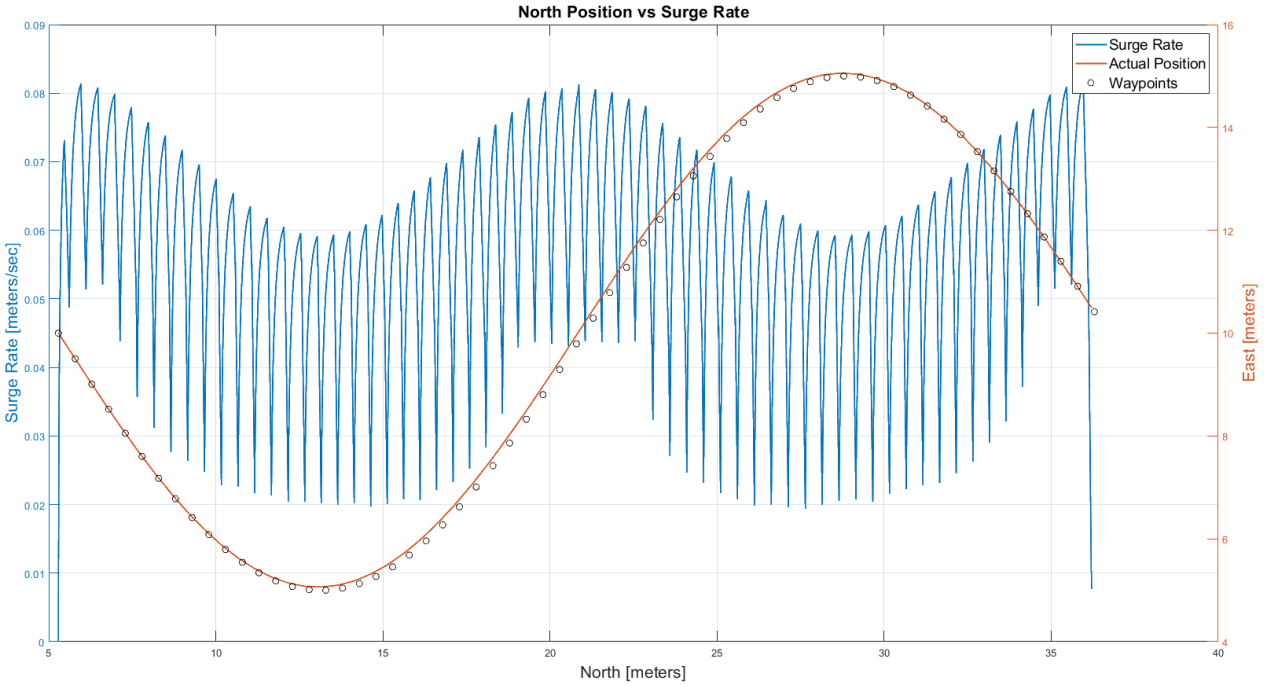

Figure 26: Analysis of the AUV behavior with surge rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

4.1.3 YZ Plane:

The simulator has been tested first in YZ plane. Here, we will observe how a synthetic underactuated AUV can track a complex trajectory like a sine wave with multiple waypoints using all the actuated motion (i.e., Surge, Heave and Yaw). As AUV is operated in YZ plane, no motion is constrained other than the underactuated motions (i.e., Roll, Pitch and Sway).

Snapshots from an animation while AUV is tracking its desired trajectory is shown in the Figure 27. The behavior of the synthetic AUV with respect to the tracked waypoints, its position and velocity are shown in Figures 28 to 33.

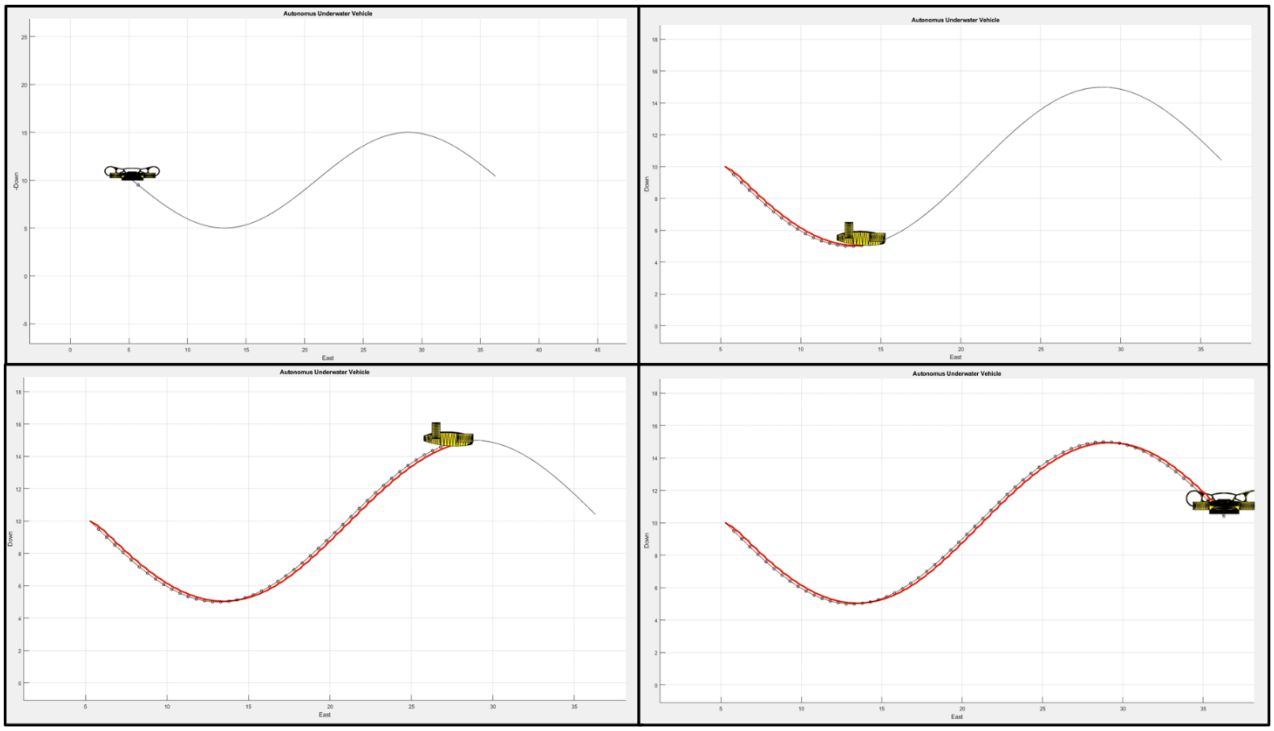

Figure 27 : Snapshots from the AUV simulator visualization showing the AUV tracking a trajectory on XZ plane using surge, yaw and heave as actuation motions

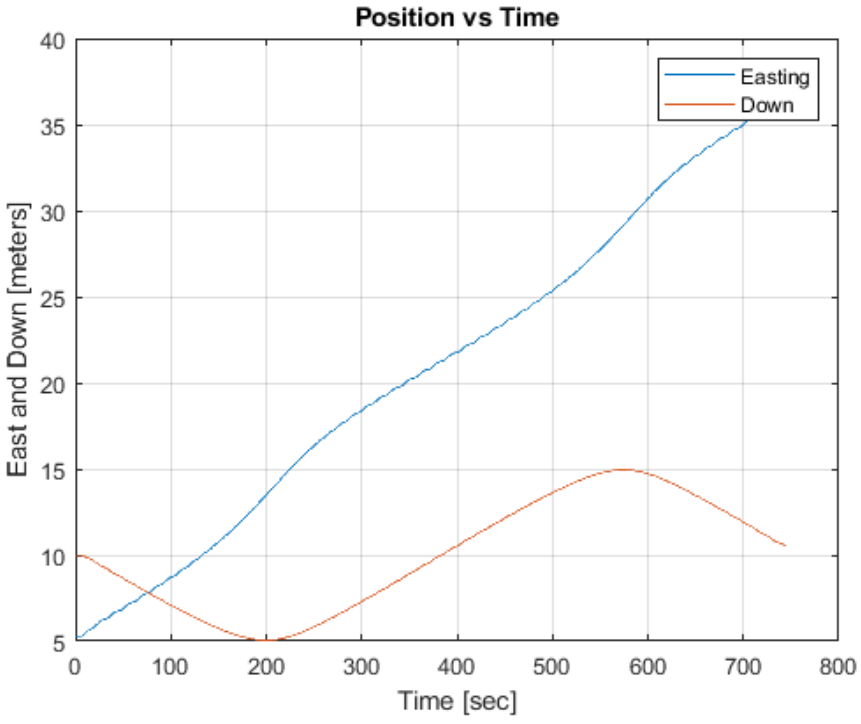

Figure 28: Plot of AUV's surge and heave as a function of time

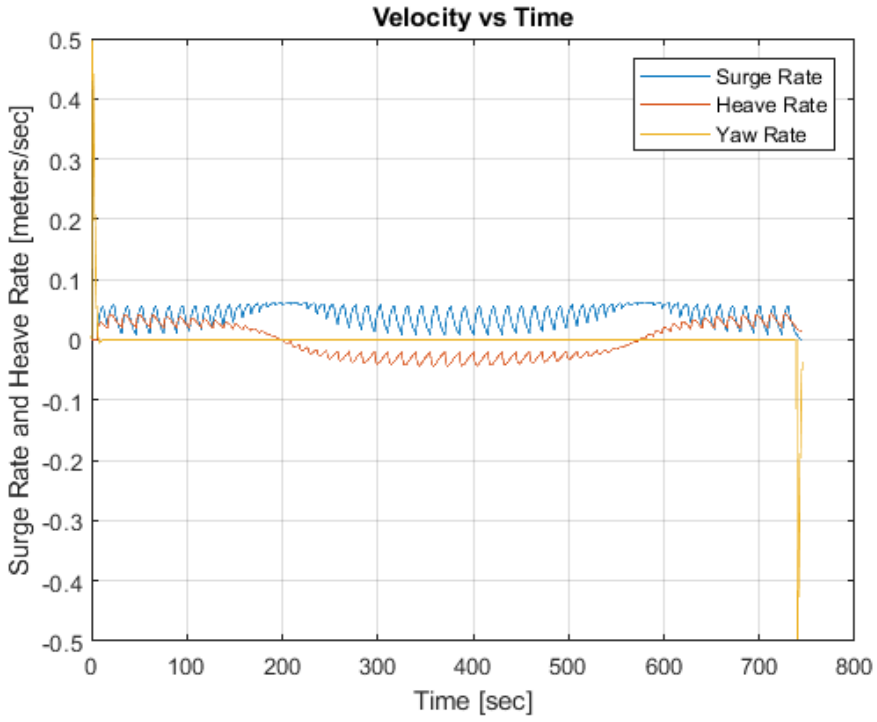

Figure 29: Plot of surge rate, yaw rate and heave rate of the AUV as a function of time

