Development of a Three-Dimensional Simulator for Integrated Testing of Path-Planners and Controllers for Autonomous Underwater Vehicles

Abhiram Kondapalli
Dept. of Mechanical & Aerospace Engr.
Old Dominion University
Norfolk, Virginia, USA
skond002@odu.edu

Payal Nandi
Dept. of Mechanical & Aerospace Engr.
Old Dominion University
Norfolk, Virginia, USA
pnand001@odu.edu

Krishnanand Kaipa
Dept. of Mechanical & Aerospace Engr.
Old Dominion University
Norfolk, Virginia, USA
kkaipa@odu.edu

OCEANS 2022, Hampton Roads | 978-1-6654-6809-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/OCEANS47191.2022.9976960

***Abstract*—Autonomous underwater vehicles (AUVs) operating in deep sea and littoral environments have diverse applications ranging from marine biology exploration and search for plane crash sites to ship-hull inspection and border patrol. This paper 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 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. Experimental validation is carried out by performing integrated waypoint planner (using the A* algorithm) and PD controller implementations that allow the synthetic AUV to track dynamically feasible trajectories in two- and three- dimensional spaces. An underwater pipe-line inspection task carried out by the AUV is demonstrated in a simulated environment.**

Keywords—autonomous underwater vehicles, underwater robotics, modeling and simulation, path planning, AUV simulators

I. Introduction

Autonomous underwater vehicles (AUVs) operating in deep sea and littoral environments have diverse applications including marine biology exploration [1], ocean environment monitoring [2], search for plane crash sites [3], inspection of ship-hulls [4] and pipelines [5], harbor patrolling [6], 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.

This paper 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. Given that most path-planning algorithms were originally developed and 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 [7]. However, path-planning algorithms for AUVs operating in three-dimensional spaces are beginning to appear [8, 9]. The approach adopted in this paper allows integration between path-planners and closed-loop

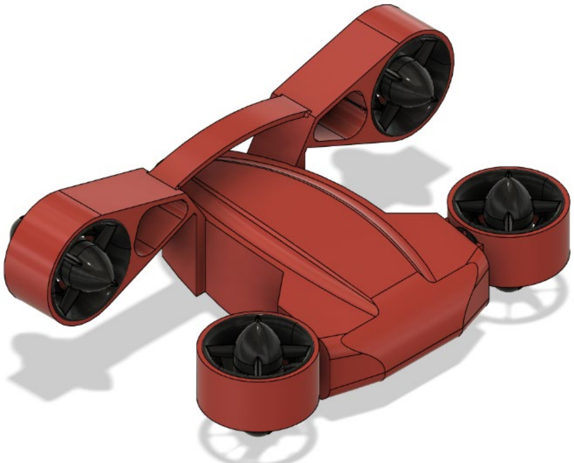


Fig 1. Assembly CAD model of the AUV prototype designed at the Collaborative Robotics & Adaptive Machines Laboratory, ODU. Four brushless DC motor-based thrusters are used to achieve surge, heave, and yaw motions in the AUV.

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.

978-1-6654-6809-1/22/$31.00 ©2022 IEEE

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. 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 (Fig. 2a), (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 (Fig. 2b) 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 A* algorithm and PD controller implementations that allow the traversal of the synthetic AUV in two-dimensional 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.

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

II. RELATED WORK

A. 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 [10]. 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 [11]. Other methods like D* [12] and fast marching (FM) algorithm [13] have been employed to solve the same problem. Petres et al. [13] 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 [7] 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 [14].

B. Control

Many control strategies have been developed to stabilize the AUV in the presence of unknown perturbations and modeling uncertainties [15]. Control is required at three levels of autonomy including stabilization to a point, path following, and trajectory tracking [16]. 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 [17].

Different control methods with adaptive properties have been developed to stabilize an underwater vehicle in the presence of external disturbances [18-20]. For the path following problem, a controller based on Lyapunov theory and Backstepping technique was designed by Lapierre et al. [21]. A robust controller based on sliding modes technique is presented by Elmokadem et al. [22], 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 [23] designed an adaptable controller capable of estimating parametric perturbations and uncertainties. Li et al. [24] designed an adaptive fuzzy PID controller to follow straight lines that are commonly used in underwater reconnaissance missions. Guerrero et al., (2018) [15] designed a robust algorithm based on the second order sliding mode technique with a self-adjusting gain proposed by Gonzalez et al. [25], 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.

