Proceedings of the ASME 2024 43nd International Conference on Ocean, Offshore and Arctic Engineering
OMAE2024
June 9-14, 2023, Singapore EXPO, Singapore

OMAE2024-124617

PERFORMANCE EVALUATION OF DIFFERENT CONTROL ALGORITHMS FOR TORPEDO-SHAPED AUVS

Shankruth Balasubramaniyam[1,*], Shriram Umashankar[1], Gowsik Gurudev [1], Madhavnambi Ramanujam [1], Abhilash Somayajula[1], Sriram Venkatachalam[1]

[1]MAV Lab, Ocean Engineering dept, Indian Institute of Technology Madras, Chennai, Tamil Nadu, India

ABSTRACT

This paper compares Linear Quadratic Regulator(LQR) and Proportional Integrative Derivative(PID) control algorithms implemented in an in-house built Autonomous Underwater Vehicle(AUV). The AUV has actuators to control five degrees of freedom and features a non-linear dynamics model established through a Python-based simulation environment. Path following is achieved by employing the Line-of-Sight (LOS) and fully actuated path-following guidance law. The evaluation of the control algorithms is based on the trade-off between effort, endurance, time taken to reach the intended path, overshoot and steady-state error.

Keywords: AUV, LQR, PID, Line of Sight, Guidance, Control

1. INTRODUCTION

The human fascination with oceans is ever-increasing and has led to the use of AUVs and ROVs for their exploration. ROVs are controlled using a tether, while AUVs function without external human intervention. As a result, AUVs have limited power on board, and this makes efficient completion of its tasks necessary. The inability to use GPS underwater to pinpoint the robot's location makes it harder to follow its path precisely. Precise path following and trajectory tracking are challenging without a robust control architecture due to water disturbances, ocean currents, and sensor noise. Therefore, effectively managing the control of these AUVs is a critical issue to address.

To achieve this, various control architectures and guidance laws have been developed. This paper makes use of two control laws, namely PID and LQR. Path-following is carried out for each controller using line-of-sight and fully actuated path-following guidance laws. The second guidance is referred to as fully actuated guidance in the upcoming sections. The guidance laws and controllers are further explained in section 3.

*Corresponding author

The first step towards setting up the control and guidance problem is understanding the dynamics of the AUV. A fully actuated AUV has six degrees of freedom, non-linear dynamics and coupled motion. A lot of previously conducted research has simplified the dynamics and control of the AUV by neglecting coupling[1], reducing the degrees of freedom from six or solving the control problem in two dimensions [2–4] or linearizing the dynamic model [2]. The most common platform for simulation has been Matlab/simulink[1–3]. This paper, however, includes motion coupling of all six degrees of freedom and non-linearities. The paper aims to evaluate and compare two controllers and guidance laws through simulations conducted in a custom Python environment.

2. NON-LINEAR DYNAMIC MODELLING :-

2.1 AUV design :-

The AUV under consideration is an in-house developed torpedo-shaped AUV shown below in Figure 1.

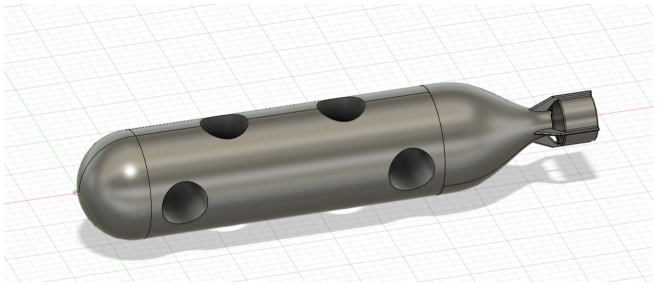

FIGURE 1: AUV

It is equipped with a contra-rotating propeller and four other thrusters for actuation. The contra-rotating propeller is kept at the aft to generate surge thrust. A contra-rotating propeller is used, as it provides the added advantage of producing net zero roll at the aft. The four thrusters are placed in tunnels along the body of the AUV and help maintain torpedo form, which helps

reduce drag. Two thrusters are placed along the sway axis and two along the heave axis to provide four degrees of freedom, i.e. sway, heave, yaw and pitch.

The sway thrusters are kept farther away and heave thrusters near the centre of the AUV, as shown in Figure 1. The AUV is made bottom-heavy, so any small roll encountered during operation will be countered naturally. The AUV is assumed to be neutrally buoyant and has an even keel.