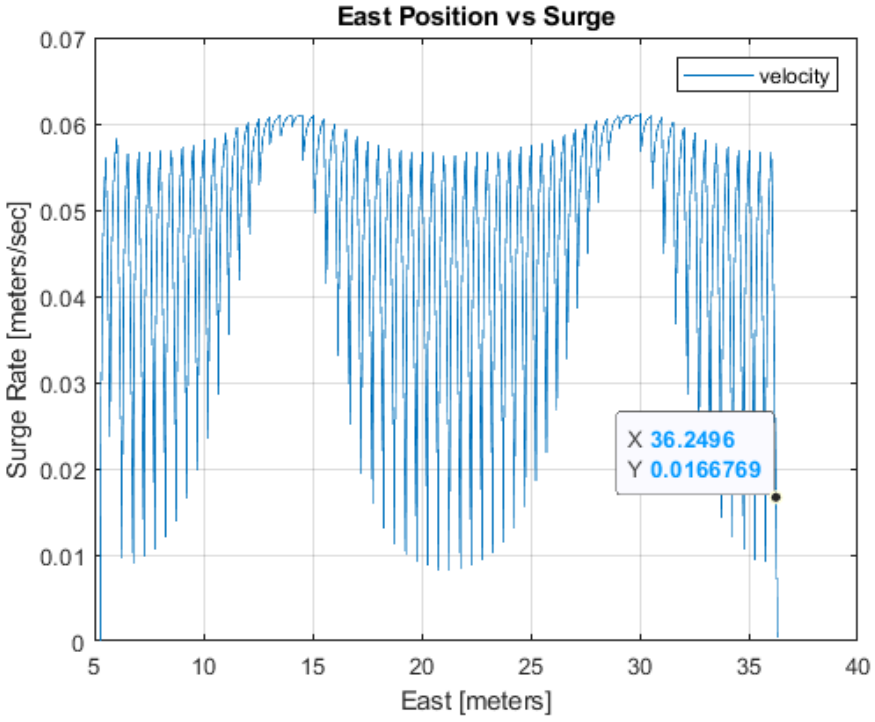

Figure 30: Plot of surge rate of the AUV as a function of position in east (Y-axis)

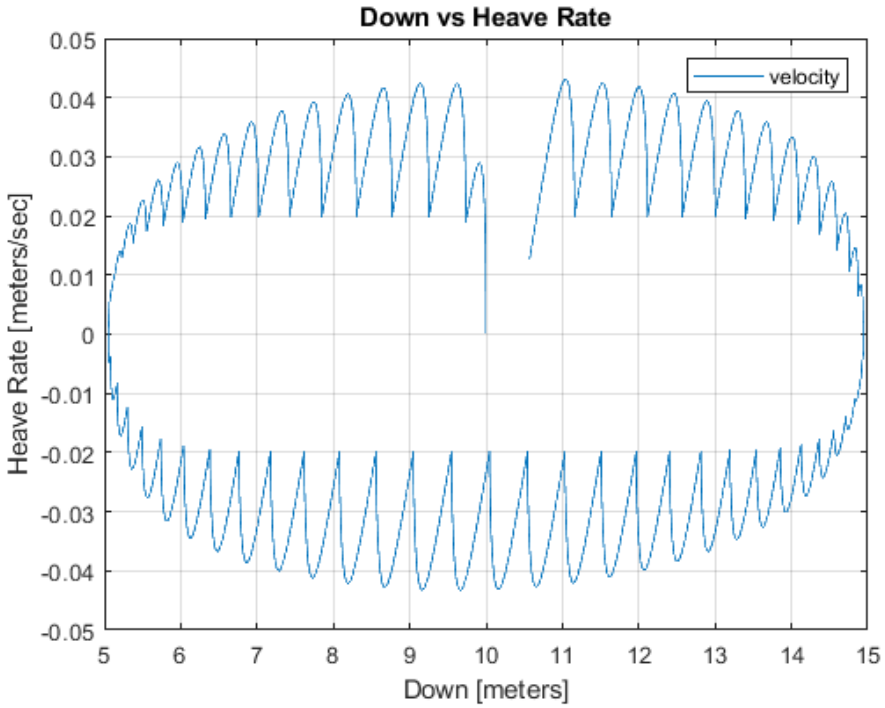

Figure 31: Plot of heave rate of the AUV as a function of position in down (Z-axis)

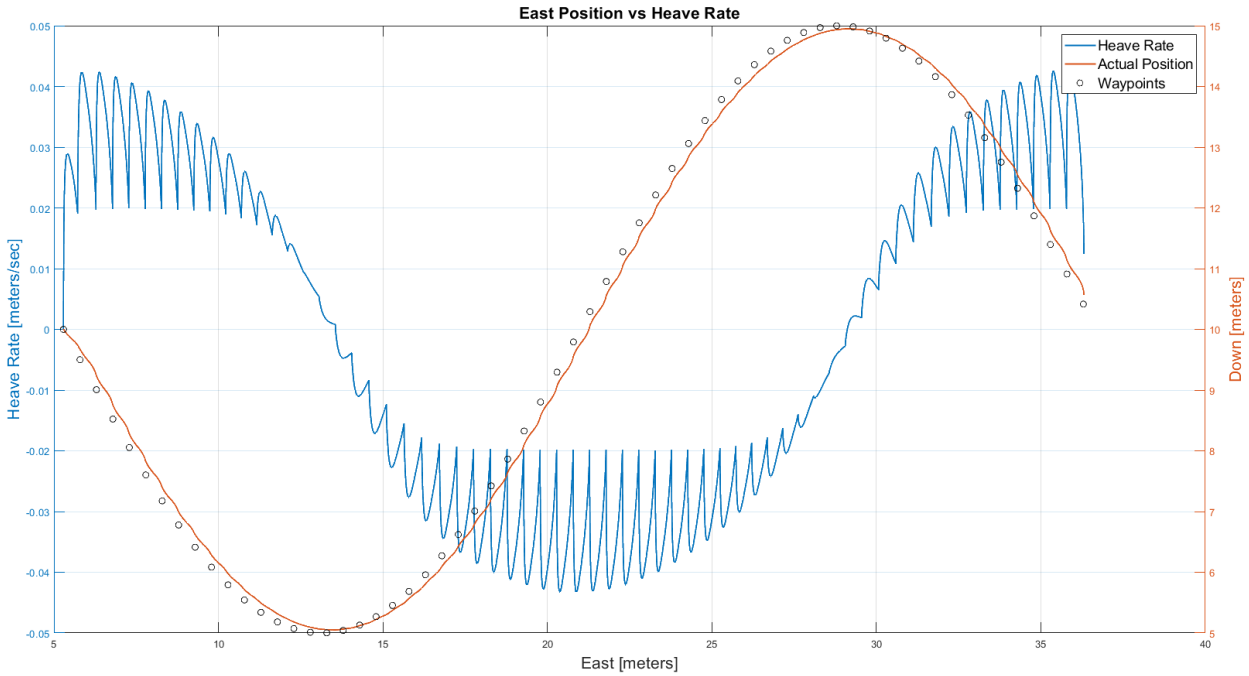

Figure 32: Analysis of the AUV behavior with heave rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

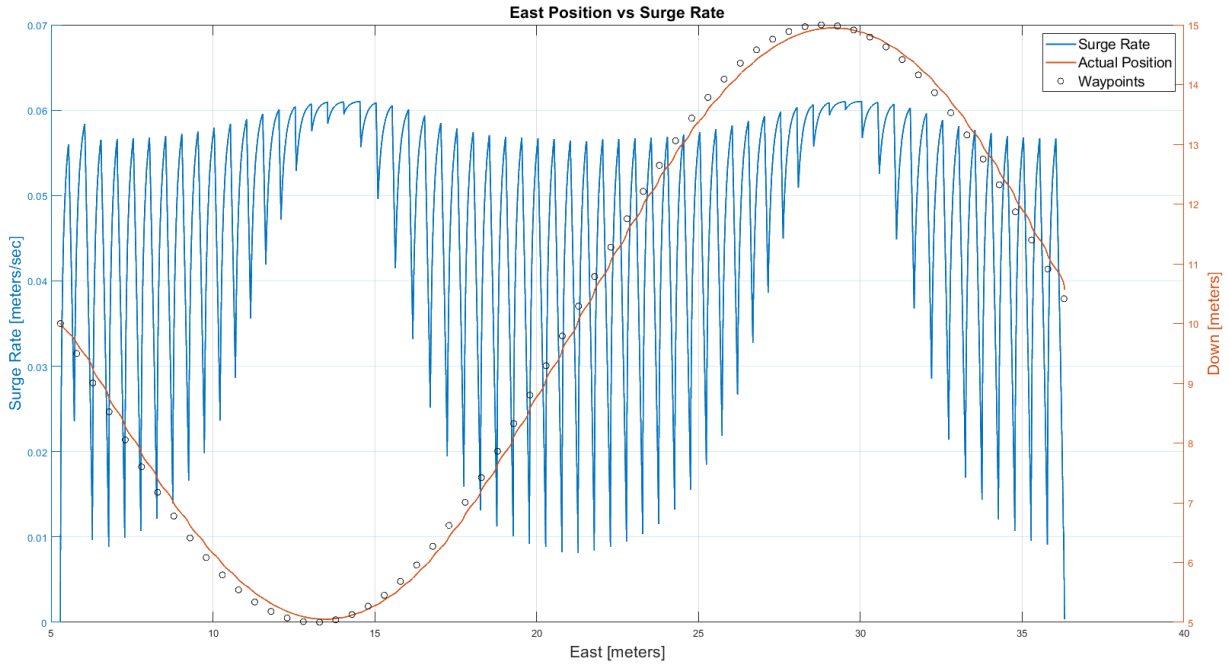

Figure 33: Analysis of the AUV behavior with surge rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

4.2 Waypoint-controller Parametric Analysis

The simulator has tested in a 3D space. We will observe how a synthetic underactuated AUV will be track a trajectory like a square wave with multiple waypoints, using all the actuated motion (i.e., Surge, Heave and Yaw). The test is to analyze the impact of time taken to finish the mission by different threshold setting ranging from 0.15m to 1m. No motion of the AUV is constrained other than the underactuated motions (i.e., Roll, Pitch and Sway).

