IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 11, Issue 4 Ver. I (Jul. – Aug. 2016), PP 151-158
www.iosrjournals.org

Path Planning and Open-Loop Control Algorithms for a Differential Thrust Autonomous Underwater Vehicle

Mohammad Hasankashefi[1], Farhad Bolouri[2], Keivan Bolouri[3]
[1]Department of Electronic Engineering, Islamic Azad University South Tehran Branch, Tehran, Iran
[2]Department of Electronic Engineering, Islamic Azad University Central Tehran Branch, Tehran, Iran
[3]Department of Mechanical Engineering, the University of Adelaide, Adelaide, Australia

***Abstract**: The world's interest in underwater applications of unmanned vehicles has experienced a significant increase in recent years. The hazardous and erratic conditions of underwater missions along with their cost and the environmentally hazardous effects of manned underwater vehicles, contribute to the desire to use smaller, more energy efficient, relatively cheap and environmentally friendly AUVs. This paper presents a novel differential thrust control method of an AUV that attempts to broaden their applicability through design for complete Automation in addition to development of a three dimensional path generation tool which provides the trajectory data.*

***Keywords**: autonomous underwater vehicles, differential thrust, Thruster control method, trajectory generating GUI*

I. Introduction

Oceanic research and exploration missions around the globe are carried out through the use of Autonomous Underwater Vehicles (AUVs) developed to deliver various functions and features. Based on their mission, the AUVs will be designed or modified to meet a series of specific requirements necessary for their missions. AUVs have shown great potential and promise in various industries, such as oil and gas industries, fishery, oceanography, etc. [1]. Although they may be fairly low-cost compared to the manned underwater vehicles, the fact remains that considering the implemented technology, their somewhat intricate internal system in addition to the cost of maintenance, they can be rather hard to afford for small businesses which leads to restrictions on their popularity amongst all maritime industries.

This projects AUV is required to inspect an off-shore wave power generation platform on a regular basis and is preferred to be of small size, low cost and good maneuverability. The AUV has 3 radial thrusters positioned 120 degrees apart from each other. .They are controlled via motor controllers connected to an embedded PC enclosed within the vehicle. The AUV uses a unique Differential Thrust drive-system to steer through waypoints. The aim of this study is to develop a Simulink model that can predict the differential thrust values for the thrusters in order to propel the vehicle in a desired 3 dimensional trajectory.

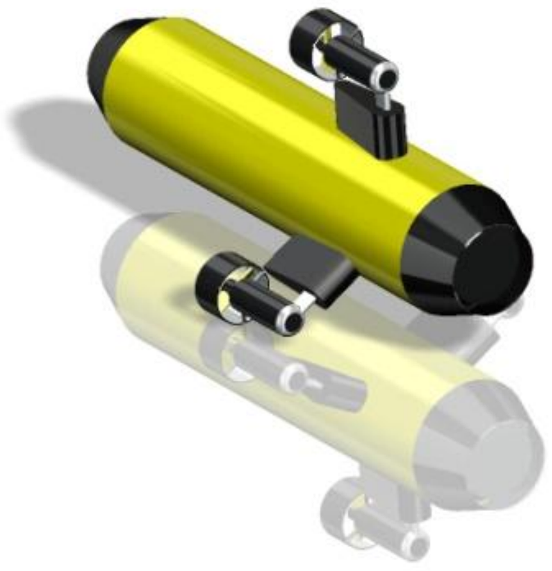


Fig1: The AUV's unique form

II. Design Features

The project's setup structure of components and devices is demonstrated in Fig.

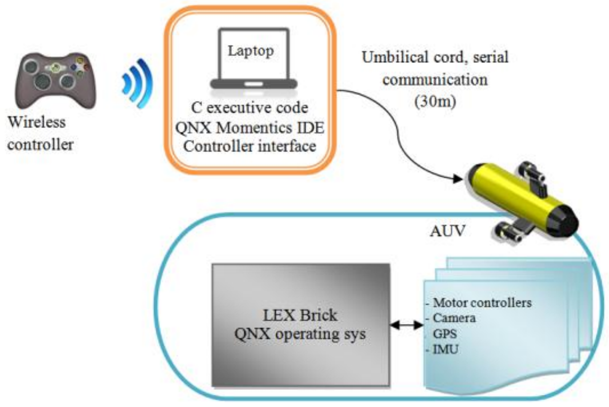


Fig2: The control system layout

The controller is wirelessly connected to the laptop. The application runs a C code which interprets the controller's inputs and transfers to the embedded PC inside the vehicle via an umbilical cord using a RS232 serial interface. Additional Linux based interface-modules are run on the laptop allowing it to communicate with the LEX brick (the embedded PC) running a QNX Momentum's Integrated Development Environment (IDE) real-time operating system. The LEX brick will be loaded with the data from the PC and will send the proper commands to motor controller units.