TABLE 1: TABLE OF PARTICULARS

Parameter	Value
Mass	43 Kg
Length	1.12 m
Diameter	0.252 m
Buoyancy	421.83 N
Sway Thruster Gap	0.495 m
Heave Thruster Gap	0.235 m

The origin in the body frame of reference is the centre of the AUV, i.e., the Centre of Buoyancy when fully submerged. The x-axis is towards the front of the AUV, the y-axis is to the right and the z-axis is pointing vertically down. With respect to this coordinate frame, the Centre of Gravity was found to be [0,0,0.081] (metre).

2.2 Dynamics and Kinematics:-

Solving the dynamics and kinematics [5] equations is crucial to controlling an AUV. The six dynamic equations to be solved are given below in (1).

$$[M + A]\ddot{\mathbf{x}} + [C(v)]\mathbf{v} + [D(v)]\mathbf{v} + \mathbf{G} = \mathbf{F}_{ext} \tag{1}$$

F_{ext} in (1) includes the external forces due to actuators, current disturbances and noise. In this paper noise and disturbances are neglected.

M corresponds to the mass matrix in the above equation, and A represents the added mass. The added mass matrix was calculated using HYDRA(Hydrodynamic Response Analysis) software previously known as SIMDYN [6]. The combined mass matrix is given below.

$$[M+A] = \begin{bmatrix} 47.98 & -0.14 & -0.11 & -1.1e^{-3} & -3.85 & -0.48 \\ 0.02 & 87.57 & -0.09 & 3.92 & 0.05 & 0.92 \\ 0.02 & -0.11 & 87.64 & 0.40 & -0.94 & -0.07 \\ 1e^{-4} & 3.92 & 0.40 & 6.35 & 1e^{-4} & 11.05 \\ -3.93 & 0.06 & -0.94 & -2e^{-4} & 146.12 & 0.04 \\ -0.39 & 0.92 & -0.05 & 11.05 & 0.03 & 128.27 \end{bmatrix} \tag{2}$$

C represents the coriolis force matrix. Non-diagonal terms in the combined mass matrix and the C matrix cause motion coupling. Moreover, the equation incorporates non-linearities due to drag and Coriolis forces, accurately modelling real-world situations.

$$d = \frac{1}{2} \cdot \rho \cdot Cd \cdot A \cdot |\mathbf{v}| \tag{3}$$

[D] is a diagonal matrix whose values are calculated using (3).'A' represents the projected area, ρ is the density of water and Cd is the drag coefficient. The necessary parameters to calculate the drag forces are provided in Table 2. The term **G** represents the restoring force vector.

TABLE 2: CD VALUES

Axis	Cd Values	Projected Area(m^2)
Surge	0.19	0.06
Sway	0.59	0.26
Heave	0.59	0.26
Roll	0.31	0.06
Pitch	0.35	0.26
Yaw	0.35	0.26

The kinematics equations convert velocities from the body frame to the NED(North East Down) reference frame. The six equations are represented in matrix form in (4), where **v** represents translational velocities, and ω represents rotational velocities.

$$\begin{bmatrix} V_{ned} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} [R] & 0 \\ 0 & [T] \end{bmatrix} \begin{bmatrix} V_{body} \\ \omega_{body} \end{bmatrix} \tag{4}$$

The R matrix converts the body frame linear velocities to the NED reference frame. Here, letters c,s and t represent cosine, sine and tangent functions, respectively, while the symbols θ,ϕ, and ψ represent pitch, roll and yaw angles. The order of rotation used to obtain the matrix R is given as z-y-x rotation.

$$R = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi c_\phi s_\theta \\ s_\psi c_\theta & c_\psi c_\phi + s_\phi s_\theta s_\psi & -c_\psi s_\phi + s_\theta s_\psi c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{5}$$

The T matrix converts body frame rotational velocities to NED reference frame. The above matrix also uses the same notation as matrix R.

$$T = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \tag{6}$$

(1) and (4) combined give twelve differential equations, and solving them gives velocities and displacements in the world frame.

3. CONTROL THEORY AND GUIDANCE LAWS :-

The two main control algorithms used in simulations were PID and LQR. PID is the abbreviated form of Proportional, Integral and Derivative control. The control output given by (7) is the summation of the three terms mentioned above. The proportional control gives an output proportional to the error, the derivative term gives the rate of change of error, and the integral term gives the accumulation of the errors over time.

$$u = Kp \cdot e + Kd \cdot \dot{e} + Ki \int e(t)dt \tag{7}$$

In (7), u is the control output and e is the error from desired state.

