Abstract—Testing and gauging Autonomous Underwater Vehicle (AUV) missions through observations and experimental trials prove to be time consuming and costly. A system capable of verifying the AUV control models prior to practical test is needed. One of the key aspects in the presented framework is AUV modeling which mainly covers the area of AUV dynamics. Based on this designed simulation framework, custom scenarios provided by the user can be modeled and its corresponding 6DOF dynamics can be observed. The simulation framework is built based on a developed AUV test-bed which was jointly upgraded by DSTO and the University of Adelaide.

Keywords— Autonomous Underwater Vehicle (AUV), simulator, framework, robotics, maritime robot

I. INTRODUCTION

AUVs play a significant role in modern robotics. AUVs are involved in a number of maritime areas such as maritime security [4], oceanography [1] and submerged structure-inspection and maintenance. The main objective for this project is to develop a robust simulator capable of mimicking real-life AUV mission scenarios underwater. In order to achieve this, various knowledge of the physical aspects of the AUV such as the kinematics, dynamics, physical limitations, and environmental effects are required. This paper discusses on AUV 6DOF simulator framework and the core idea behind building this simulator.

The work in this paper was adapted from [7] and [9]. In [7], the author discusses the mathematical modeling of AUV kinematics while [9] introduces configurable AUV thruster modeling for an AUV with 4 thrusters. The work presented in this paper involves creating a simulation framework using 6DOF AUV dynamics with 6 thrusters configuration based on a real-world AUV.

One of the benefits of building a robust AUV simulator is that it provides a tool for further studies on AUV control systems. This is ideal for AUV control theory since it saves time and cost, where a designed control system can be observed and carried out using a simulated model rather than real-life mission testing.

The next section describes the physics, mathematical formulae and assumptions made in order to model a real-time AUV. The modeling criteria used in this project is based on a real world AUV developed by DSTO and The University of Adelaide, School of Mechanical Engineering. The following section describes the simulation protocol for a simple scenario, simulation results obtained and discussions. The final section concludes this paper and presents future work for simulator upgrades.

II. AUV MODELING

A. Background

The whole simulation framework developed is based on an existing AUV test-bed [8], as shown in Fig. 1. The AUV system consists of 4 separate modules:

- Inertial Navigation Sensors – Thrusters, GPS receiver, compass, accelerometer, pressure sensor, Doppler Velocity Log (DVL), and altimeter,
- Detection, Tracking and Identification Sensors – Sonar, and stereo vision camera.
- Communication system – Radio Frequency (RF) air to air modems, and underwater communication modems.
- Command and Control system – PC104 stack.

![Figure 1. AUV Robot test bed](image)

The AUV simulation framework is constructed using Matlab with the design structure shown in Fig. 2.
In the "desired motion" module, a user is required to key in a scenario, or desired final coordinates. This information will be used to calculate the required trajectory using cubic polynomials [6]:

\[
\begin{align*}
  x(t) &= a_0 + a_1t + a_2t^2 + a_3t^3 \\
  \dot{x}(t) &= a_1 + 2a_2t + 3a_3t^2 \\
  \ddot{x}(t) &= 2a_2 + 6a_3t 
\end{align*}
\]  

Where,

\[
\begin{align*}
  a_0 &= x_0 \\
  a_1 &= 0 \\
  a_2 &= \frac{3}{t_f}(x_f - x_0) \\
  a_3 &= -\frac{2}{t_f^2}(x_f - x_0)
\end{align*}
\]

The “Equation of Motion” module converts the required acceleration and velocities into the required forces using (3) and (5) discussed in the following section. The environmental forces and AUV physical limitations are implemented in the simulation framework but will not be discussed in this research paper. Based on these calculated forces and external forces from the environment, the required thruster forces can be calculated in the “Thruster Force Matrix” module. Applying these thruster forces based on configurations displayed in Fig. 4, the “Vehicle Model” module and “AUV Global Coordinate Updater” module are activated where the resultant acceleration vector, velocity vector and the AUV’s actual position in the simulated environment are calculated.

B. AUV Model

The AUV dynamic model is formulated using both a body fixed frame and a global frame. The body fixed frame describes the AUV motions, both rotational and translational as

\[
\dot{\mathbf{p}}_{AU V} = \left[ \mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z, \mathbf{\theta}, \mathbf{\phi}, \mathbf{\kappa} \right]^T
\]

where \(\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z\) and \(\mathbf{\theta}, \mathbf{\phi}, \mathbf{\kappa}\) represent the accelerations in x, y, and z-axis while \(\mathbf{\theta}, \mathbf{\phi}, \mathbf{\kappa}\) are the angular acceleration along x, y, and z-axis, as shown in Fig. 3.