Snapshots from an animation while AUV is tracking its desired trajectory is shown in the Figure 34. The behavior of the synthetic AUV with respect to the tracked waypoints, its trajectory with different threshold and analysis of time versus threshold settings are shown in Figures 35 to 37.

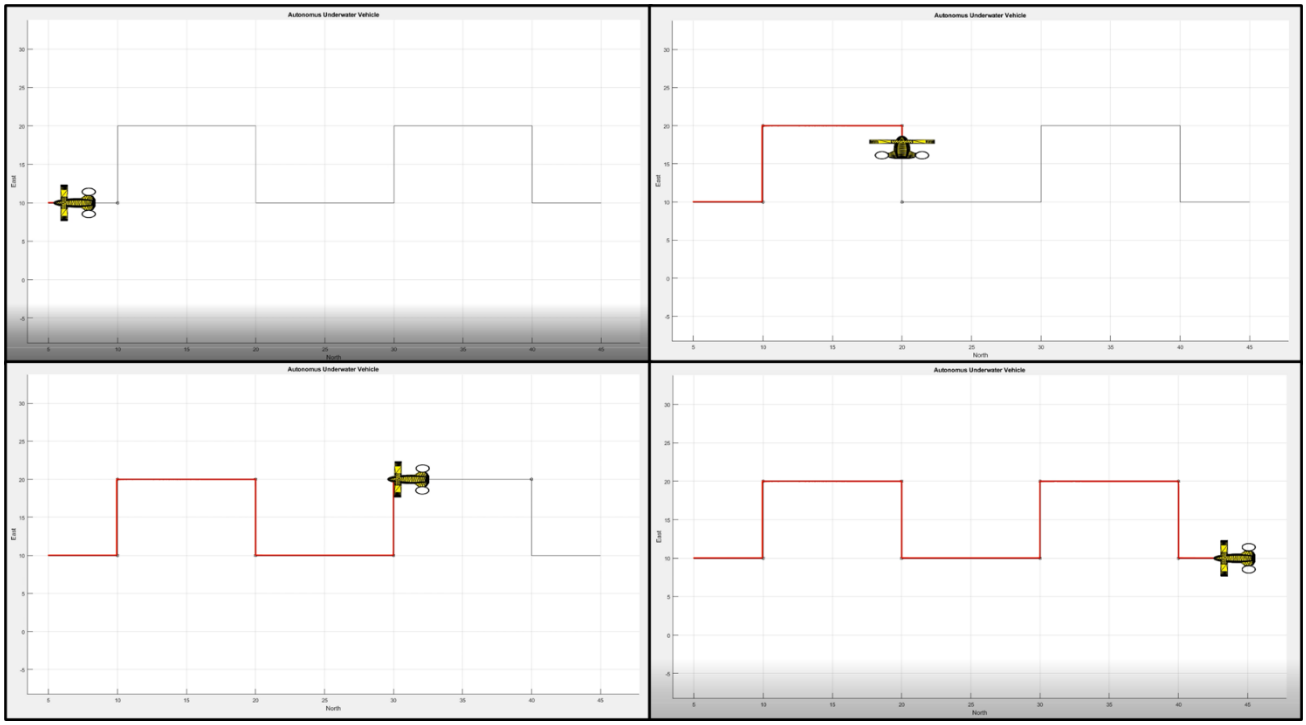

Figure 34: Snapshots from the AUV simulator visualization showing the synthetic AUV tracking a trajectory for waypoint-controller parametric analysis using a underactuated model

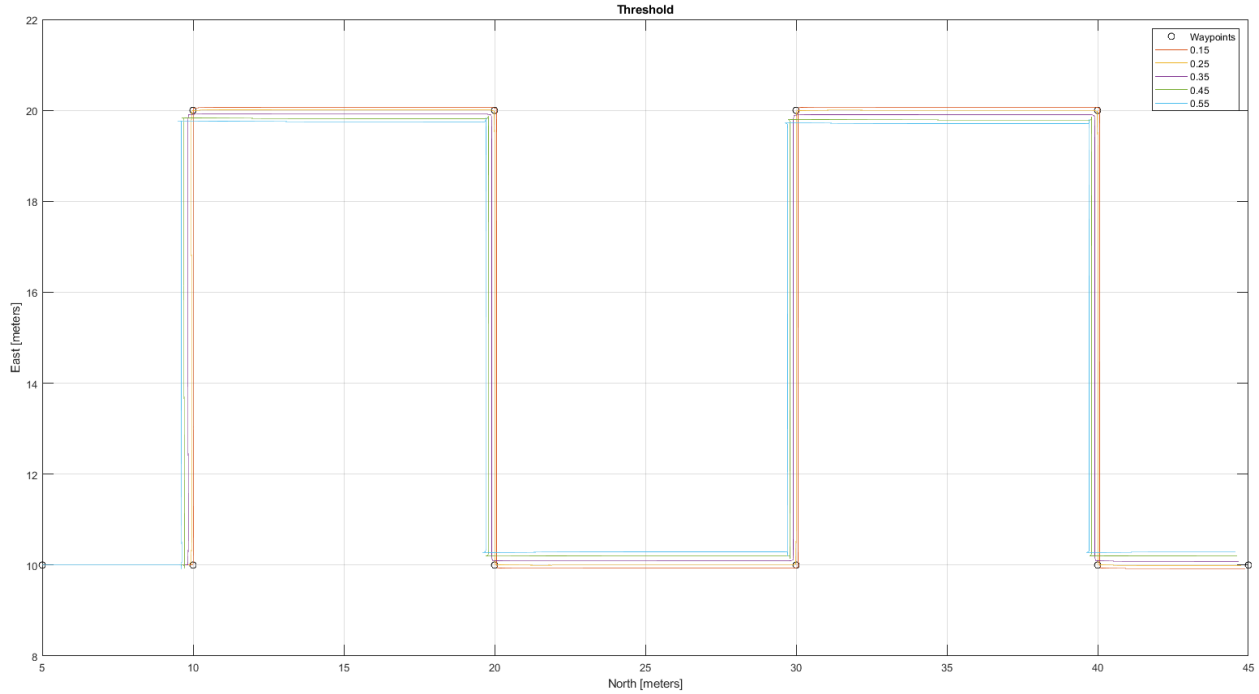

Figure 35: Trajectory tracking of the synthetic AUV with different threshols settings ranging from 0.15m to 0.55m and comparing the results

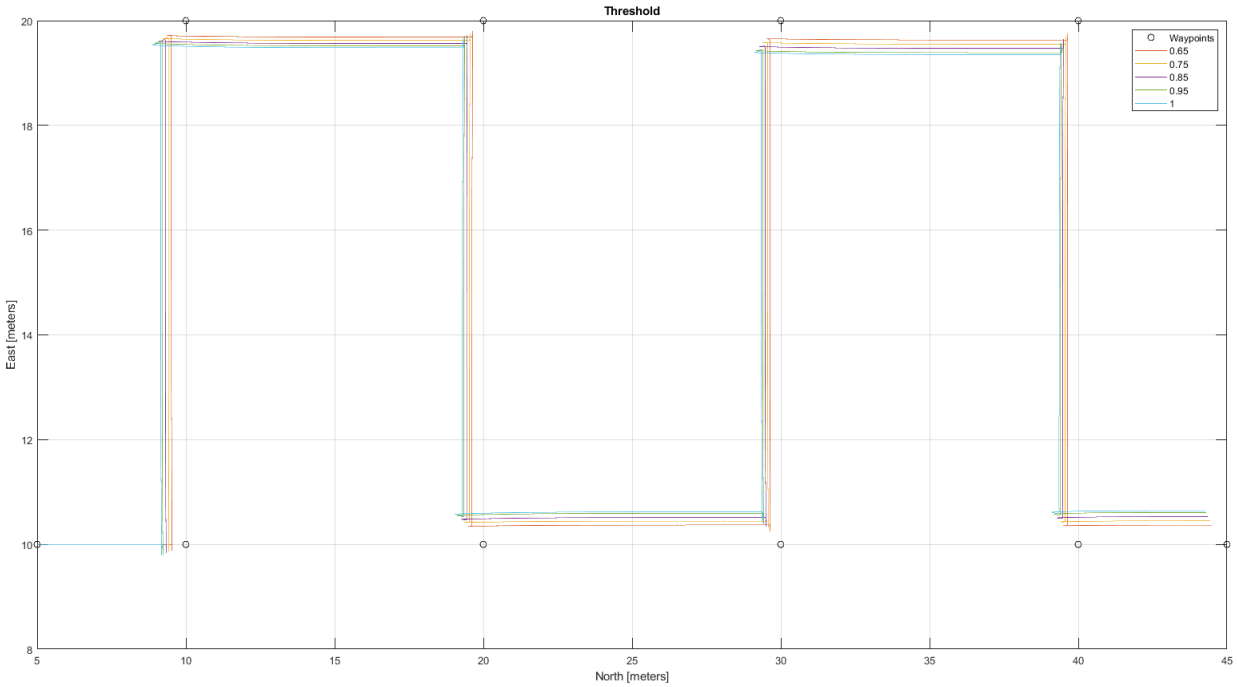

Figure 36: Trajectory tracking of the synthetic AUV with different thresholds settings ranging from 0.65m to 1m and comparing the results