C. 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 [26]. The longitudinal model considers the AUV using the surge, heave, and pitch to move in two-dimensional vertical plane (Silvestre et al., 2008).

Identify applicable funding agency here. If none, delete this text box.

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.

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 [7, 11]. However, path-planning algorithms for AUVs operating in three-dimensional spaces are beginning to appear [8, 9].

D. AUV Simulators

Many graphical simulators have been developed in the past [28, 29]. 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 does not last for one second of the reality. In both cases, the temporal properties of the implemented algorithms are not considered [29]. Several MATLAB/SIMULINK based offline simulators developed in the past for different applications: Comparison of different control architectures for AUVs [30], Simulation of a system for fault diagnosis and recovery applied to ROVs [31]. There also exist more high-fidelity three-dimensional dynamic simulators for AUVs such as SUBSIM and DEEPWORKS [32]. 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 [28]. A notable example is the development of an intelligent AUV simulator by [17].

III. DEVELOPMENT OF AUV MODELING AND SIMULATION PLATFORM

A. 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 $\eta = [x \quad y \quad z \quad \theta \quad \phi \quad \psi]^T$. A cylindrical coordinate system is considered in the motion model (Fig. 3). 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

$$\tau = \begin{bmatrix} F_x & F_y & F_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T \tag{1}$$

Consider the forces acting on the AUV with the help of a free body diagram. 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.

$$\tau = M\dot{v} + (C + D)v + g(\eta) \tag{2}$$

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 [33]. The mass and inertia matrix consists of a rigid body mass M_{RB} and an added mass M_A, respectively.

$$M = M_A + M_{RB} \tag{3}$$

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.

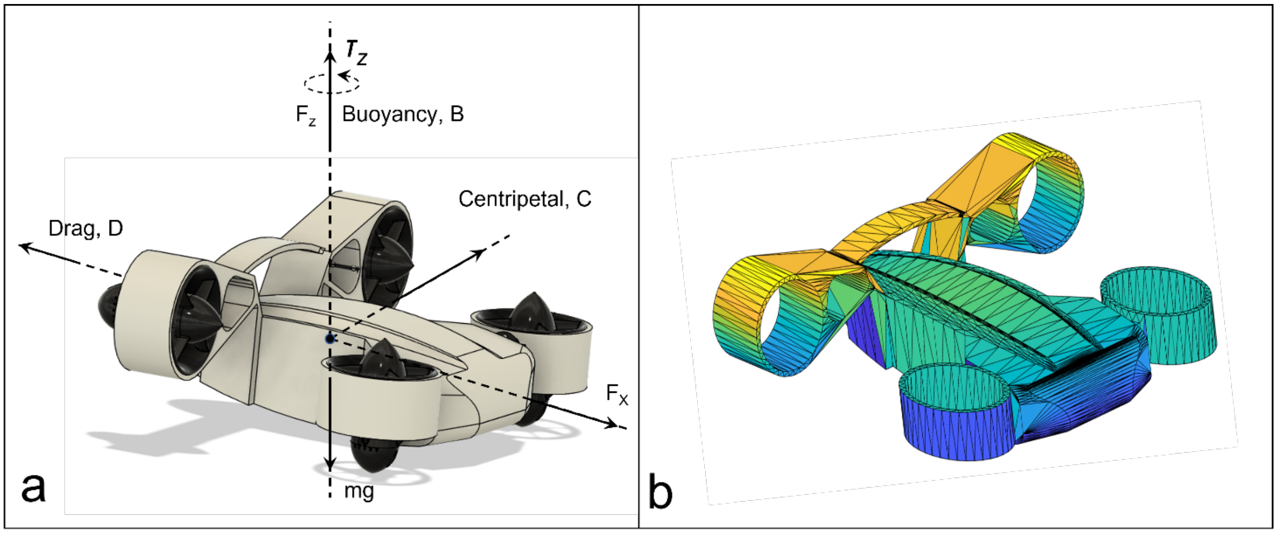


Fig 2 (a) Free body diagram showing the external forces and actuation forces/torques acting on the AUV. (b) 3D rendering used for visualization of the AUV in the animation environment