As the AUV uses Differential thrust drive system to steer through waypoints, a tabulated list of differential thrust values corresponding to a trajectory are needed in a timely manner for the AUV to use as a look up table. This table will be generated via the designed Simulink model for each desired path, and will be included in the C code running the AUV.

III. Methodology

It is imperative to develop an accurate dynamic model of the AUV. The more accurate dynamic model yields closer results to reality.This will help validating the results with field tests quicker. Moreover, helps developing the foundations of a control law for the AUV. Fully understanding the vehicles dynamics and properly designing a controller, assures effective and efficient maneuvering and thus results in battery preservation [4]. The aim is to optimize the dynamic model of the vehicle by getting feedback from a set of open-loop maneuvers performed in the test field. In this approach, first the dynamic model of the AUV is developed from theory and laboratory tests and then, using this initial model a Simulink model is developed. The Aim of this model is to generate time structured thruster values corresponding to a desired trajectory which the AUV requires to follow. These thruster values if given to the AUV through time will ideally cause the AUV to follow the desired trajectory. In reality, the device will have deviations from the path since the dynamic model is not initially accurate. Through the data acquired from the test results, the initial dynamic model is updated and fine-tuned. The process can be explained in the block diagram given in Figure .

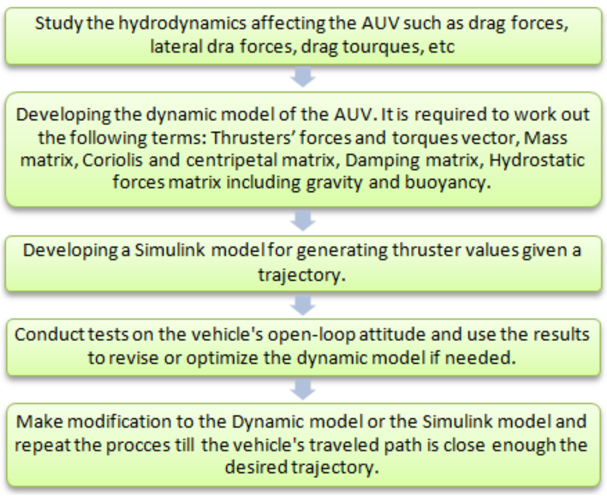

Study the hydrodynamics affecting the AUV such as drag forces, lateral dra forces, drag tourques, etc
Developing the dynamic model of the AUV. It is required to work out the following terms: Thrusters' forces and torques vector, Mass matrix, Coriolis and centripetal matrix, Damping matrix, Hydrostatic forces matrix including gravity and buoyancy.
Developing a Simulink model for generating thruster values given a trajectory.
Conduct tests on the vehicle's open-loop attitude and use the results to revise or optimize the dynamic model if needed.
Make modification to the Dynamic model or the Simulink model and repeat the procces till the vehicle's traveled path is close enough the desired trajectory.

Figure 3: The methodology.

In order for a trajectory to be appropriate for use in this study, it must be smooth, continuous and three times differentiable with respect to time [2]. The trajectory data consisting the translational and angular velocities and accelerations, will be given to the Thruster value generator Simulink model. Any discontinuity in the data will cause errors in the process. Therefore it is eminent that the trajectory satisfies the cited criteria. As part of this study a trajectory or path generation tool with graphic user interface was developed. The purpose of this tool is to provide the user with an interactive interface to quickly approximate a path given a set of waypoints and then modify it in real-time. The tools interface is designed for convenience of use and ease of conception. The results of this tool are the trajectory data which are provided for the thruster value generating Simulink model.

Developing the path generation tool

Matlab's graphical user interface has been used for the development of the path generation tool. The tool is designed to generate

Data (location, translational/angular velocities and accelerations given the location of waypoints and their corresponding velocities. The GUI will first produce a trajectory for a set of waypoints at a constant depth producing a planar trajectory and later, the desired depth at each waypoint can be defined via the depth control feature. The path is created from multiple Bezier curves between waypoints.