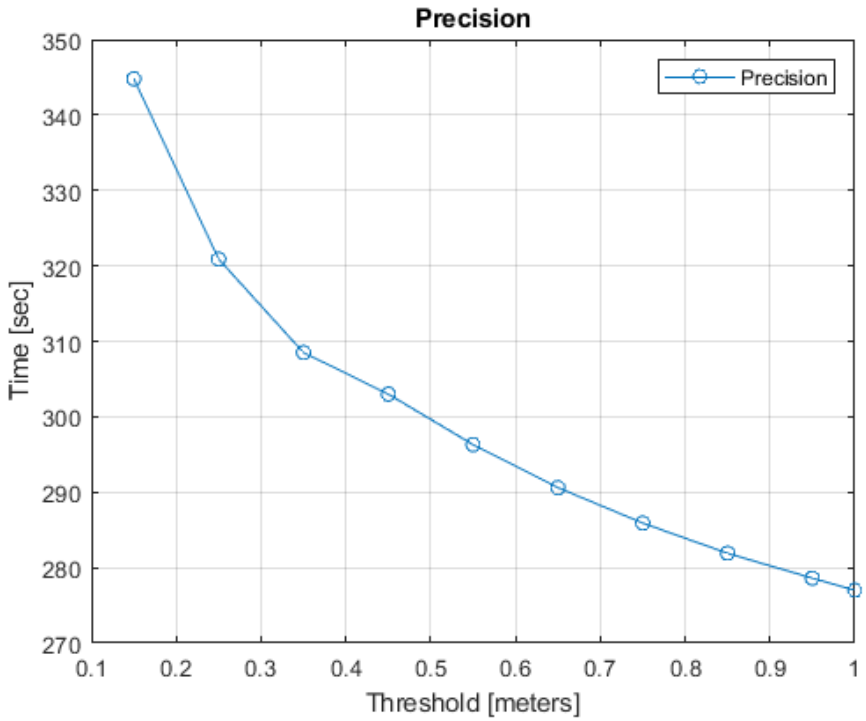

Figure 37 : Analysis of the synthetic AUV with the study time taken to traverse along fixed path to finish the mission with different threshold settings ranging from 0.15m to 1m.

4.3 A-Star Path Planner Analysis

The simulator was tested in a 3D space. We will observe how a synthetic underactuated AUV will track a trajectory of waypoints given by A star path planner to see whether AUV has a capability to interface with path planner, using all the actuated motion (i.e., Surge, Heave and Yaw). The test is to analyze the impact of time taken to finish the mission by setting different threshold ranges. No motion of the AUV is constrained other than the underactuated motions (i.e., Roll, Pitch and Sway).

Snapshots from an animation while AUV is tracking its desired trajectory is shown in the Figure 44. The behavior of the synthetic AUV with respect to the tracked waypoints, its position and velocity are shown in Figures 38 to 43.

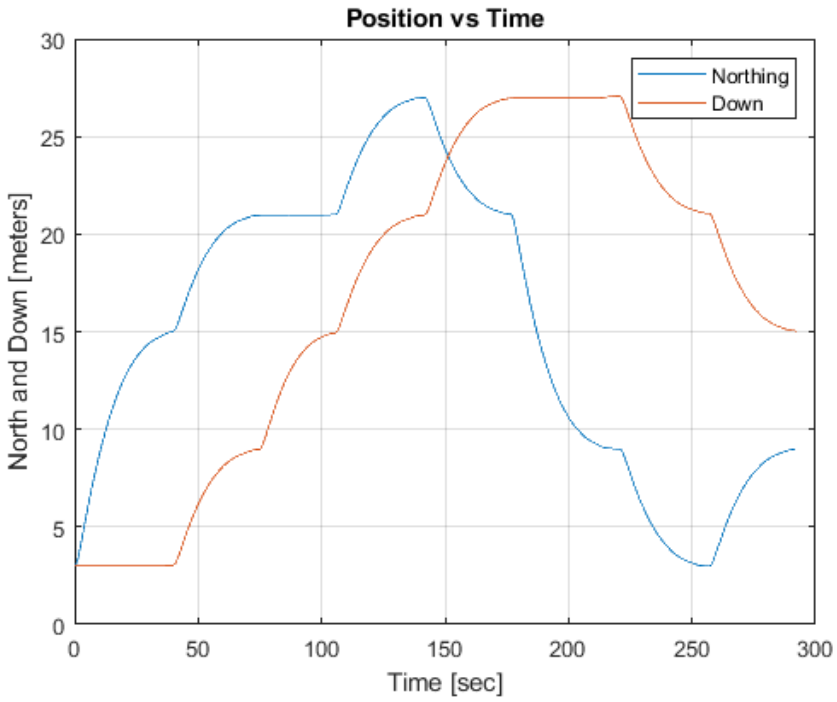

Figure 38 : Plot of AUV's surge and heave as a function of time

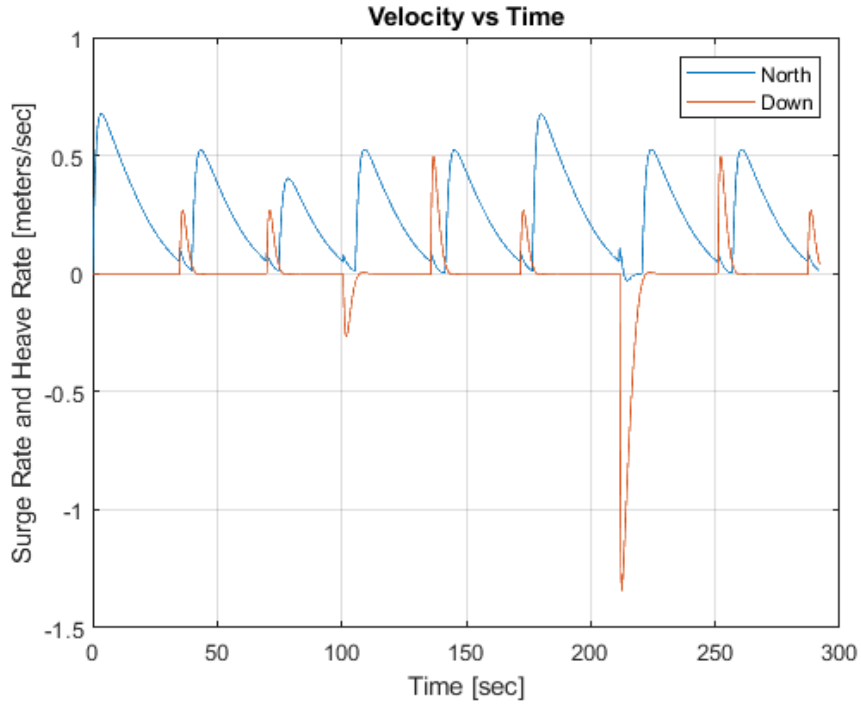

Figure 39: Plot of surge rate and heave rate of the AUV as a function of time

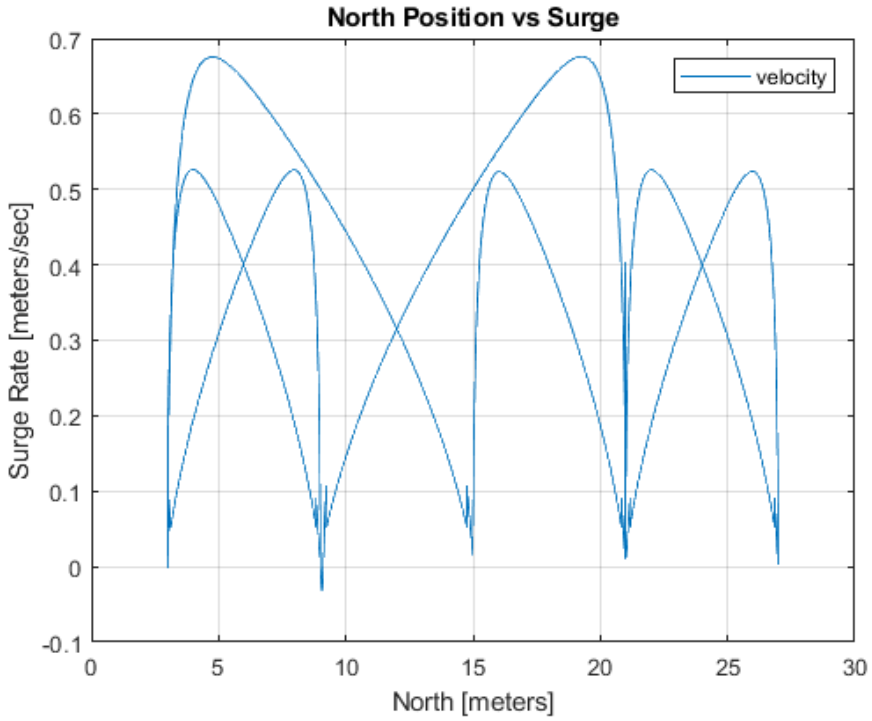

Figure 40: Plot of surge rate of the AUV as a function of position in north (X-axis)