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 $C_A(v)$,

$$C(v)=C_{RB}(v)+\ C_A(v) \tag{4}$$

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

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

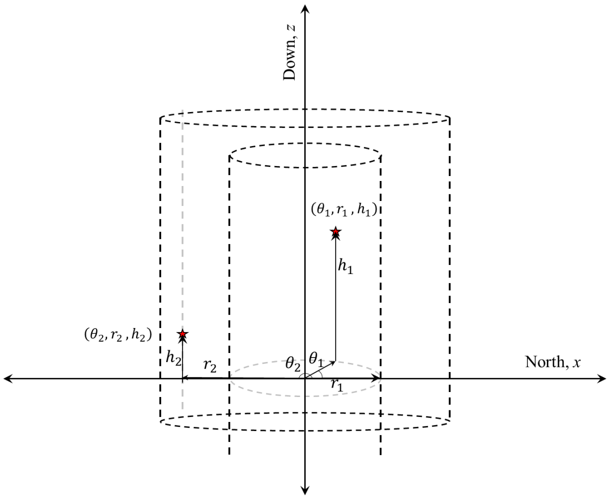


Fig 3. Cylindrical coordinate system used to match the actuation model of the AUV

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

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.

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.

$$D(v) = D_q(v) + D_l(v) \tag{5}$$

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

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

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

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.

$$r_G = [x_G \quad y_G \quad z_G]^T \tag{6}$$

$$r_B = [x_B \quad y_B \quad z_B]^T \tag{7}$$

B. State Space Formulation of the AUV

The equations of motion presented in the previous section are translated into a state space formulation. 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 Table 1.

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

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

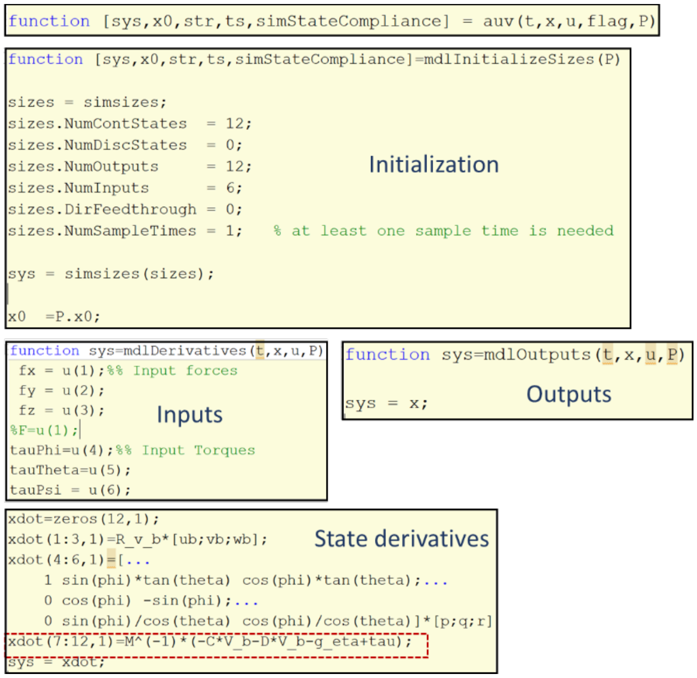
The state space equations are represented in a matrix form as below:

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

C. Modeling in Simulink

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 (Fig. 4). The S-function block (**auv**) interacts with the MATLAB function **auv.m** 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

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.



```
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;
```

```
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);
```