LQR or Linear Quadratic Regulator is a control algorithm that uses full state feedback law. It uses matrices A and B corresponding to the state space equation (8). It represents (1) and (4).

$$\dot{\mathbf{x}} = [A]\mathbf{x} + [B]\mathbf{u} \tag{8}$$

Matrices Q and R are a part of the cost function.The cost function is given in (9)

$$J = \int \left(\mathbf{x}_k^T[Q]\mathbf{x}_k + \mathbf{u}_k^T[R]\mathbf{u}_k\right) \tag{9}$$

The control law used in lqr is given by (10).

$$\mathbf{u} = [K] \cdot (\mathbf{r} - \mathbf{x}) \tag{10}$$

In (10), x is the current state vector, and r is the desired state vector. The gain matrix is calculated using (11).

$$[K] = [R]^{-1}[B]^T[P] \tag{11}$$

Where P is calculated as the solution of the Ricatti equation given by (12),

$$[A]^T[P] + [P][A] - [P][B][R]^{-1}[B]^T[P] + [Q] = 0 \tag{12}$$

3.1 Simulation Environment

Figure 2 shows the overall architecture of the custom simulation environment created in Python. Python was used as it can be interfaced with ROS to test the results of the simulations on the actual AUV. The first step in solving the control problem is calculating the error. This is done by taking the difference between desired and current states of the AUV. The desired state is calculated based on the guidance laws. Then, the error is fed to the controller to generate the desired body frame force vector. This is then used to get force for each thruster by multiplying with the inverse of the thrust allocation matrix [B]. The thruster forces are then capped based on the saturation limit. Then, the body frame forces are again calculated based on (13).

$$\tau = [B] \cdot \mathbf{u} \tag{13}$$

Then, the state vector is obtained from the dynamics block by solving the differential equations using numerical integration, and this process is repeated for the entire duration of the simulation.

3.2 LOS Guidance:-

LOS makes use of only three degrees of freedom of the AUV, namely surge, pitch, and yaw, even though it is capable of sway and heave motions as well. This guidance law controls the heading of the AUV by providing the desired pitch and yaw angles. As the desired heading changes with each instant of time, the sway and yaw thrusters continuously adjust the heading of the AUV until it reaches its intended path. The surge thruster is made to run at a constant rpm throughout the process to propel the AUV forward. Calculations for the desired angles[7–9] were carried out using a look-ahead distance, which was taken approximately equal to twice the length of the AUV. The term look-ahead distance denotes the distance between a specific point along the desired path of the AUV, situated on the perpendicular line drawn from the robot's current location, and the point that the AUV should orient towards. The researcher establishes this distance to guide the AUV in adjusting its heading, minimizing sway error, and aligning itself with the intended path

3.3 Fully Actuated Guidance :-

While line-of-sight guidance uses pitch, yaw, and surge motions, this method optimizes the rise time by utilizing four degrees of freedom, namely surge, sway, heave and yaw. This method, too, uses a fixed thrust at the aft by maintaining the same constant propeller rpm. Heave thrusters assist in controlling the heave motion and maintaining zero pitch, i.e. an even keel, whereas sway thrusters are utilized to attain the required yaw angle and reduce cross-track error. Simultaneous motions increase the power consumed initially, and the coupling of various degrees of freedom makes the motion complex and more challenging to control.

4. RESULTS AND DISCUSSIONS :-

This section provides an in-depth analysis of the guidance laws and controllers mentioned previously. It is divided into three parts. The first part focuses on comparing the controllers within each guidance law, and the second compares them across different guidance laws. These two parts focus on the trade-off between effort, endurance, time taken to reach the intended path, overshoot and steady-state error, the most important being energy consumed. It is calculated by making use of the thrust force vs power consumed graph obtained from thruster characteristics. Even though this graph was generated based on the assumption that the thruster is static, it was assumed to be fairly accurate even though the AUV is in motion. The last part takes a look at the roll caused during the motion of the AUV. This is significant as the roll is the only motion that cannot be actively controlled using the actuators present. The initial position of the AUV is set as (6,4,-2) with the heading angles set to 0^o. The final path it has to attain is given by the three-dimensional line equations y=1.034x-34 and z=30 in the NED(North, East, Down) reference frame.

4.1 LOS

The figure below shows a plot of Z vs time. It indicates that the PID rises marginally faster than LQR. However, it has an overshoot, while LQR settles faster with minimal overshoot.

Figure 4 gives a plot of X vs Y. Both controllers perform relatively similarly. At the same time, PID has a more significant overshoot as compared to LQR, similar to the graph seen in the previous case.