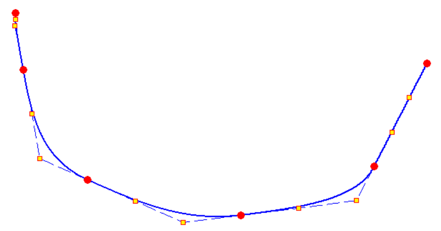


Fig4: Formation of the path via Bezier curves

An algorithm was developed to initially place the control points of the Bezier curves (which control the slope of the curve at the end points) to make the generated curve balanced as can be seen in Fig.

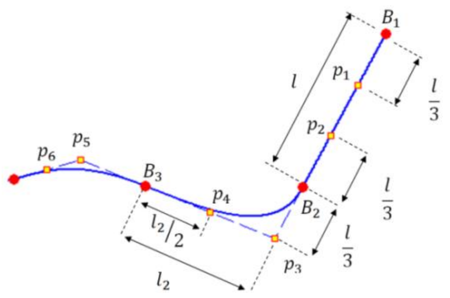


Fig5: Placement of the control points

Preserving the slope of the curves at the blending points ensures the smoothness of the path after modification. An algorithm has been devised to preserve the slope of the path at blending points. The method has been illustrated in Fig.

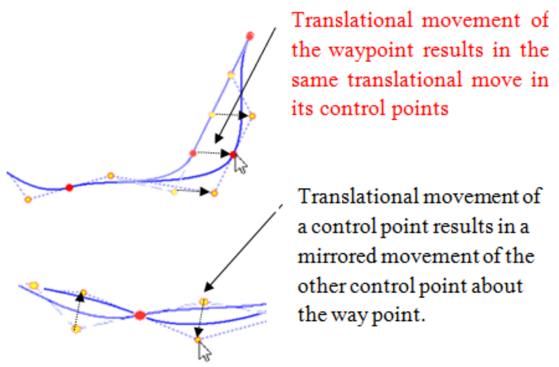

Translational movement of the waypoint results in the same translational move in its control points
Translational movement of a control point results in a mirrored movement of the other control point about the way point.

Fig6: The method devised to preserve the slope at the blending points.

To complete the trajectory and to be able to extract the trajectory data such as velocities and accelerations, a time structure should be developed for the path. The total speed of the vehicle at one way point will gradually reach the total speed at the next waypoint. By knowing the total speed of the vehicle at each point throughout the path, the time structure can be worked out for the whole path.

After developing the time structure for the path, the trajectory is ready to be used. The path generation GUI tool is shown in Figure 1.

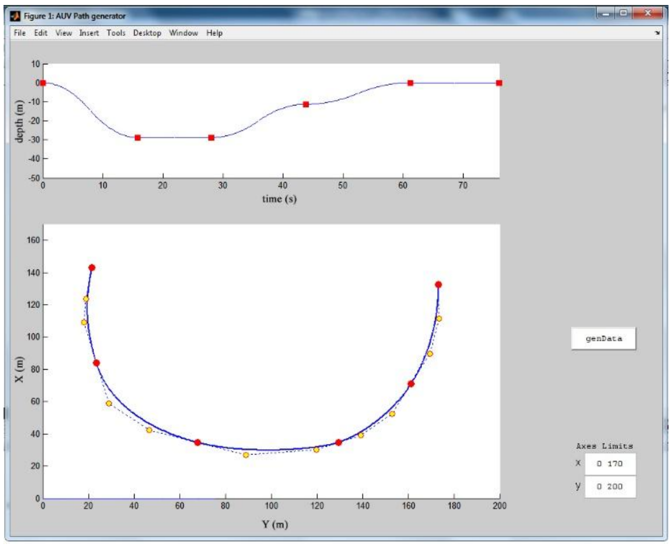


Figure 1: AUV Path generator
File Edit View Insert Tools Desktop Window Help
depth (m)
time (s)
X (m)
Y (m)
genData
Axes Limits
X 0 170
Y 0 200

Figure 1: Interface of the GUI

After inputting the waypoints in to the tool by clicking on the desired locations on the screen, the tool will generate a path by automatically creating the control points. The depth control feature will also become available and a pre-assign depth of zero will be given to all the waypoints initially. The tool enables the user to modify all the features such as location of the waypoints/control points, depth/velocity at each waypoint and range of the graph's axes if needed.