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

sys = x;
```

```
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;
```

Fig 4. 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

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.

D. 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 (Fig. 3) 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.

E. 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 (Fig. 5). 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.

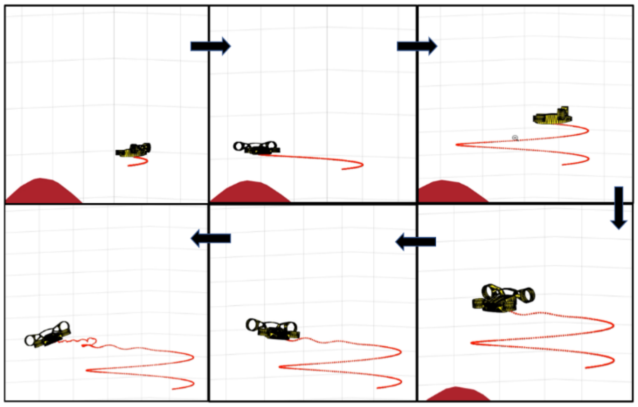


Fig 5. Unstable wobbly behavior of the AUV when it reaches the set point attributed to external torque disturbances about roll and pitch axis caused due to offset between the centers of buoyancy and gravity

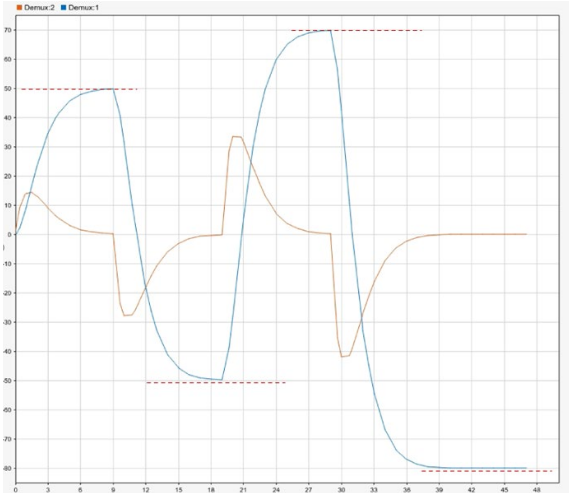
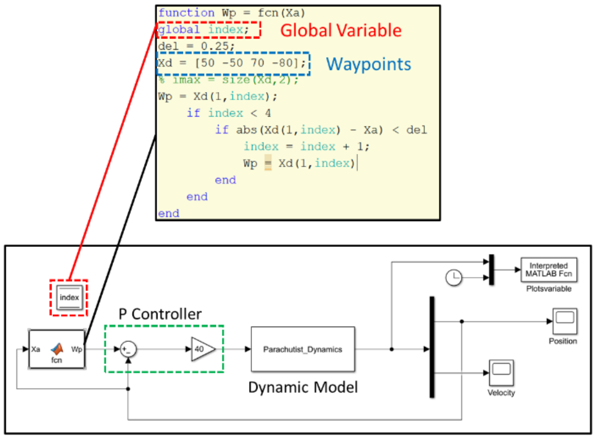


Fig 6. Waypoint planning of the one-dimensional parachutist system and tracking using a critically damped controller