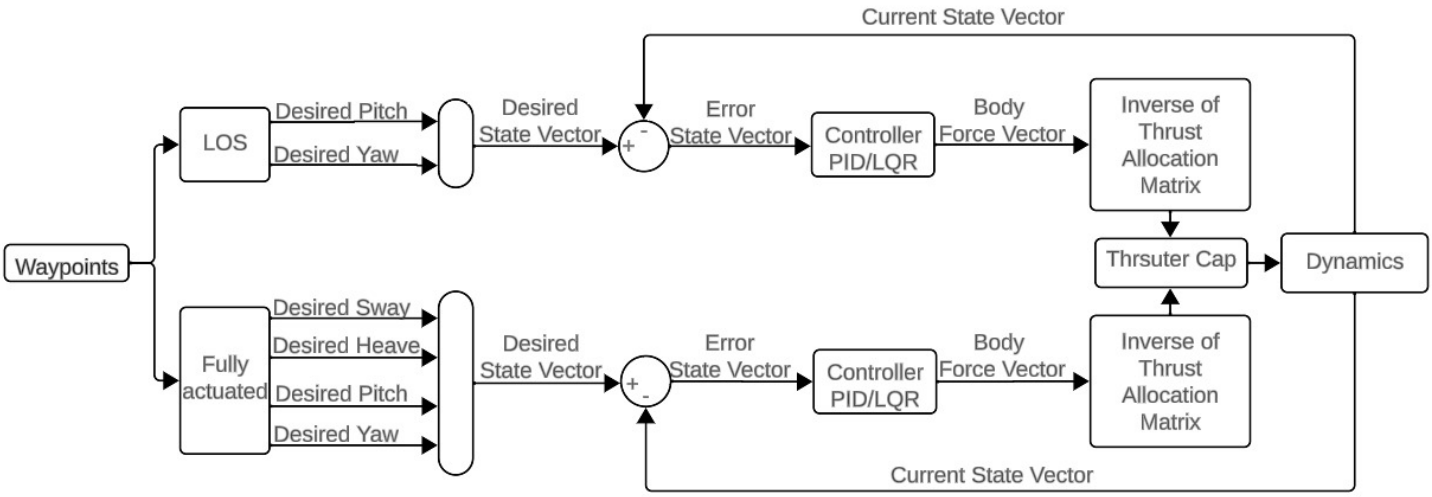

FIGURE 2: SIMULATION ENVIRONMENT

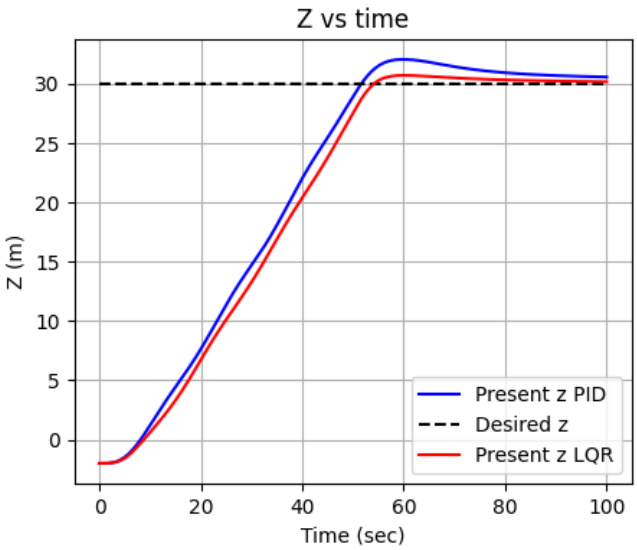

FIGURE 3: DEPTH VS TIME

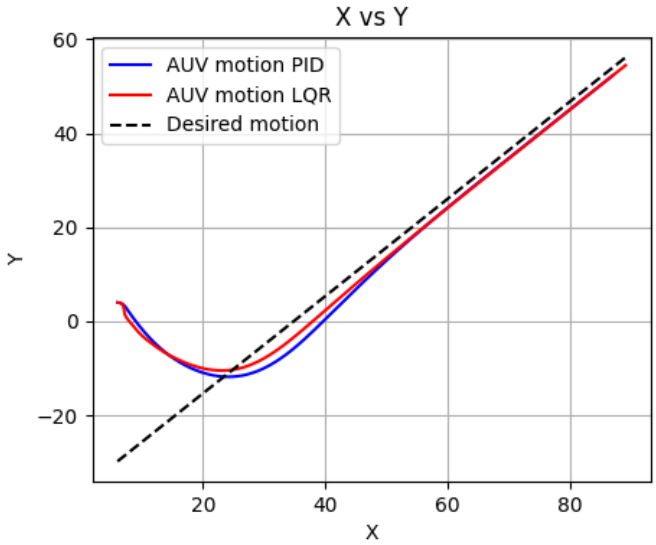

FIGURE 4: X VS Y

As seen in Figure 3 and Figure 4, there is a minimal steady-state error in the heave and sway directions. In the case of PID, there is no direct control over sway and heave motion as control is exerted on the heading of the AUV, i.e. pitch and yaw angles are controlled. In LQR, however, the absence of an integral term causes the steady-state error.