1) Thruster configurations

![Figure 3. AUV local frame](image)

![Figure 4. Rear View of AUV with 6 Thrusters configuration](image)
The axes are chosen to coincide with the principal axes of inertia and they are defined as:

- longitudinal axis, $X_B$ (directed to front)
- transversal axis, $Y_B$ (directed to starboard)
- normal axis, $Z_B$ (directed down)

Horizontally configured thrusters, $HT^i$ and vertically configured thrusters, $VT^i$ exert thrusts (force) $T_{HT^i}$, $T_{VT^i}$ and torque (moment) $Q_{HT^i}$, $Q_{VT^i}$ respectively.

The position vector, $r_{HT^i} = [x, y, z]^T_{HT^i}$ and $r_{VT^i} = [x, y, z]^T_{VT^i}$ describe the position of the point of attack of the forces $T_{HT^i}$ and $T_{VT^i}$, relative to the base frame.

The moments due to the forces are $Q_{HT^i} = r_{HT^i} \times T_{HT^i}$ and $Q_{VT^i} = r_{VT^i} \times T_{VT^i}$.

The orientation of horizontal and vertical thrusters $HT^i$ and $VT^i$ relative to base frame is defined by the unit vectors $e_{HT^i} = [e_x, e_y, e_z]^T_{HT^i}$ and $e_{VT^i} = [e_x, e_y, e_z]^T_{VT^i}$.

The vectors $e_{HT^i}$ and $e_{VT^i}$ represent the positive direction of the force $T_{HT^i}$ and $T_{VT^i}$ respectively. This means that if the angular velocity of the propeller is positive (counter clockwise, according to right hand rule), it will exert force $T_{HT^i}$ and $T_{VT^i}$ in the direction of $e_{HT^i}$ and $e_{VT^i}$ respectively. The thruster force matrix can then be formulated as (4).

\[
(f, \tau) = \begin{bmatrix}
T_i \\
Q_i
\end{bmatrix} =\begin{bmatrix}
e_x \\
e_y \\
e_z
\end{bmatrix}
\begin{bmatrix}
r_x \\
r_y \\
r_z
\end{bmatrix}
\begin{bmatrix}
e_x \\
e_y \\
e_z
\end{bmatrix}
\begin{bmatrix}
T_{HT^i} \\
T_{VT^i}
\end{bmatrix}
(4)
\]

$(f, \tau)_i$ is the total vector of propulsion forces $(f_x, f_y, f_z)$ and moments $(\tau_x, \tau_y, \tau_z)$.

$Q_i$ is the Moment Vector generated by each thrusters

$T_i$ is the Force Vector generated by each thrusters

$e_i$ is the Orientation Vector of the Thrusters

$(r_i \times e_i)$ is the cross product between the Positional Vector and the Orientation Vector

The superposition of each individual contribution $(f, \tau)$ leads to total vector propulsion forces and moments [3]:

\[
(f, \tau) = \begin{bmatrix}
f_x \\
f_y \\
f_z \\
\tau_x \\
\tau_y \\
\tau_z
\end{bmatrix}
\begin{bmatrix}
e_x \\
e_y \\
e_z \\
e_x \\
e_y \\
e_z
\end{bmatrix}
\begin{bmatrix}
T_{HT^i} \\
T_{VT^i}
\end{bmatrix}
(5)
\]
### TABLE I. SUMMARY OF FORCES AND TORQUES GENERATED BY THRUSTERS

<table>
<thead>
<tr>
<th>Forces, $f_{TH}$</th>
<th>$f_x$</th>
<th>$T_{HT_1} + T_{HT_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_y$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_z$</td>
<td>$T_{VT_1} + T_{VT_2} + T_{VT_3} + T_{VT_4}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Torques, $\tau_{TH}$</th>
<th>$\tau_x$ (Roll)</th>
<th>$-r_y T_{VT_1} + r_y T_{VT_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_y$ (Pitch)</td>
<td>$-r_{\text{hop}} T_{VT_1} + r_{\text{hop}} T_{VT_4}$</td>
<td></td>
</tr>
<tr>
<td>$\tau_z$ (Yaw)</td>
<td>$r_{\text{by}} T_{HT_1} - r_{\text{by}} T_{HT_2}$</td>
<td></td>
</tr>
</tbody>
</table>

### 2) Equation of motion in free space

The motion of the AUV is comprised of both the translational and rotational components. Using Newton’s second law of motion, the translational component can be derived by considering the location of a body-fixed frame located at the AUV’s centre of mass. As for the rotational component, the formula was derived based on total applied moments about the AUV’s centre of mass. In the derivation of the AUV’s equation of motion, some basic assumptions were made, they are:

- AUV has constant mass and inertia tensor
- The formula takes account of axes not coinciding with the AUV’s principal axis of inertia, therefore, the product of inertia is non-zero

