COMPARISON OF PITCH CONTROL SYSTEM FOR AN UNMANNED FREE-SWIMMING SUBMERSIBLE VEHICLE WITH PD CONTROLLER AND LINEAR QUADRATIC REGULATOR USING MATLAB

BY

MOHAMMAD JAWAD MASUD, FAISAL FAYYAZ QURESHI, ABDUL REHMAN CHISHTI, MUHAMMAD ZUBAIR ASLAM, MUHAMMAD AZHAR GHAURI and NASIR NAEEM

The Islamia University of Bahawalpur, Pakistan,
1Dept. of Electronics Engineering, University College of Engineering & Technology
2Dept. of Telecom Engineering, University College of Engineering & Technology
3Univeristy of Bedford Shire, UK.

Received: September 12, 2014
Accepted for publication: November 5, 2014

Abstract. In this paper we have examined the results of the pitch control system, that are used in robotic submersible automobiles by using PD controller and linear quadratic regulator and results are compared. The penetration of the vehicle is well-ordered in a way that during frontward motion an elevator seeming on the vehicle is identified by a nominated aggregate. The identification reasons the automobile to revolve about the pitch axis. The pitch of the axis generates a vertical influence that causes the vehicle to immerse or grow. It is to be guaranteed for pitch control analysis that pitch angle of automobile should follow the pitch command angle. The class of tracing is well-ordered by the pitch control. In the first part of the paper, we have observed performance of closed loop control systems without any controller. Second part is related to the investigation of the performance in closed loop system controlled by means of a proportional integral controller. After that system performance is improved using derivative controller. Then PD controller on the system is implemented. The simulation results are taken

*Corresponding author; e-mail: rehmanchishti@gmail.com
as well. Finally, system performance is also been observed by using linear quadratic regulator and results are compared.

**Key words:** Unmanned free swimming submersible vehicle (UFSS); vehicle model; pitch control system; system response; step response; PD controller; linear quadratic regulator.

*2010 Mathematics Subject Classification*: 93C83.

1. Introduction

**A. Model of Vehicle**

Pitch angle of vehicle $\theta_f(t)$ which is followed by a pitch command angle $\theta_r(t)$ is objective of pitch control system. System’s block diagram is given in Fig. 1. Here by multiplying two transfer functions, we get a representation of vehicle dynamics. Let the two transfer functions be $G_1(s)$ and $G_2(s)$, which then generate the transfer function named as $G(s)$ which is between the angle of elevator deflection and pitch angle $\theta_f(t)$ (output) which is established by the elevator actuator (input) as in (1). (Discrete PI and PID Controller Design and Analysis for Digital Implementation)

$$G(s) = G_1(s) \cdot G_2(s)$$ (1)

From (Introduction: PID Controller Design) we obtained the coefficients for transfer functions, since;

$$G_1(s) = \frac{-0.125}{s^2 + 0.226 + 0.02535}$$ (2)

$$G_2(s) = \frac{s + 0.435}{s + 1.23}$$ (3)

![Fig. 1 – Pitch Control Loop for Unmanned Free Swimming Submersible Vehicle.](image-url)
Here the negative sign of controller is very important. It is compulsory to mention it, which helps to duck the existence of positive feedback in the system (Introduction: PID Controller Design).

**B. Actuator of Elevator**

As the actuator of elevator accepts the directed angle of elevator deflection ($\theta_{e,\text{com}}$) and then places the surface of elevator accordingly, at an angle $\theta_e$ ideally, it should be like $\theta_{e,\text{com}} = \theta_e$, as we all know that the actuator is not an (perfect) ideal device and have a narrow bandwidth. Since, the relation between them is considered by using a filter (should be low pass) which has a transfer function which is given in (4) (Kuo & Golnaraghi, 2003).

$$H(s) = \frac{2}{s+2} \quad (4)$$

**C. The Pitch Control System**

The pitch control contains the two controllers, which are gain (-K) and pitch rate sensor. Basically the sensor is a differentiator in which pitch rate is produced by pitch magnitudes. It is characterized by a transfer function $K_s s$.