Figure 5 and Figure 6 have similar trends to Z vs time and X vs Y graphs. PID has a more significant overshoot, while LQR settles faster. The faster rising time and overshoot in the case of PID can be attributed to a high proportional gain in the case of PID as compared to LQR.It can be noticed that the pitch and yaw angles attain their intended values as there is direct control exerted over them in the case of PID as well as LQR controllers. As seen in the Figure 6, the desired pitch sometimes reaches 80^o, but the AUv isn't able to pitch above a certain pitch angle.

This happens because of the restoring force. Initially, the

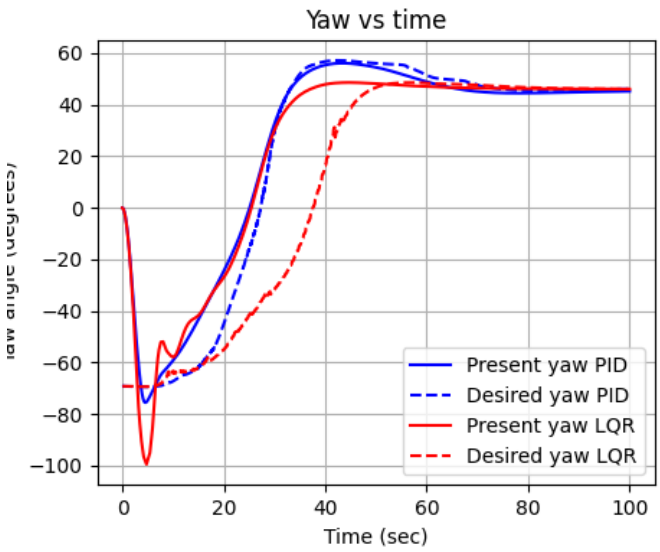

FIGURE 5: YAW VS TIME

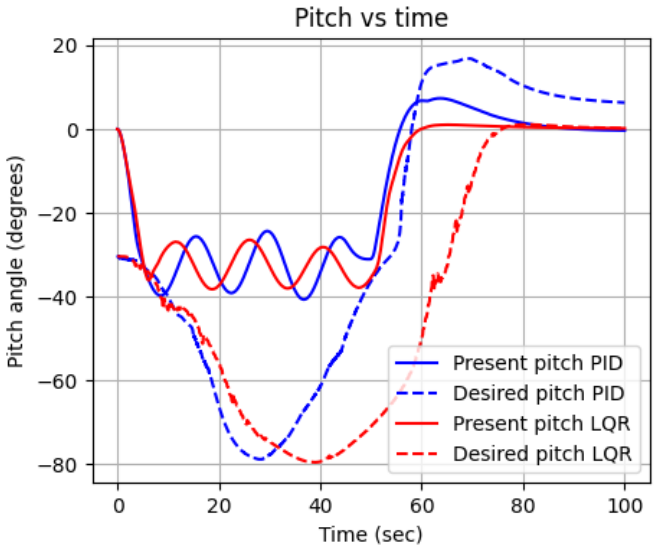

FIGURE 6: PITCH VS TIME

restoring force provides no torque. The only torque provided is by the pair of heave thrusters. When the AUV begins to rotate, the Torque, due to the restoring force, increases and, at a certain angle, equals the torque produced by the heave thrusters, thus preventing it from pitching further. This is reflected in Figure 6, where the pitch angle saturates at around 40^o.

Also, there is an offset in the desired pitch for PID. This is because, for the simulation time provided, PID has a steady state error in Z. This makes the AUV pitch, correcting the steady state error and returning to its intended path.