```
function Wp = fcn(Xa)
global index;
del = 0.25;
Xd = [50 -50 70 -80];
% imax = size(Xd,2);
Wp = Xd(1,index);
    if index < 4
        if abs(Xd(1,index) - Xa) < del
            index = index + 1;
            Wp = Xd(1,index)
        end
    end
end
```

Fig 7. 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.

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.

F. *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 (Fig. 6). 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 (Figs. 7 & 8). 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.

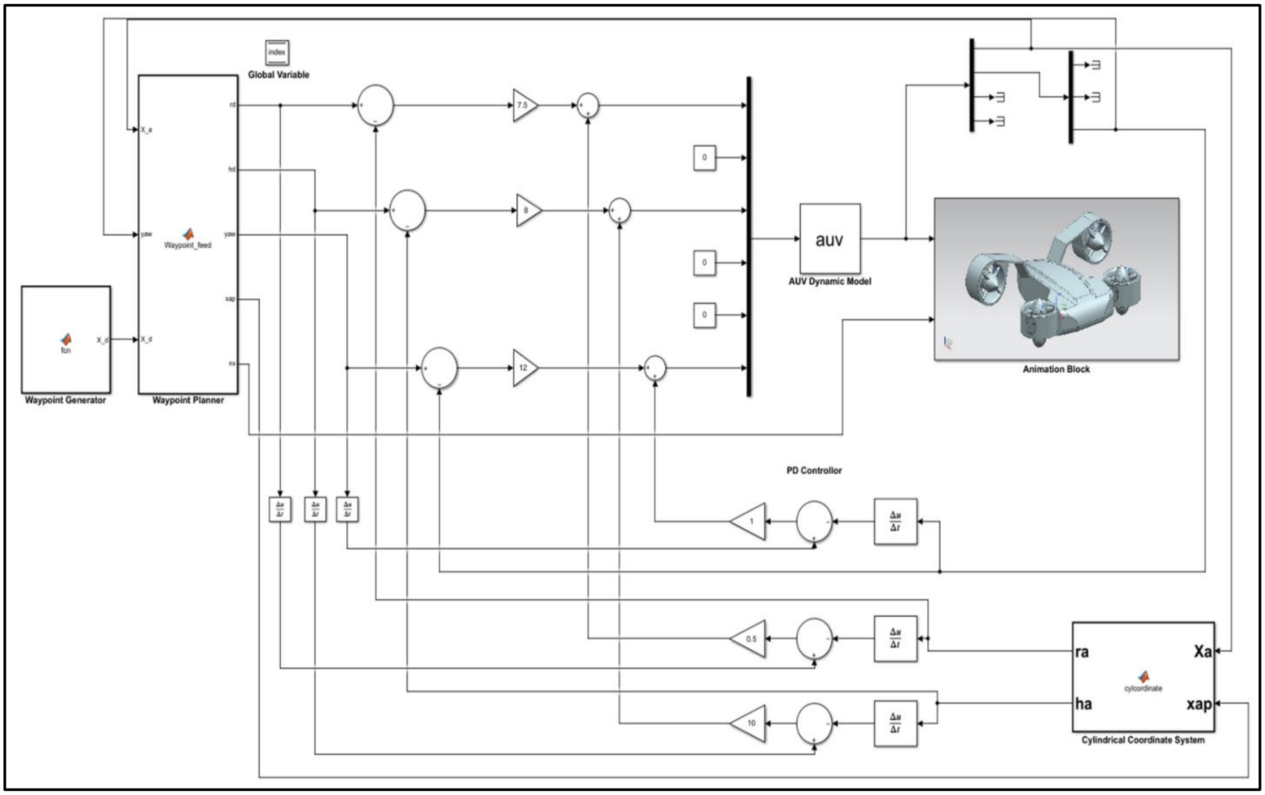


Fig 8. A full-blown Simulink Model of the AUV with integration between waypoint planner, controller, and animation modules

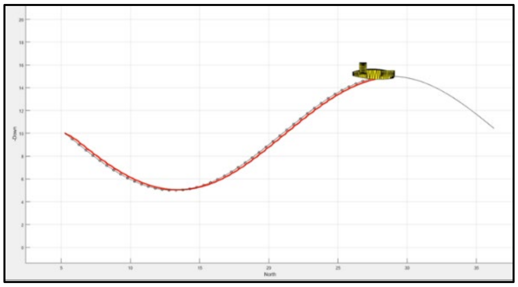


Fig 9. Snapshot from the AUV simulator visualization showing the AUV tracking a trajectory in XZ plane using surge and heave motions

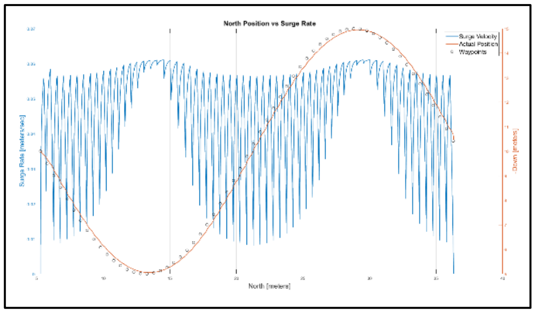


Fig 10. Surge rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV.

IV. 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

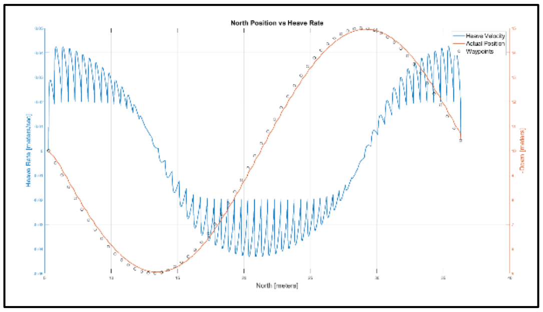


Fig 11. heave rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.

parametric analysis, (3) A* path planner implementation, (4) AUV velocity model.

A. Planar Motion Analysis

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. Snapshot from an animation while AUV is tracking its desired trajectory is shown in the Fig. 9. The behavior of the synthetic AUV with respect to the tracked

waypoints, its position and velocity are shown in Figs. 10 & 11.

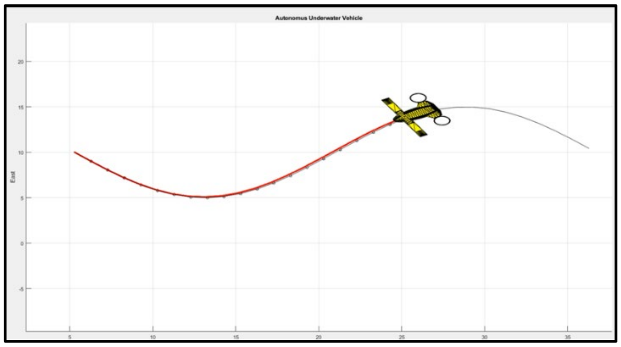


Fig 12. Snapshot from the AUV simulator visualization showing the AUV tracking a trajectory in XY plane using surge and heave as actuation motions.

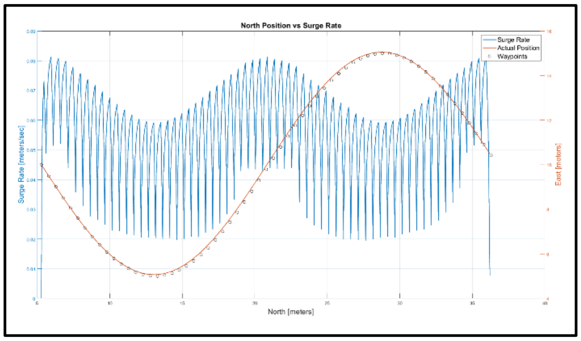


Fig 13. Surge rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

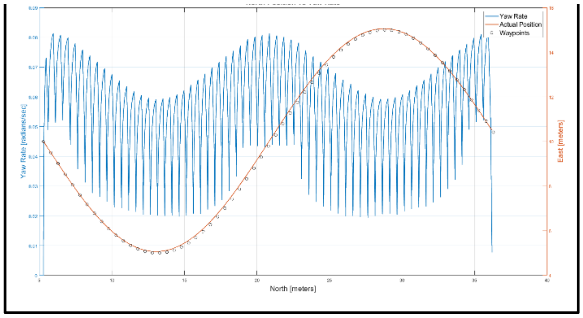


Fig 14. Yaw rate as a function of waypoints by using two different scales on the vertical axis and a plot of the actual path traversed by the AUV

XY Plane: The simulator has been tested 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. Snapshot from an animation while AUV is tracking its desired trajectory is shown in the Fig. 12. The behavior of the synthetic AUV with respect to the tracked waypoints, its position and velocity are shown in Figs. 13 & 14.

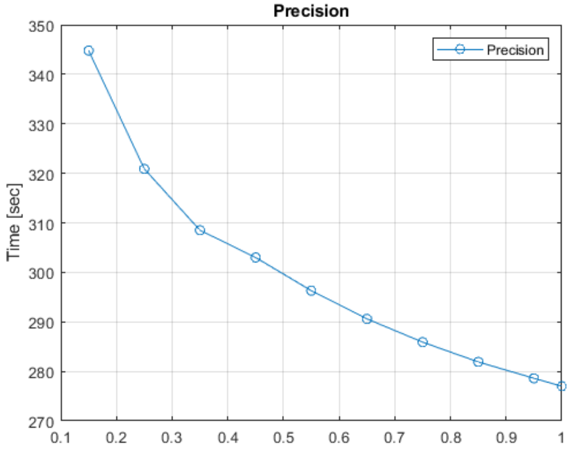