Developing the thruster value generating Simulink model

The Aim of this Simulink model can be summarized in the form of the following question: if all the trajectory data (such as time structure, translational and angular velocities, translational and angular accelerations) at all the trajectory points were known, would it be possible to predict the thrusters' values required for the AUV to follow the trajectory when given those thruster values? The problem borders on an inverse kinematics problem. By rewriting the governing dynamic equations of motion for the vehicle and solving them for the global forces and torques, ultimately the thruster values can be calculated by incorporating all the dynamic/hydrodynamic forces and torques. The dynamic model of the AUV has been provided in a separate publication [3]. Using the same principles stated in [3], the Thruster Value GeneratingSimulink model is constructed.

The Block diagram of the developed Simulink model is given in Figure .

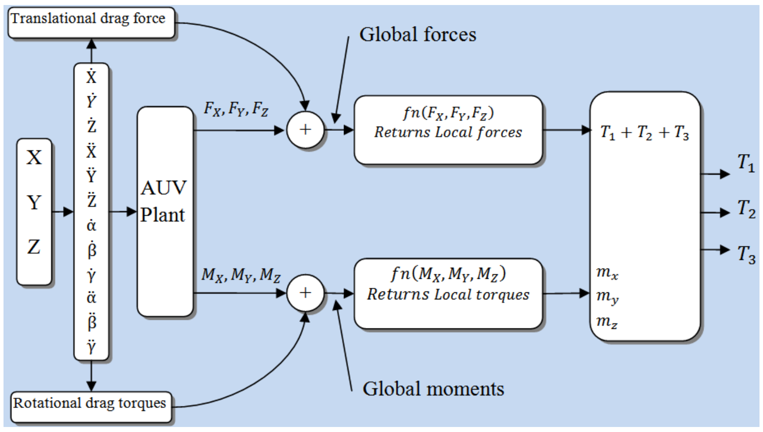

Translational drag force
Global forces
X
Y
Z
$\dot{X}$ $\dot{Y}$ $\dot{Z}$ $\ddot{X}$ $\ddot{Y}$ $\ddot{Z}$ $\dot{\alpha}$ $\dot{\beta}$ $\dot{\gamma}$ $\ddot{\alpha}$ $\ddot{\beta}$ $\ddot{\gamma}$
AUV Plant
F_X, F_Y, F_Z
$fn(F_X, F_Y, F_Z)$ Returns Local forces
$T_1 + T_2 + T_3$
T_1
T_2
T_3
M_X, M_Y, M_Z
$fn(M_X, M_Y, M_Z)$ Returns Local torques
m_x m_y m_z
Global moments
Rotational drag torques

Figure 8: Block diagram of the Simulink model.

The XYZ block represents the properties of the path in terms of its X, Y, and Z location data as a function of time. From this data, the global translational and rotational velocities will be calculated along with the global accelerations and the angles.
The AUV plant can be written in matrix form in the following manner.

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \times \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \\ \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \end{bmatrix} \tag{1}$$

The equations for the global forces and torques are defined In The Matrix equation (1) in which $\ddot{X},\ddot{Y},\ddot{Z}$denote the global accelerations, $\ddot{\alpha},\ddot{\beta},\ddot{\gamma}$ are the angular accelerations, m is the vehicle's mass, and$I_{xx}, I_{yy}, I_{zz}$are the second moments with respect to x, y, and z axis respectively. The forces and moments caused by the vehicle's accelerations are simply calculated by having the angular and translational accelerations and the vehicles properties (mass and second moment inertias). The next step is to add the force and moments caused by the drag forces to the forces and moments calculated in the previous step in order to get the global forces and moments.
The local translational drag forces are calculated as follows:

$d_x = \frac{1}{2}\rho . c_{d_x} \cdot A_{f_x} \cdot u^2$ (2)

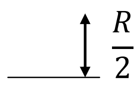
$d_y = \frac{1}{2}\rho . c_{d_y} \cdot A_{f_y} \cdot v^2$ (3)

$d_z = \frac{1}{2}\rho . c_{d_x} \cdot A_{f_y} \cdot w^2$ (4)

Local velocities must be used in the above equations to calculate the local drag. To get the local velocities, the global velocities must simply be multiplied by a transformation matrix.