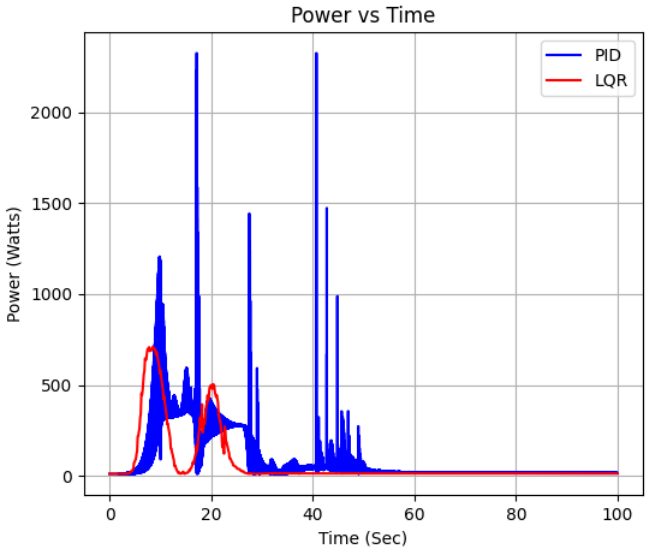

FIGURE 7: POWER LOS

The power-versus-time graph generated from the thruster characteristics indicates that PID consumes more energy over time than LQR. This is evident in the more significant peaks with higher magnitudes in PID's graph, suggesting more abrupt manoeuvres and increased power consumption. These peaks are a result of high kd values. Lowering the kd increases oscillations in the system. Thus, higher kd is adopted to reduce oscillations of the system. On the other hand, the cost function in the case of LQR results in a smoother graph with fewer peaks and reduced energy consumption.

4.2 Fully Actuated Guidance :-

The path following guidance law indicates zero steady-state error as seen in Figure 8 and Figure 9. Moreover, the system is critically damped, unlike the plot seen for LOS guidance. This is because there is direct control over sway, heave, pitch and yaw motions. The heave thrusters maintain the desired depth while the sway thrusters help sway and yaw.

The Figure 10 indicates that LQR takes longer to converge than PID and has a slightly more significant overshoot for these particular gain values.

It can be seen from Figure 11 that LQR has a more significant pitch motion as compared to PID. In the case of PID, minimal pitch and direct control are exerted on the pitch. In the case of