Fig 15. Analysis of the synthetic AUV with the study time taken to traverse along fixed path to finish the mission with different threshold settings ranging from 0.15 m to 1 m

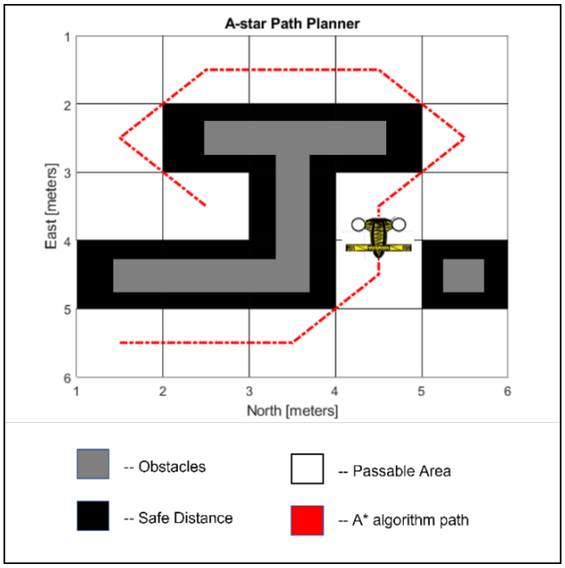


Fig 16. Snapshots from the AUV simulator visualization showing the synthetic AUV following a path generated by a A* planner

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).

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.

B. *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). Analysis of the synthetic AUV with the time taken to traverse along fixed path to finish the mission with different threshold settings ranging from 0.15 m to 1 m is shown in Fig. 15.

C. *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 Fig. 16.

D. *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 can maintain the desired user-defined velocities and follow a smooth curvy trajectory rather than stop and go motion. No motion of the AUV is constrained other than the underactuated motions (i.e., Roll, Pitch and Sway). A snapshot of AUV's motion using the velocity model is shown in Fig. 17.

E. *AUV Pipeline Inspection Task*

The simulator has given a set of waypoints in a fashion, such that is it able to follow the structure of a simple pipeline (Fig. 18). 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).

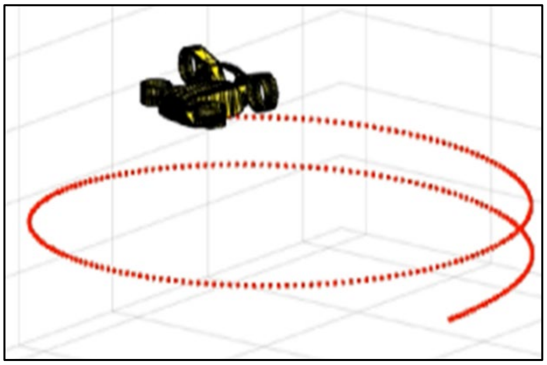


Fig 17. Synthetic AUV performing a kinematic velocity model using a underactuated model

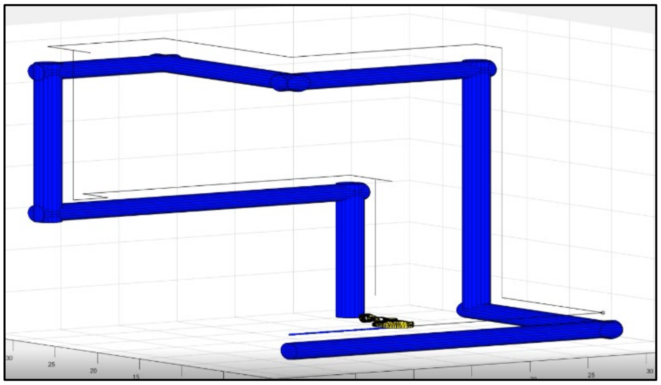


Fig 18. Snapshot from the AUV simulator visualization showing the AUV tracking a complex trajectory along a pipeline to illustrate its ability to inspect it using a underactuated model

V. Conclusions and Future Work

This paper 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. Some important directions for future work include: (1) Introducing a 3D path planner and testing the capability in 3D environment, (2) Investigating the effect of water currents on the AUV's trajectory response, (3) Comparing different controllers on the synthetic AUV, (4) Developing the dynamic model by considering an asymmetric design, where the CG and CB are not at the center of the vehicle and investigating how it impacts the stability of the AUV, (5) Building a physical AUV and analyzing the simulator data in conjunction with that of the physical version to validate the utility of the AUV simulator. With future improvements to the current design, this research is expected to contribute significantly to the field of autonomous underwater vehicles.