The final form of the AUV equation of motion can be simplified as seen in (6):

$$M_{RB} \ddot{V} + C_{RB}(x)\dot{V} = f_{\text{total}}$$

$$M_{RB} =
\begin{bmatrix}
  m & 0 & 0 & 0 & m z_{CoG} & -m y_{CoG} \\
  0 & m & 0 & -m z_{CoG} & 0 & m x_{CoG} \\
  0 & 0 & m & m y_{CoG} & -m x_{CoG} & 0 \\
  0 & -m z_{CoG} & m y_{CoG} & I_{xx} & I_{xy} & I_{xz} \\
  m z_{CoG} & 0 & -m x_{CoG} & I_{xy} & I_{yy} & I_{yz} \\
  -m y_{CoG} & m x_{CoG} & 0 & I_{xz} & I_{yz} & I_{zz}
\end{bmatrix}$$

$$C_{RB}(x) =
\begin{bmatrix}
  0 & -m \kappa & m \phi & m (y_{CoG} \phi + z_{CoG} \kappa) & -m x_{CoG} \phi & -m x_{CoG} \kappa \\
  m \kappa & 0 & -m \theta & -m y_{CoG} \phi & m (z_{CoG} \kappa + x_{CoG} \phi) & -m y_{CoG} \kappa \\
  -m \phi & m \theta & 0 & -m z_{CoG} \phi & -m z_{CoG} \theta & m (x_{CoG} \phi + y_{CoG} \theta) \\
  -m (y_{CoG} \phi + z_{CoG} \kappa) & m y_{CoG} \phi & m z_{CoG} \theta & 0 & -I_{xx} \phi + I_{xz} \theta + I_{xz} \kappa & I_{xx} \kappa + I_{xz} \phi \\
  m x_{CoG} \phi & -m z_{CoG} \kappa + x_{CoG} \phi & m z_{CoG} \phi & I_{xz} \phi + I_{xz} \theta - I_{xz} \kappa & 0 & -I_{xz} \kappa + I_{xy} \phi + I_{xz} \theta \\
  m x_{CoG} \phi & m x_{CoG} \phi & -m (x_{CoG} \phi + y_{CoG} \theta) & -I_{xz} \kappa - I_{xy} \theta + I_{xz} \phi & I_{xz} \kappa + I_{xy} \phi & 0
\end{bmatrix}$$
\( M_{RB} \) represents the rigid body mass matrix and \( C_{RB}(x) \) represents the state dependent matrix containing the rigid body coriolis and centrifugal terms. The right hand side of (6) is the vector sum of forces induced by AUV thrust forces and also the external environmental forces and moments. These external forces and moments are composed of hydrostatic and hydrodynamic forces which will not be discussed in this paper.

III. SIMULATION PROTOCOL AND RESULTS

In this section, the simulation involves running a simple trajectory following AUV. This simulation shows the implementation of 6DOF dynamics together with the AUV thruster control system. In the display in Fig. 6, a 3D global map is generated together with the AUV velocities and the thruster outputs can be seen.

The simulator starts with the input of desired goal point and the time required to finish the task. The simulator then generates corresponding translational and rotational part of both the velocity and acceleration components. Applying the equation of motion converts these accelerations and velocities into required forces. The next step combines these required forces with the environmental forces, which in this case is zero, to compute the resultant forces and moments in \( x, y, \) and \( z \) direction. This resultant force matrix \( \begin{bmatrix} f_{x}, f_{y}, f_{z}, \tau_{x}, \tau_{y}, \tau_{z} \end{bmatrix}_{\text{total}} \) describes the actual forces acting on the AUV body which cause the motion.

Using the modified equation of motion, (7), and the resultant acceleration matrix can be calculated.

\[
\dot{V} = M_{RB}^{-1} \left( f_{\text{total}} - C_{RB}(x)V \right)
\]

Since the terms known are \( f_{\text{total}} \) for the next time step or the forces which will induce the AUV motion, and the AUV position information from previous time step, \( M_{RB} \) and \( C_{RB}(x) \) can be calculated. However, since both \( V \) and \( \dot{V} \) are not known, \( V \) from previous time step must be used to resolve (7). This will cause the simulator to produce a small error term where the desired motion and the actual motion will have a small “time lag” whereby it can be reduced if the time step used to simulate the scenario is set to a small value. As for this paper, the time step is set to very small value and the error can eliminated. Fig. 7 shows a typical diving scenario with its corresponding horizontal thruster outputs observed in Fig. 8 and vertical thruster outputs observed in Fig. 9.

![Figure 6. MATLAB GUI layout](image-url)
IV. CONCLUSION AND FUTURE WORK

The 6DOF AUV simulator developed in Matlab using only the AUV model is completed. By applying various mathematical models, the AUV motion based on any given scenario can be traced and its thruster output identified.

Future development of the simulator involves three stages. The first stage of this simulator development involves adding environmental forces such as hydrostatic, hydrodynamics forces, and ocean currents into the system. The second stage involves modeling of the sensors for localization and obstacle avoidance underwater. The final stage is to compare the simulation with real world AUV in a controlled environment. Once these developments are completed, the simulator can be used for preplanned AUV missions, mission diagnostic as well as control system testing.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial and kind support received from DSTO (MOD) and The University of Adelaide, School of Mechanical Engineering. Also, a special thanks to fellow postgraduate student, Tommie Liddy for his programming help and encouragement.

REFERENCES