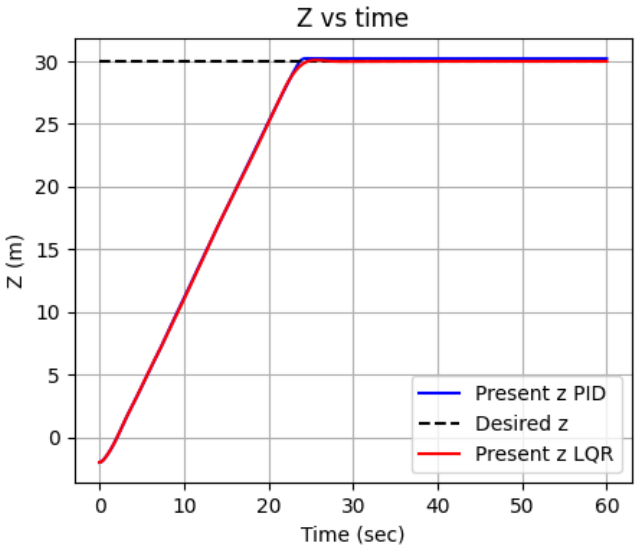

FIGURE 8: DEPTH VS TIME

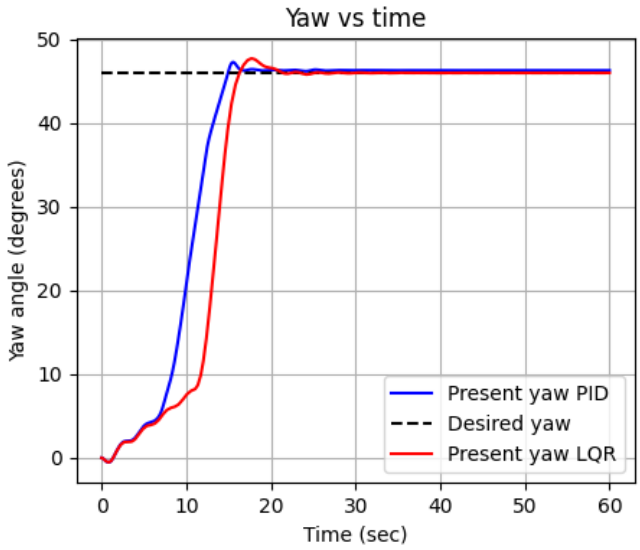

FIGURE 10: YAW VS TIME

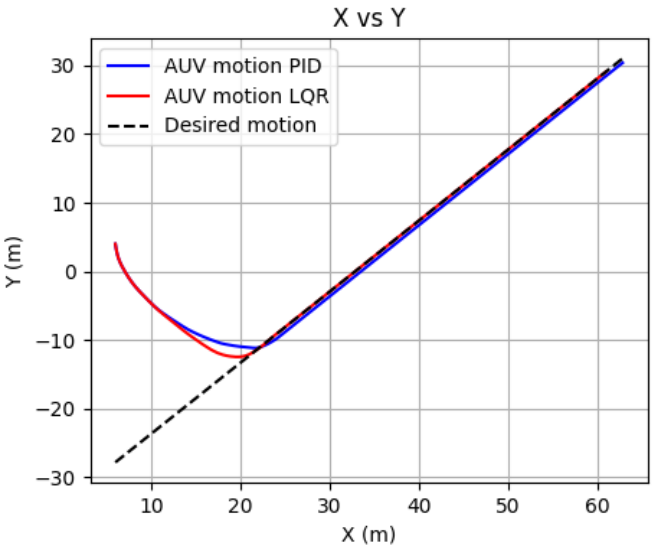

FIGURE 9: X VS Y

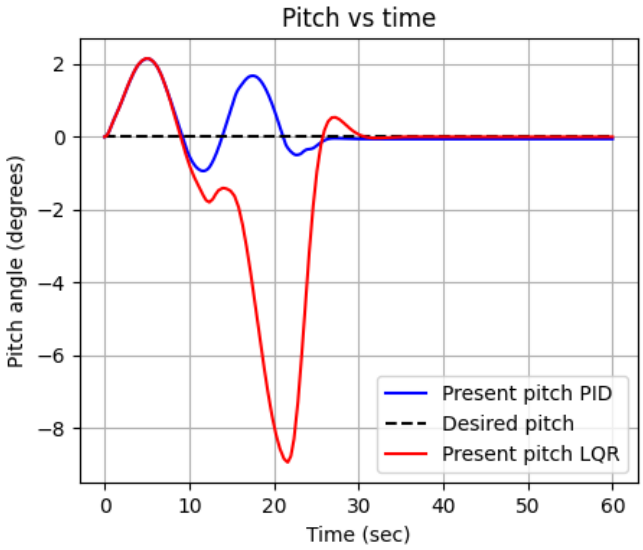

FIGURE 11: PITCH VS TIME

LQR, it is interesting that the AUV pitches down while descending to a greater depth. This is because it optimizes the descending time as surge motion helps in the descent. Effort is also reduced as surge drag is lesser due to the torpedo shape of the AUV.

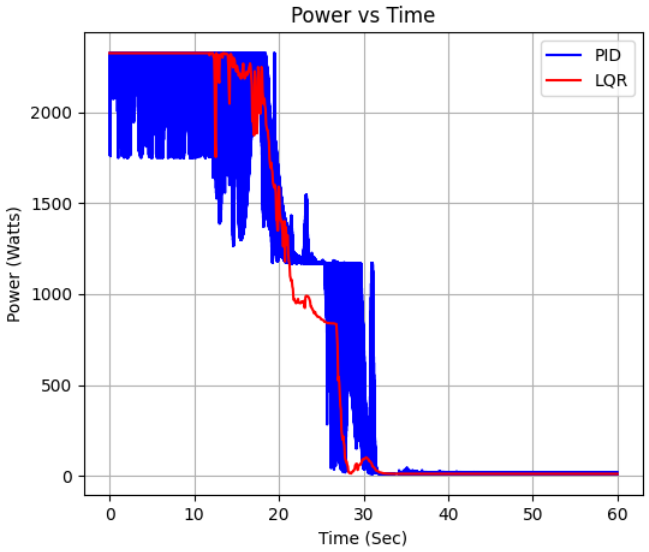

FIGURE 12: POWER

The graph in Figure 12 shows that LQR consumes less energy compared to the PID. It can also be noted that initially have maximum power consumption. This is because all the thrusters are active and running at full capacity to reduce the errors in sway, yaw,heave and surge directions.