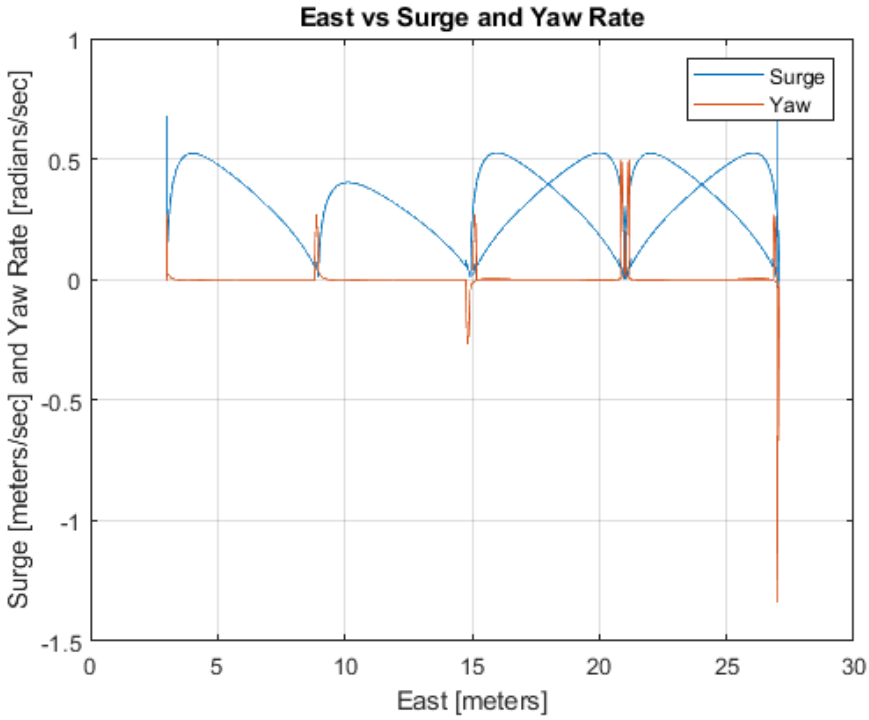

Figure 41: Plot of surge rate and yaw rate of the AUV as a function of position in east (Y-axis)

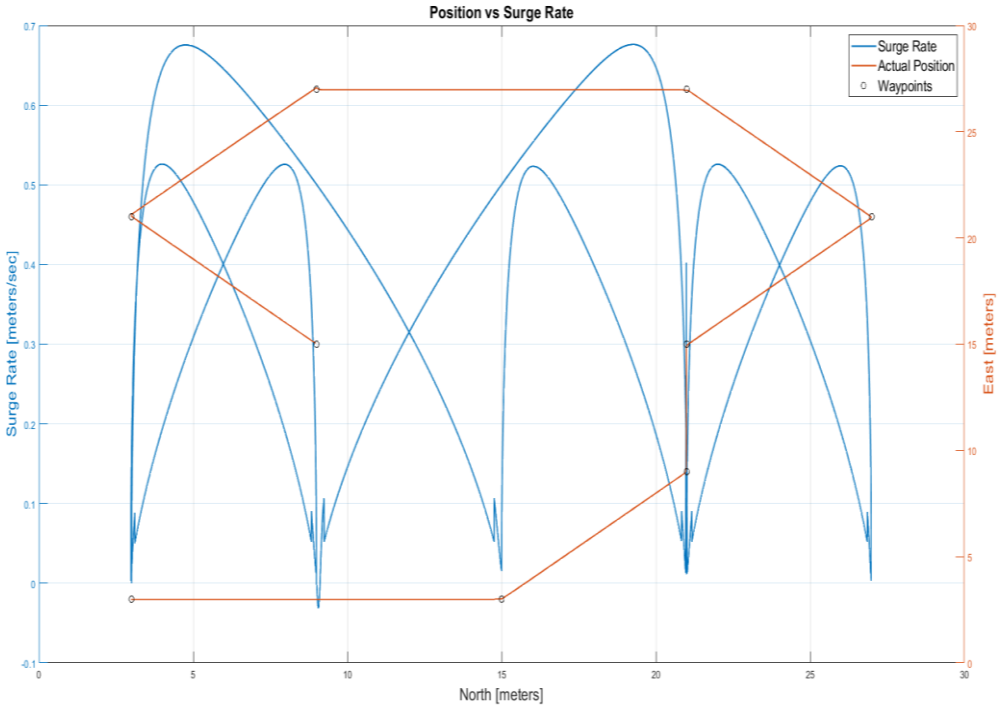

Figure 42: Analysis of the AUV behavior with surge rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

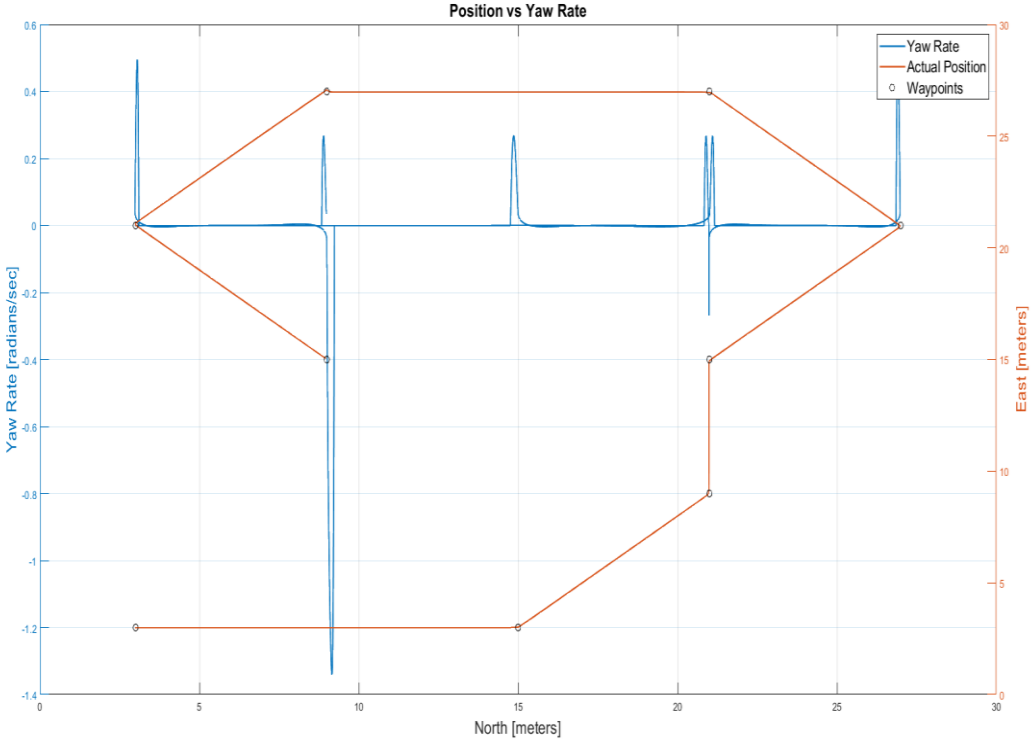

Figure 43: Analysis of the AUV behavior with yaw rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

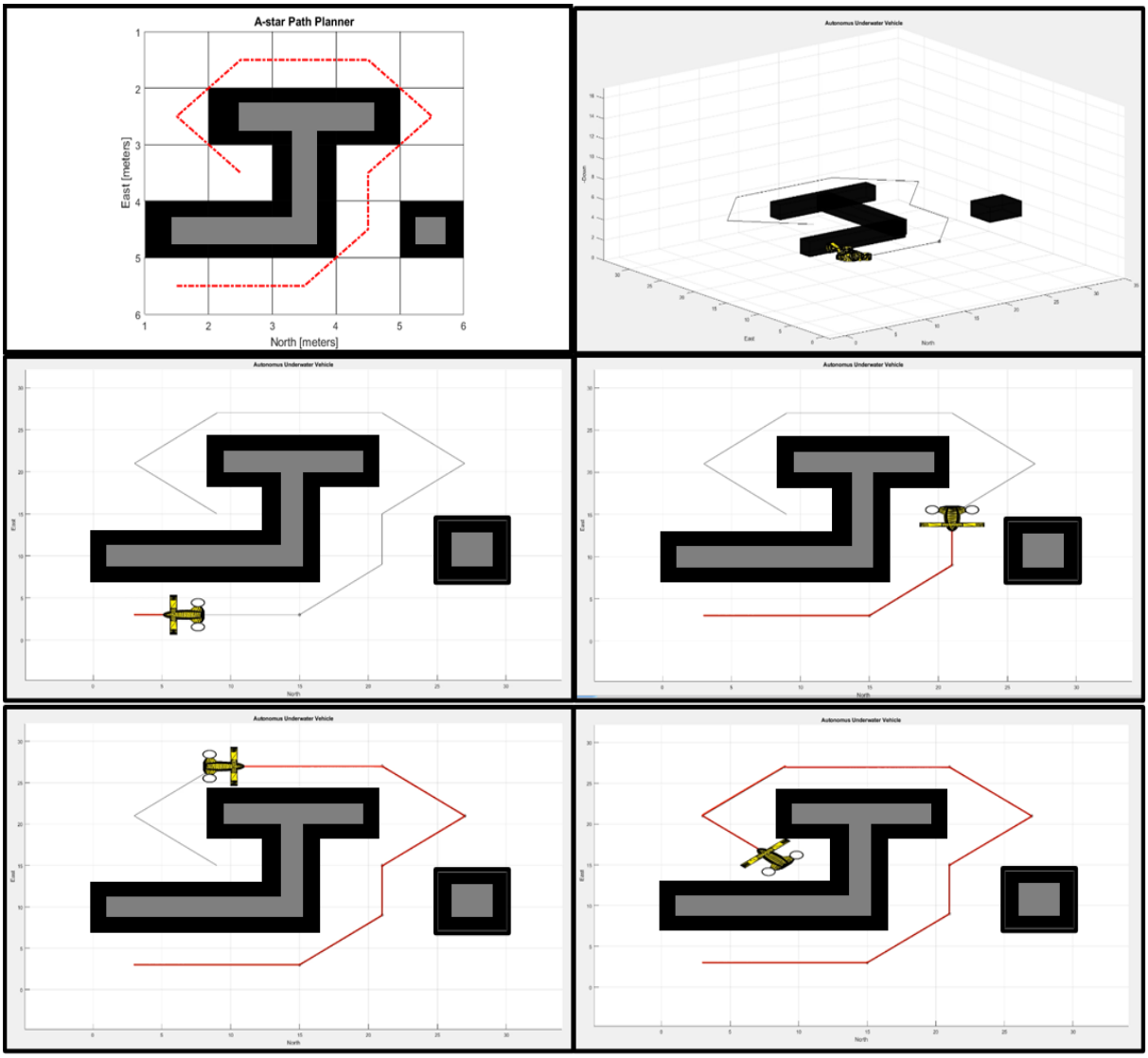

Figure 44 : Snapshots from the AUV simulator visualization showing the synthetic AUV following a path generated by a A* planner

4.4 AUV Velocity Model

In this analysis, we implemented a velocity input model rather than a traditional waypoint input model. The sole purpose of this model is to analyze whether the AUV is maintain the desired user-defined velocities and follow a smooth curvy trajectory rather than stop and go motion. The test is to analyze the behavior of the synthetic AUV. No motion of the AUV is constrained other than the underactuated motions (i.e., Roll, Pitch and Sway).