Acknowledgment

This work is supported partially by the Office of Naval Research (#N00014-19-1-2486), sub-contracted through University of Cincinnati (#012288-002). Such support does not necessarily constitute an endorsement by the sponsors of the opinions expressed by the authors in this paper.

References

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

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

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.

[3] C. Becker, R. David, and P. Ridao, “Simultaneous sonar beacon localization & AUV navigation," *IFAC Proceedings* Volumes 45.27 (2012): 200-205.

[4] A. Kim and R. Eustice, “Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009.

[5] D. Li, P. Wang, and L. Du, “Path planning technologies for autonomous underwater vehicles-a review,” *IEEE Access* 7 (2018): 9745-9768.

[6] B. Allotta et al., “A low-cost autonomous underwater vehicle for patrolling and monitoring,” In *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 231.3 (2017): 740-749.

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

[8] P. Yao and S. Zhao, “Three-dimensional path planning for AUV based on interfered fluid dynamical system under ocean current,” *IEEE Access* 6 (2018): 42904-42916.

[9] H. Cao, N. E. Brener, and S. S. Iyengar, “3D large grid route planner for the autonomous underwater vehicles*,” International Journal of Intelligent Computing and Cybernetics,* 2009.

[10] M. Panda et al., “A comprehensive review of path planning algorithms for autonomous underwater vehicles,” *International Journal of Automation and Computing* 17.3 (2020): 321-352

[11] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, “Anytime dynamic A*: an anytime, replanning algorithm,” In: *5th International Conference on Automated Planning and Scheduling* (ICAPS 2005), pp 262–271.

[12] J. Carsten, D. Ferguson, and A. Stentz, “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, 2006.

[13] C. Petres, Y. Pailhas, J. Evans, Y. Petillot, and D. Lane, “Underwater path planning using fast marching algorithms,” *Oceans Eur Conf* 2:814–819, 2005.

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

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