To get the global drag, similarly, the local drag must be multiplied by the transformation matrix $\mathbf{J_1}$ given in equation 5.

$$\mathbf{J_1} = \begin{bmatrix} \cos\gamma\cos\beta & -\sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha & \sin\gamma\sin\alpha + \cos\gamma\sin\beta\cos\alpha \\ \sin\gamma\cos\beta & \cos\gamma\cos\alpha + \sin\gamma\sin\beta\sin\alpha & -\cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix} \tag{5}$$



$$\mathbf{J_2} = \begin{bmatrix} 1 & \sin\alpha\tan\beta & \cos\alpha\tan\beta \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \frac{\sin\alpha}{\cos\beta} & \frac{\cos\alpha}{\cos\beta} \end{bmatrix} \tag{6}$$

Fig9: Distances from the Cob

$$\begin{bmatrix} D_X \\ D_Y \\ D_Z \end{bmatrix} = \mathbf{J_1}^{-1} \cdot \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \tag{7}$$

The equivalent global drag torques can be calculated as follows:

$T_{d_x} = 0 \ (no angular speed about x axis)$ (8)

$T_{d_y} = \frac{1}{2} \cdot \rho \cdot c_{dc} \cdot A_{fc} \cdot l_c^3 \cdot \dot{\beta}^2$ (9)

$T_{d_z} = \frac{1}{2} \cdot \rho \cdot c_{dc} \cdot A_{fc} \cdot l_c^3 \cdot \dot{\gamma}^2$ (10)

In the next step, the drag forces and drag moments are added to the calculated forces in equation 1 and torques respectively. The overall forces and torques will then be given to the two function blocks which will calculate the local force and torque values.

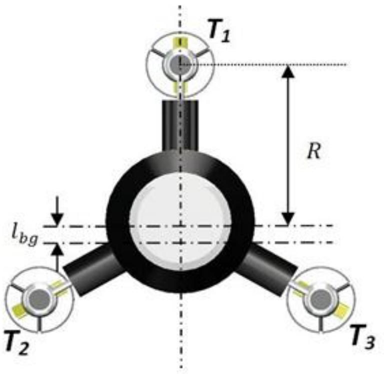
Function 1: converts the global forces in to local forces. In the local system, forces only exist in the X direction which is along the vehicles longitudinal axis and is the summation of the thrusters' values $(T_1 + T_2 + T_3)$.

$$\begin{bmatrix} T_1 + T_2 + T_3 \\ 0 \\ 0 \end{bmatrix} = fn(F_X, F_Y, F_Z) = \mathbf{J_1}^{-1} \cdot \begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix} \tag{11}$$

Function 2: converts the global moments in to local moments. In the local system there are no forces that can cause a moment about the x axis therefore m_x is equal to zero.

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} 0 \\ m_y \\ m_z \end{bmatrix} = fn(M_X, M_Y, M_Z) = \mathbf{J_2}^{-1} \cdot \begin{bmatrix} M_X \\ M_Y \\ M_Z \end{bmatrix} (12)$$

The last block in the block diagram receives three values corresponding to three equations $(T_1 + T_2 + T_3, m_y$ and $m_z)$. The three equations are solved for T_1, T_2 and T_3 which are the thruster values in Newtons.



According to **Error! Reference source not found.**, the equations for m_y and m_z can be calculated. The three equations for Total thrust, m_y *and* m_z are given below:

$$Total\ thrust_{in\ x\ direction} = T_1 + T_2 + T_3 \ (13)$$

$$m_y = T_1 \cdot R - (T_2 + T_3) \cdot \left(\frac{R}{2}\right) (14)$$

$$m_z = (T_2 - T_3) \cdot \frac{R \cdot \sqrt{3}}{2} (15)$$

By solving these equations at each time step for T_1, T_2 and T_3, the thruster values can be saved as structures with time. By providing the AUV with these thruster values, it will move on a trajectory which hopefully will be close to the same trajectory that led to the creation of these thruster values.

IV. Simulation Of The Vehicle

In this section, thruster values resulted from a created trajectory will be given to the AUV. The AUV will then take a path corresponding to the thruster values. The resulted path will then be compared with the fist path to if any deviations have occurred. The type of the deviation and its degree will help revising the dynamic model. The results from the final revised model are shown here.