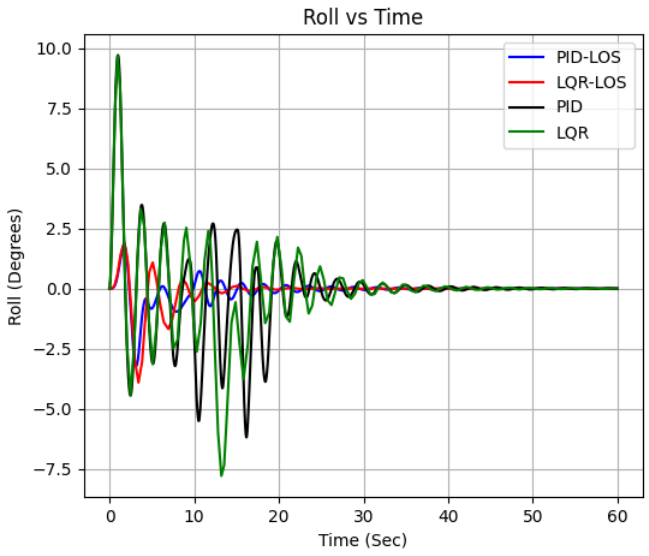

FIGURE 13: ROLL VS TIME

4.3 LOS vs Fully Actuated Guidance :-

The two guidance laws provide two ends of the spectrum where LOS takes more time to converge while using less power. In contrast, fully actuated guidance takes up more power and causes more significant roll while converging faster. PID consumes 10848 J and 5345 J, and LQR consumes 6152 J and 4158 J in fully actuated and LOS, respectively. Thus LOS consumes relatively less energy to reach its intended path. Therefore, fully actuated guidance is suitable for short missions requiring the bot to reach the intended path quickly, while LOS is suited for long missions requiring greater endurance.

4.4 Roll Control :-

Roll motion is the only one among the six degrees of freedom that the AUV cannot actively control. Roll motions are caused primarily by coupling and non-linearities present in the dynamics. To counteract the roll produced, the AUV is made bottom-heavy, as previously mentioned. It can be noted from Figure 13 that the system handles roll relatively well, limiting maximum roll to 2.5^o in the case of LOS. At the same time, there is significantly more roll in the case of a fully actuated system. This is because all motions co-occur, and each is coupled with the roll motion; thus, the effect gets compounded to produce a more significant roll. It can also be noted that PID handles roll better than LQR in both cases.

5. CONCLUSION :-

The comparison of PID and LQR in the preceding section highlights the distinct performance variations across different aspects. While both methods have strengths and weaknesses, PID was superior to LQR in controlling roll motion and displayed a marginally faster rising time. Additionally, using fully actuated guidance, PID could maintain a zero pitch angle better than LQR. However, concerning critical aspects such as minor overshoot, smaller settling time, and lower power consumption, LQR outperforms PID by a significant margin. Thus, it is clear that LQR is a better fit for long-endurance AUVs. It is important to note that the choice of guidance law ultimately depends on the specific use case. LOS is better for more extended missions, while fully actuated guidance is better for shorter ones.

REFERENCES

[1] Bayusari, Ike, Alfarino, Albert Mario, Hikmarika, Hera, Husin, Zaenal, Dwijayanti, Suci and Suprapto, Bhakti Yudho. "Position Control System of Autonomous Underwater Vehicle using PID Controller." *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*: pp. 139–143. 2021. DOI 10.23919/EECSI53397.2021.9624231.

[2] Lakhwani, Deepa A. and Adhyaru, Dipak M. "Performance comparison of PD, PI and LQR controller of autonomous under water vehicle." *2013 Nirma University International Conference on Engineering (NUiCONE)*: pp. 1–6. 2013. DOI 10.1109/NUiCONE.2013.6780183.

[3] Joshi, SD and Talange, DB. "Performance Analysis: PID and LQR Controller for REMUS Autonomous Underwater Vehicle (AUV) Model." *International Journal of Electrical*

Engineering & Technology (IJEET) Vol. 3 No. 2 (2012): pp. 320–327.

[4] Abdurahman, Bilal, Savvaris, A and Tsourdos, A. "A comparison between guidance laws for AUVs using relative kinematics." *OCEANS 2017-Aberdeen*: pp. 1–6. 2017. IEEE.

[5] Fossen, Thor I. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons (2011).

[6] Somayajula, A and Falzarano, J. "Large-amplitude time-domain simulation tool for marine and offshore motion prediction: Ergodicity study of coupled parametric roll motion in irregular seas." *Marine Systems and Ocean Technology* Vol. 10 No. 1 (2015): pp. 1–17. DOI 10.1007/s40868-015-0002-7. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84964771395&doi=10.1007%2fs40868-015-0002-7&partnerID=40&md5=0a326687814bb6de84b94c39b4f8e13e.

[7] Breivik, Morten and Fossen, Thor. *Guidance Laws for Autonomous Underwater Vehicles* (2009). DOI 10.5772/6696.

[8] Fossen, Thor. "An Amplitude-Phase Representation of the North-East-Down Kinematic Differential Equations." *IEEE Access* Vol. PP (2023): pp. 1–1. DOI 10.1109/ACCESS.2023.3242331.

[9] Fossen, Thor. "An Adaptive Line-of-sight (ALOS) Guidance Law for Path Following of Aircraft and Marine Craft." *IEEE Transactions on Control Systems Technology* Vol. 36 (2023): pp. 2887–2894. DOI 10.1109/TCST.2023.3259819.