[16] P. Encarnacao and A. Pascoal, “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, 2001.

[17] C-W. Chen, J-S. Kouh, and J-F Tsai. "Modeling and simulation of an AUV simulator with guidance system," *IEEE Journal of Oceanic Engineering,* 38.2 (2013): 211-225.

[18] R. Cui, X. Zhang, and D. Cui, “Adaptive sliding-mode attitude control for autonomous underwater vehi-cles with input nonlinearities,” *Ocean Engineering*, 123, 45–54, 2016.

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

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

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

[22] T. Elmokadem, M. Zribi, and K. Youcef-Toumi, “Trajectory tracking sliding mode control of underactuated auvs,” *Nonlinear Dynamics*, 84(2), 1079–1091, 2016.

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

[24] Y. Li, Y. Q. Jiang, L. F. Wang, J. Cao, and G. C. Zhang, “Intelligent pid guidance control for auv path tracking,” *Journal of Central South University*, 22(9), 3440–3449, 2015.

[25] T. Gonzalez, J. A. Moreno, and L. Fridman, “Variable gain super-twisting sliding mode control,” *IEEE Transactions on Automatic Control*, 57, 2011.

[26] A. J. Healey and D. Lienard, “Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles,” *IEEE Journal of Oceanic Engineering* 18.3 (1993): 327-339.

[27] C. Silvestre et al., “A bottom-following preview controller for autonomous underwater vehicles,” *IEEE Transactions on Control Systems Technology*, 17.2 (2008): 257-266.

[28] D. Cook, A. Vardy, and R. Lewis, “A survey of AUV and robot simulators for multi-vehicle operations,” *IEEE/OES Autonomous Underwater Vehicles (AUV).* IEEE, 2014.

[29] P. Ridao et al., “Graphical simulators for AUV development,” *First International Symposium on Control, Communications and Signal Processing, IEEE*, 2004.

[30] M. Carreras et al., “An overview on behaviour-based methods for AUV control,” *IFAC Proceedings*, Volumes 33.21 (2000): 141-146.

[31] E. Omerdic, G. N. Roberts, and P. Ridao, “Fault detection and accommodation for ROVs,” *IFAC Proceedings*, Volumes 36.21 (2003): 127-132.

[32] O. Matsebe, 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.

[33] J.H.A.M. Vervoot, “Modeling and control of an unmanned underwater vehicle,” *Master Traineeship Report*, University of Canterbury, November 2008.

Authorized licensed use limited to: University of Obuda. Downloaded on October 03,2023 at 19:44:50 UTC from IEEE Xplore. Restrictions apply.