The path given in **Error! Reference source not found.** is created by the GUI tool for the test. The AUV is set to reach *{0 4 4 4 3 0} m/s* velocities and *{0 0 -8 -8 -8 0} m* depth at given waypoint shown in red.

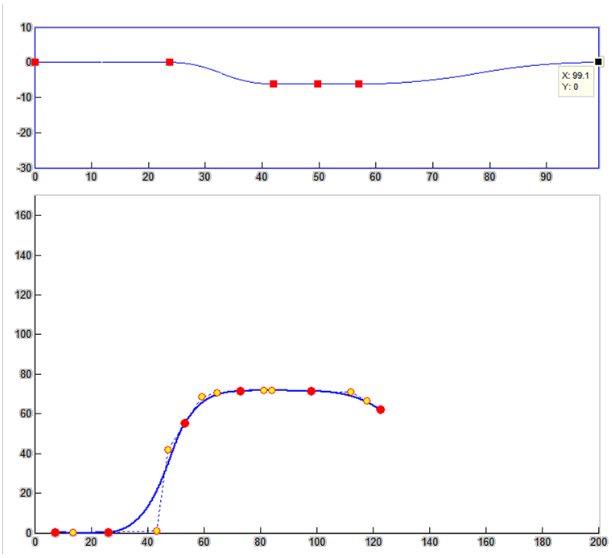


Fig 10: Path generated for test 3

The resultant thruster values as a function of time are given in the figures bellow.

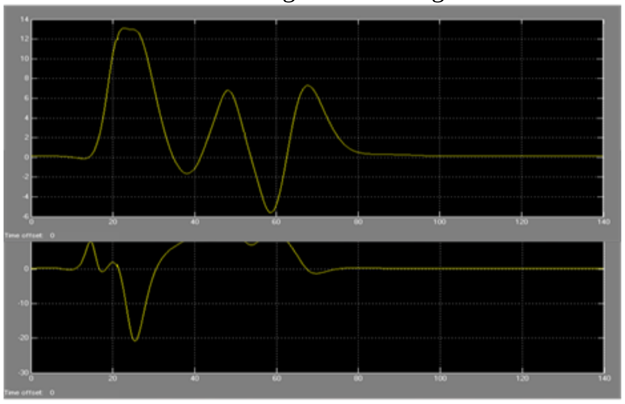


Fig 11: Thruster values calculated for the path in test 3, from the top: the top thruster, the left thruster and the right thruster value in Newton.

As can be seen in **Error! Reference source not found.**, the right thruster's value rises and the left thruster's value decreases when doing a left turn and vice versa. Thruster values are then saved to the workspace. The values are then given to an AUV simulation and it will simulate the vehicles behavior according to the input thruster values. The resulted trajectory is shown in **Error! Reference source not found.**.

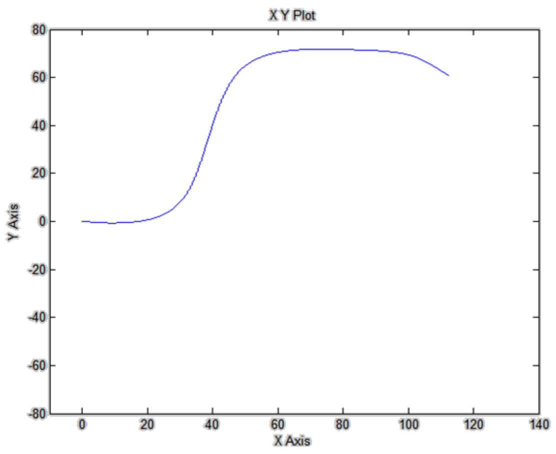


Fig12: The AUV's simulated trajectory according to the input thruster values

By putting together the two paths, it can be seen that the simulated path is fairly close to the path generated by the GUI (Fig13: Desired path compared to the simulation's result).