The moment of AUV is while performing the test has been captured for reference. The following graphs illustrate the behavior of the synthetic AUV with respect to the waypoints, position, and velocity.

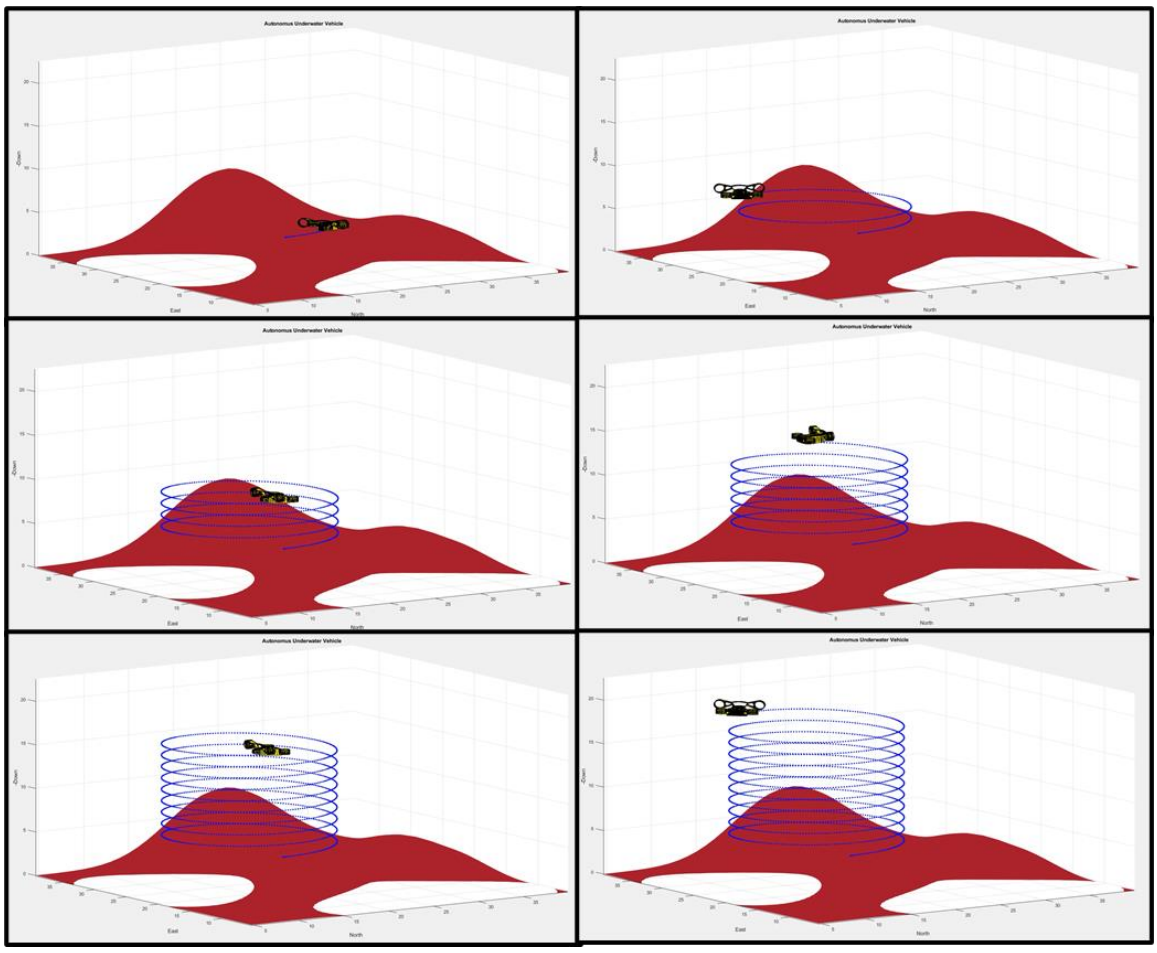

Figure 45: Snapshots from the AUV simulator visualization showing the synthetic AUV performing a kinematic velocity model using a underactuated model

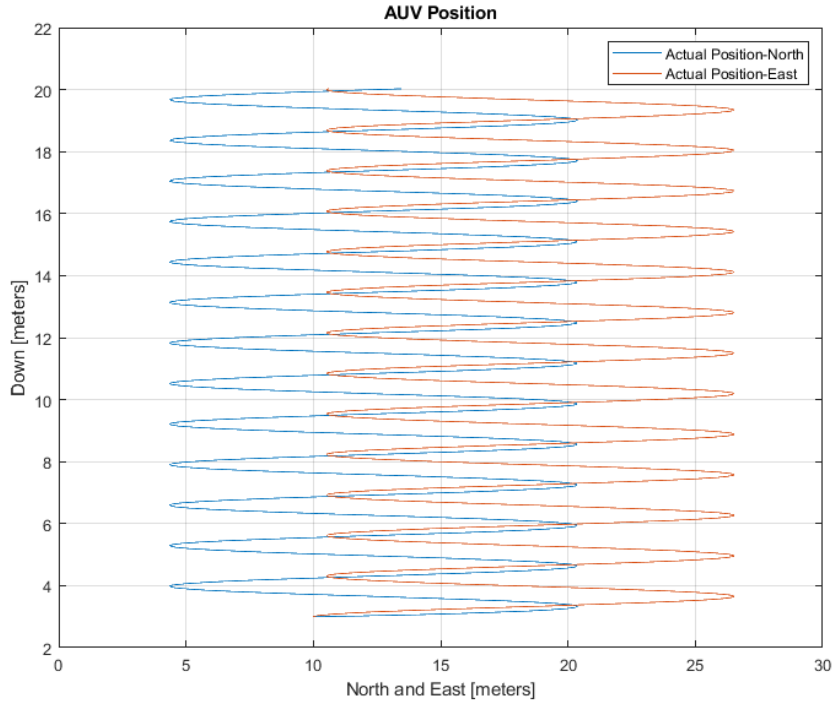

Figure 46: Plot of AUV's heave as a function of position in north(X-axis) and east (Y-axis)

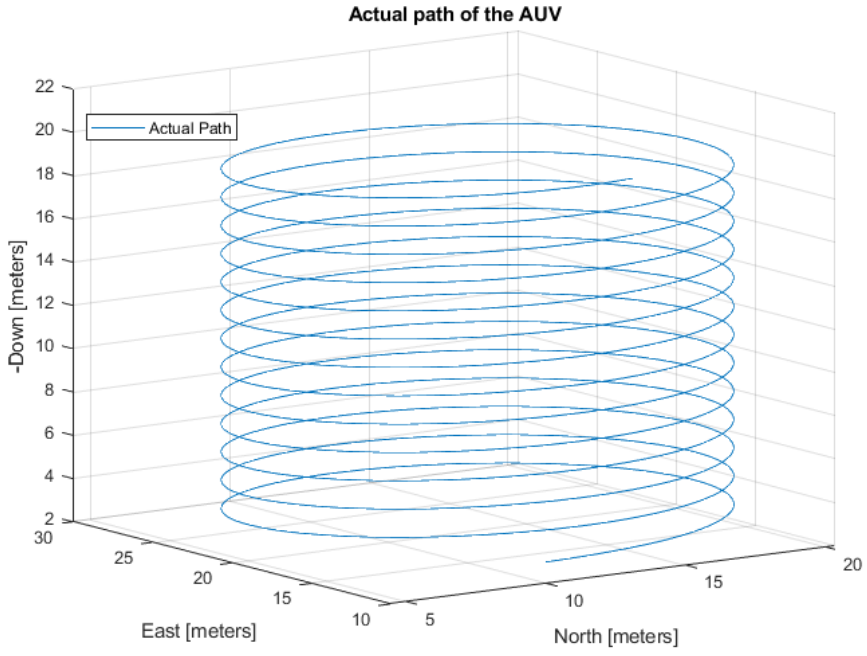

Figure 47: Plot of actual path travesed by the synthetic AUV using a underactuated model

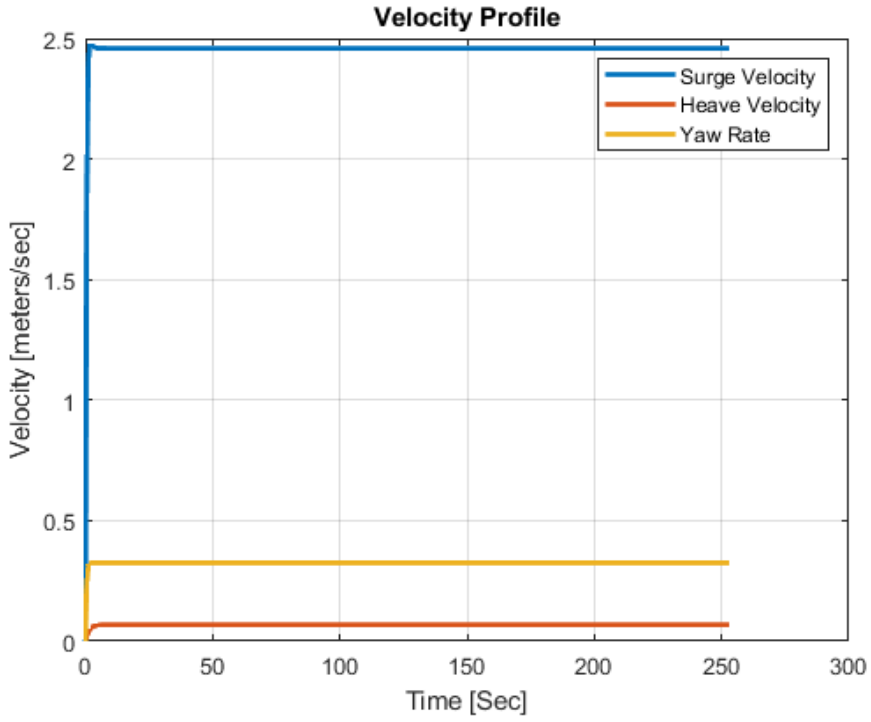

Figure 48: Plot of surge rate, yaw rate and heave rate of the AUV as a function of time

4.5 AUV Pipeline Inspection

The simulator has given a set of waypoints in a fashion, such that is it able to follow the structure of a simple pipeline. The sole purpose is to see whether AUV is able to track its trajectory when bunch of waypoints is given for wide area mission plan and check whether AUV is following all the waypoints in a sequence or neglecting any. No motion of the AUV is constrained other than the underactuated motions (i.e., Roll, Pitch and Sway).

The moment of AUV is while tracking its desired trajectory has been captured for your reference. the following graphs illustrate the behavior of the synthetic AUV with respect to the waypoints, position, and velocity

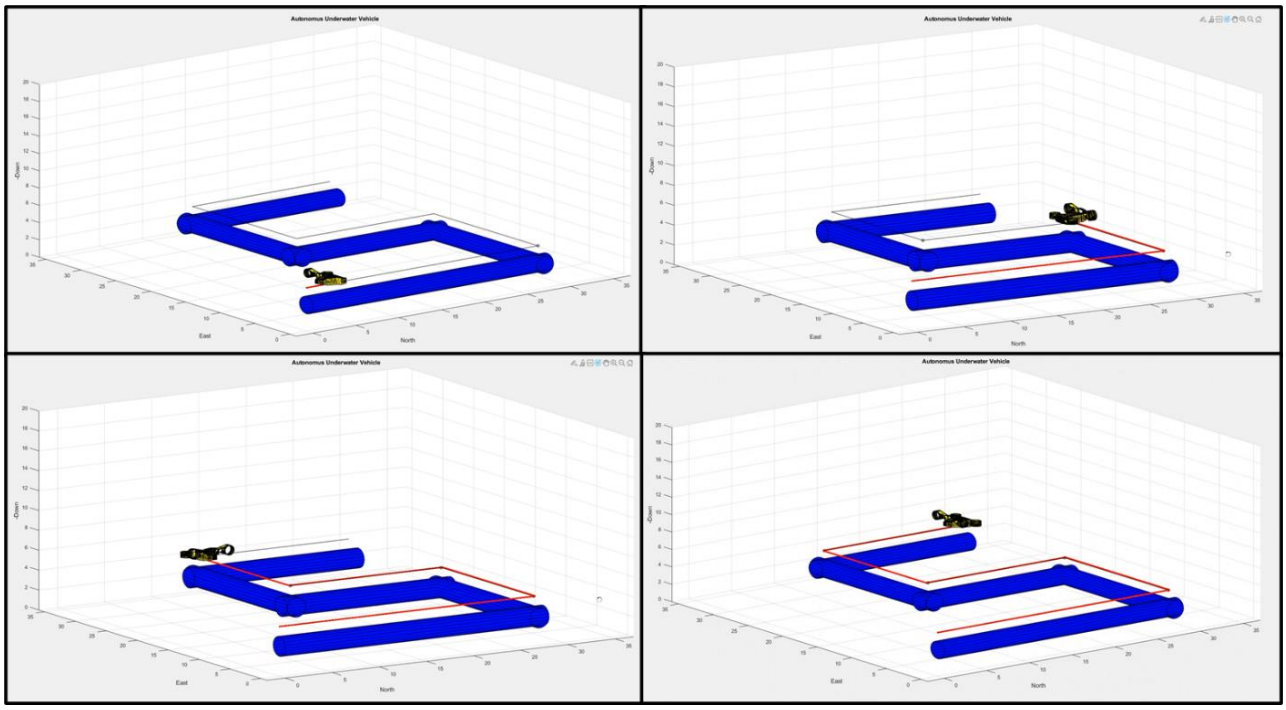

Figure 49: Snapshots from the AUV simulator visualization showing the AUV tracking a trajectory along a pipeline to illustrate its ability to inspect it using a underactuated model

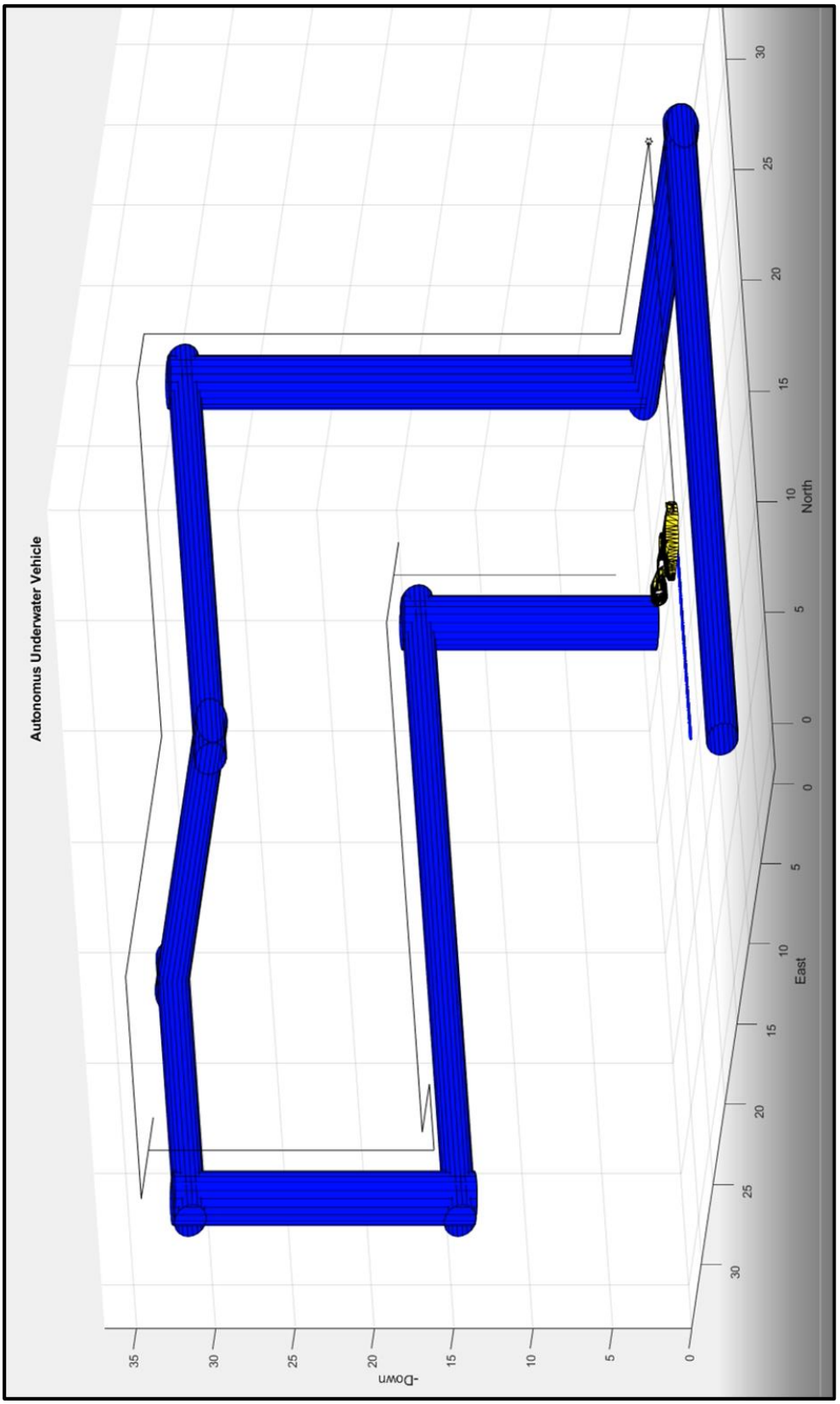

Figure 50: Snapshots from the AUV simulator visualization showing the AUV tracking a complex trajectory along a pipeline to illustrate its ability to inspect it using a underactuated model

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

This thesis presented the modeling and simulation of autonomous underwater vehicles by borrowing inspiration from current ongoing demand in underwater robotics research. The goal was to develop a modeling and simulation platform of the AUV, which allows investigation of a multitude of design, analysis, and research problems relevant to achieving fully autonomous underwater vehicles. For instance, we worked on implementing a path planner with a PD controller on a dynamic model of an AUV. Similarly, we can study the impact of water currents and do an analysis on how an AUV will track its trajectory. Although the autonomous underwater vehicle met with success in showcasing its ability, there is still scope for improvements in the near future. Some important directions for future work include:

1) Introducing a 3D path planner and testing the capability in 3D environment.
2) Implement the water current effect on the AUV and see how the vehicle is behaving because of it and how it is trying to track its trajectory.
3) Implement and compare different controllers on the synthetic AUV.
4) Develop the dynamic model by taking an asymmetric design, where the CG and CB are not at the center of the vehicle and observe whether it is controlling its instability.
5) Build an actual AUV model and analyze the simulator data with physical model and validate the utility of the AUV simulator

With future improvements to the current design, this research is expected to contribute significantly to field of autonomous underwater vehicles.

REFERENCES

Allotta, Benedetto, et al. "A low-cost autonomous underwater vehicle for patrolling and monitoring." Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment 231.3 (2017): 740-749.

Becker, Carlos, David Ribas, and Pere Ridao. "Simultaneous sonar beacon localization & AUV navigation." IFAC Proceedings Volumes 45.27 (2012): 200-205.