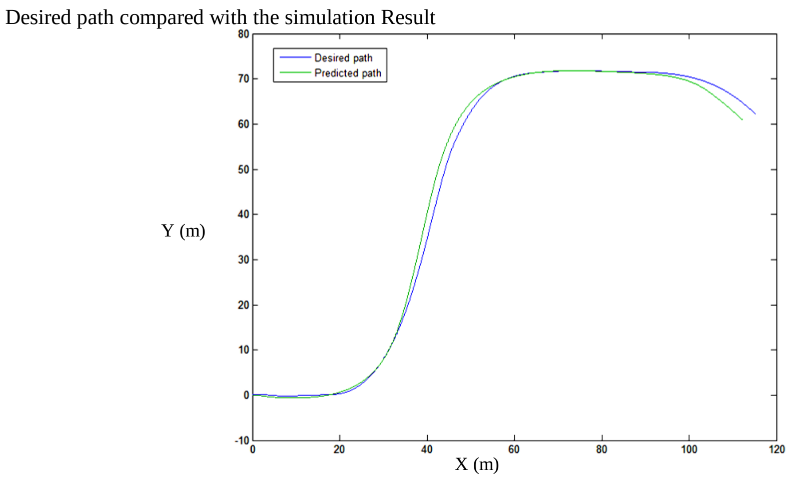


Fig13: Desired path compared to the simulation's result

The depth calculated by the trajectory prediction Simulink model is presented in **Error! Reference source not found.**. The depth at each waypoint was desired to be *{0 0 -8 -8 -8 0}* meters but as it can be seen in the figure, the AUV's simulated movement does not follow the depth correctly.

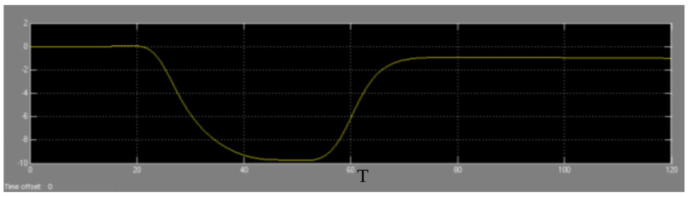


Fig14: Simulation result's depth

The cause of the slight deviation from the path is the vehicle's inability to comply with the desired movements. Either the velocities at the beginning of the curves were too high or the rate of changes in the angles was too much for the vehicles dynamic to handle.

V. Conclusion And Future Work

This project concerns the development of a thrusters control method and a path generation algorithm for an AUV. A path generation GUI was developed for the purpose of creating three dimensional trajectories suitable for the AUV. In order for that to happen, a set of way points and their corresponding depth and speed must be provided for the tool. An algorithm was devised to generate a path from the waypoints and their data. As a result, the developed GUI is fully capable of generating planar and three dimensional trajectories as a function of time.

The developed Simulink model of the AUV was to use the trajectory data resulting from the GUI tool and to output the thruster values as a function of time. As long as the trajectory is consistent with the vehicles dynamics and manoeuvrability, rational thrusters values for the thrusters can be obtained from the model. The developed Simulink model results were very promising according to the number of tests done. The thruster values generated by this model can be given to the AUV in the form of a lookup table and the AUV will follow the path associated to those thruster values with least deviation.

The next step is to modify the developed thrusters control method to incorporate the feedbacks from IMU and GPS systems for path correction. After completion, the AUV can travel the path which contains waypoints in which the vehicle resurfaces and after getting data from the GPS, corrects its path and continues with its mission under the water.

The model cannot handle paths which contain a slope of 90 degrees. This matter can also be worked on in future works.

References

[1]. J. Kim; K. Kim; Choi, H.S; WoojaeSeong; Kyu-Yeul Lee, "Estimation of hydrodynamic coefficients for an AUV using nonlinear observers", IEEE Journal of Oceanic Engineering, ISSN 0364-9059, 10/2002, Volume 27, Issue 4, pp. 830 – 840

[2]. J.W Choi, R.E. Curry, G.H Elkaim, "Continuous Curvature Path Generation Based on Bezier Curves for Autonomous Vehicles", *IAENG International Journal of Applied Mathematics,*

[3]. X. Jin, C. kestell, S. grainger, "Agile Motion of a Differential Thrust Autonomous Underwater Vehicle", IEEEnternational Conference on MX. Technology, Melbourne, 2011.

[4]. M.E Rentschler, F.S Hover, C. Chryssostomidis, "System Identification of Open-Loop Maneuvers Leads to Improved AUV Flight Performance", *IEEE Journal of Oceanic Engineering*, ISSN 0364-9059, 2006, Volume 31, Issue 1, pp. 200 – 208

[5]. S. Sharma, "Trajectory Generation and Path Planning for Autonomous Aerobots", IEEE International Conference on Robotics and Automation, Roma, Italy, April 10-14, 2007