Cao, Hua, Nathan E. Brener, and S. Sitharama Iyengar. "3D large grid route planner for the autonomous underwater vehicles." International Journal of Intelligent Computing and Cybernetics (2009).

Carreras, M., et al. "An overview on behaviour-based methods for AUV control." IFAC Proceedings Volumes 33.21 (2000): 141-146.

Carroll, Kevin P., et al. "AUV path planning: An A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones." *Proceedings of the 1992 symposium on autonomous underwater vehicle technology*. IEEE, 1992.

Carsten J, Ferguson D, Stentz A (2006) 3D field D*: improved path planning and replanning in three dimensions. In: IEEE international conference on intelligent robots and systems (IROS ’06), pp 3381–3386

Chen, Chen-Wei, Jen-Shiang Kouh, and Jing-Fa Tsai. "Modeling and simulation of an AUV simulator with guidance system." IEEE Journal of Oceanic Engineering 38.2 (2013): 211-225.

Cheng, Chunxi, et al. "Path planning and obstacle avoidance for AUV: A review." *Ocean Engineering* 235 (2021): 109355.

Cook, Daniel, Andrew Vardy, and Ron Lewis. "A survey of AUV and robot simulators for multi-vehicle operations." 2014 IEEE/OES Autonomous Underwater Vehicles (AUV). IEEE, 2014.

Cristi, Roberto, Fotis A. Papoulias, and Anthony J. Healey. "Adaptive sliding mode control of autonomous underwater vehicles in the dive plane." IEEE journal of Oceanic Engineering 15.3 (1990): 152-160.

Cui, R., Zhang, X., and Cui, D. (2016). Adaptive sliding-mode attitude control for autonomous underwater vehi-cles with input nonlinearities. Ocean Engineering, 123, 45–54.

Elmokadem, T., Zribi, M., and Youcef-Toumi, K. (2016). Trajectory tracking sliding mode control of underactu-ated auvs. Nonlinear Dynamics, 84(2), 1079–1091.

Encarna¸c˜ao, P. and Pascoal, A. (2001). Combined tra-jectory tracking and path following: an application to the coordinated control of autonomous marine craft. In Decision and Control, 2001. Proceedings of the 40th IEEE Conference on, volume 1, 964–969. IEEE.

Fossen, Thor I. "Marine control systems–guidance. navigation, and control of ships, rigs and underwater vehicles." Marine Cybernetics, Trondheim, Norway, Org. Number NO 985 195 005 MVA, www. marinecybernetics. com, ISBN: 82 92356 00 2 (2002).

Gonzalez, T., Moreno, J.A., and Fridman, L. (2011). Variable gain super-twisting sliding mode control. IEEE Transactions on Automatic Control, 57.

Guerrero, J., et al. "Autonomous underwater vehicle robust path tracking: Auto-adjustable gain high order sliding mode controller." *IFAC-PapersOnLine* 51.13 (2018): 161-166.

Healey, Anthony J., and David Lienard. "Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles." IEEE journal of Oceanic Engineering 18.3 (1993): 327-339.

J. C. Hyland, "Optimal Obstacle Avoidance Path Planning for Autonomous Underwater Vehicles," *Proceedings of the 6th International Symposium on Unmanned Untethered Submersible Technology*, 1989, pp. 266-278, doi: 10.1109/UUST.1989.754721.

Kim, Ayoung, and Ryan Eustice. "Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection." 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009.

Kinsey, J.C., Eustice, R.M., and Whitcomb, L.L. (2006). A survey of underwater vehicle navigation: Recent ad-vances and new challenges. In IFAC Conference of Maneuvering and Control of Marine Craft, volume 88.

Lapierre, L., Soetanto, D., and Pascoal, A. (2003). Non-linear path following with applications to the control of autonomous underwater vehicles. In Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, volume 2, 1256–1261. IEEE.

Li, Y., Jiang, Y.q., Wang, L.f., Cao, J., and Zhang, G.c. (2015). Intelligent pid guidance control for auv path tracking. Journal of Central South University, 22(9), 3440–3449.

Li, J.H. and Lee, P.M. (2005). Design of an adaptive nonlinear controller for depth control of an autonomous underwater vehicle. Ocean engineering, 32(17), 2165– 2181.

Li, Daoliang, Peng Wang, and Ling Du. "Path planning technologies for autonomous underwater vehicles-a review." *IEEE Access* 7 (2018): 9745-9768.

Likhachev M, Ferguson D, Gordon G, Stentz A, Thrun S (2005) Anytime dynamic A*: an anytime, replanning algorithm. In: 5th international conference on automated planning and scheduling (ICAPS 2005), pp 262–271

Liu, Y., Wang, F., Lv, Z., Cao, K., Lin, Y., 2018. Pixel-to-action policy for underwater pipeline following via deep reinforcement learning. In: 2018 IEEE International Conference of Intelligent Robotic and Control Engineering. IEEE, pp. 135–139.

Mahmoud Zadeh, Somaiyeh, et al. "A novel versatile architecture for autonomous underwater vehicle's motion planning and task assignment." *soft computing* 22.5 (2018): 1687-1710.

Manzanilla, A., Castillo, P., and Lozano, R. (2017). Non-linear algorithm with adaptive properties to stabilize an underwater vehicle: real-time experiments. IFAC-PapersOnLine,50(1),6857–6862.

Matsebe, O., C. M. Kumile, and N. S. Tlale. "A review of virtual simulators for autonomous underwater vehicles (AUV's)." IFAC Proceedings Volumes 41.1 (2008): 31-37.

Omerdic, E., G. N. Roberts, and P. Ridao. "Fault detection and accommodation for ROVs." IFAC Proceedings Volumes 36.21 (2003): 127-132.

Panda, Madhusmita, et al. "A comprehensive review of path planning algorithms for autonomous underwater vehicles." *International Journal of Automation and Computing* 17.3 (2020): 321-352

Petres C, Pailhas Y, Evans J, Petillot Y, Lane D (2005) Underwater path planning using fast marching algorithms. Oceans Eur Conf 2:814–819

Petres C, Pailhas Y, Patron P, Petillot Y, Evans J, Lane D (2007) Path planning for autonomous underwater vehicles. IEEE Trans Robot 23(2):331–341

Ramos, P., Cruz, N., Matos, A., Neves, M.V., Pereira, F.L., 2001. Monitoring an ocean outfall using an AUV. In: MTS/IEEE Oceans 2001. an Ocean Odyssey. Conference Proceedings. Vol. 3, IEEE, pp. 2009–2014.

Ridao, Pere, et al. "Graphical simulators for AUV development." First International Symposium on Control, Communications and Signal Processing, 2004. IEEE, 2004.

Sagala, F., Bambang, R.T., 2011. Development of sea glider autonomous underwater vehicle platform for marine exploration and monitoring. Indian J. Geo Marine Sci. 40 (2), 287–295.

Sahu, B.K. and Subudhi, B. (2014). Adaptive tracking control of an autonomous underwater vehicle. International Journal of Automation and Computing, 11(3), 299–307.

Silvestre, Carlos. "Multi-objective optimization theory with applications to the integrated design of controllers/plants for autonomous vehicles." dissertation (2000).

Silvestre, Carlos, et al. "A bottom-following preview controller for autonomous underwater vehicles." IEEE Transactions on control systems technology 17.2 (2008): 257-266.

Vervoot, J.H.A.M. " Modeling and control of an unmanned underwater vehicle." *Master traineeship report*, University of Canterbury, November 2008

Yao, Peng, and Shiqiang Zhao. "Three-dimensional path planning for AUV based on interfered fluid dynamical system under ocean current (June 2018)." IEEE Access 6 (2018): 42904-42916.

You, Sam-Sang, and Young-Ho Chai. "Diving autopilot design for underwater vehicles using multi-objective control synthesis." KSME International Journal 12.6 (1998): 1116-1125.

Warren, Charles W. "A technique for autonomous underwater vehicle route planning." IEEE Journal of Oceanic Engineering 15.3 (1990): 199-204.

VITA

Sai Krishna Abhiram Kondapalli

Mechanical & Aerospace Engineering
Old Dominion University
115 Kaufman Hall, Norfolk, VA 23529

EDUCATION

Thesis: "Development of Modeling and Simulation Platform for Path-Planning and Control of Autonomous Underwater Vehicles in Three-Dimensional Spaces"

Master of Science in Mechanical Engineering at Old Dominion University, January 2020,

Bachelor of Science in Mechanical Engineering at Koneru Lakshmaiah University, May 2017.

ACADEMIC EMPLOYMENT

Graduate Teaching Assistant for MAE 431 Mechanisms Analysis and Design to Dr. Krishnanand Kaipa, Department of Mechanical & Aerospace Engineering, Old Dominion University, Jan 2021 – May 2021. Responsibilities included grading homework, projects, and exams.

Graduate Teaching Assistant for MAE 434W Project Design and Management 1 to Dr. Stacie Ringleb, Department of Mechanical & Aerospace Engineering, Old Dominion University, Jan 2021 – May 2021. Responsibilities included grading blog posts, comments, and projects.

Graduate Teaching Assistant for MAE 336 Electro-Mechanical Systems to Dr. Krishnanand Kaipa, Department of Mechanical & Aerospace Engineering, Old Dominion University, August 2021 – December 2021. Responsibilities included Planning, preparation, and instruction of the assigned curriculum. Demonstrated the experiments and software to students. Provided support and guidance to students, evaluated student's assignments and exams. Maintained records of student's grades.

Graduate Teaching Assistant for MAE 436 Dynamic System and Control to Dr. Jen-Kuang Huang, Department of Mechanical & Aerospace Engineering, Old Dominion University, August 2021 – December 2021. Responsibilities included grading homework, projects, and exams.

Graduate Research Assistant for CRAMLAB to Dr.Krishnanand Kaipa, Department of Mechanical & Aerospace Engineering, Old Dominion University, Jan 2022 – May 2022.Responsibilities including a development of a Simulator for Autonomous Underwater Vehicle.