In this paper, initially we analyzed the system performance without any controller. Secondly we supposed that pitch rate sensor is not enabled such that $K_s$ is zero, since we used just pitch gain controller. By adjusting the value of K by hit and trial method we observed that performance is better than first case. Then finally to approach towards exception for the system response we used pitch rate sensor and here system properties can be changed by changing the values of K and $K_s$, and in this case we noticed that the system response is acceptable (Beards, 1988).

**2. System Response Without Controller**

To observe the system response without gain controller and rate sensor, block diagram of system shown in Fig. 2 and its transfer function will be representation is shown in (5).
Putting values in from (2) and (3) in (5), we get another equation which is named as (6);

\[
\frac{\theta_y}{\theta_r} = \frac{G_1 G_2}{1 + G_1 G_2}
\]

(5)

\[
\frac{\theta_y}{\theta_r} = \frac{-0.125}{s^2 + 0.226s + 0.1295}
\]

(6)

In terms of its state space, it can be expressed as;

\[
\dot{X} = AX + Bu
\]

\[
Y = CX + Du
\]

Where:

\[
A = \begin{bmatrix}
-0.226 & -0.1285 \\
1 & 0
\end{bmatrix};
B = \begin{bmatrix}
1 \\
0
\end{bmatrix};
C = \begin{bmatrix}
-0.125 & 0
\end{bmatrix};
D = 0
\]

Transfer function given in (5) is used to find the step response of the system, and Fig. 3 shows the step response of the system which is obtained by using the MATLAB command ‘step’. This figure gives us information about the system response. From this figure we can calculate percent overshoot, rise time and settling time as well. Fig. 4 shows the detailed information about percent overshoot, rise time and peak time (Beards, 1988; Bennett, 1986).
Fig. 4 − Analysis of overshoot, rise time, settling time.

From these information, it is clear that system response is worse. So we have to use a pitch gain controller to make the system performance better. (Buckley, 1976) Since by drawing its root locus by using MATLAB command ‘rlocus’, then command ‘rlocfind’ will help us to find the value of k as shown in Fig. 5.

Fig. 5 − Sketch of Root Locus.

From the observation of Fig. 5 we conclude that system performance is not as better as we want. Since to make our system’s performance better we will
use a pitch gain controller (Chow, 1986). Before using controller such as proportional or derivative controller, we should know that how does it works. Although our point of concern will be towards PD controller in this paper. So, for more convenient and to show a difference and effect of integrator controller as well, we have a brief introduction to PID controller.

### 3. PID Controller

Actually controller is a device that generates an output signal with respect to input signal (error signal) when it received from feedback. An error signal is a difference between actual value and desired value. Output signal sent to controller from feedback, it compares with the input signal if there is error then controller reduces the error. But this output signal value is depends upon the design or action of a PID controller. Fig. 6 shows the structure of PID controller (Beards, 1988; D’Azzo & Houpis, 1995).

The transfer function of the PID controller shown in Fig. 6. Mathematically it will look like as given in (7).

$$K_p + \frac{K_i}{s} + K_ds = \frac{K_i s^2 + K_ds + K_c}{s}$$  \hspace{1cm} (7)

- $K_p$ – Proportional gain
- $K_i$ – Integral gain
- $K_d$ – Derivative gain

Fig. 6 – PID Controller Structure.

Proportional controller ($K_p$), Integral controller ($K_i$) & Derivative controller ($K_d$) are the Parameters or Gains of PID Controller. Their values are depends upon each other. The effect of these parameters on each other is listed in Table 1 (Katsuhiko, 1990).
Table 1

<table>
<thead>
<tr>
<th>CI Response</th>
<th>Rise Time</th>
<th>% Overshoot</th>
<th>Settling Time</th>
<th>S-S Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kp</td>
<td>Decrease</td>
<td>Increase</td>
<td>Small Change</td>
<td>Decrease</td>
</tr>
<tr>
<td>Ki</td>
<td>Decrease</td>
<td>Increase</td>
<td>Increase</td>
<td>Eliminate</td>
</tr>
<tr>
<td>Kd</td>
<td>Small Change</td>
<td>Decrease</td>
<td>Decrease</td>
<td>No Change</td>
</tr>
</tbody>
</table>

4. System Response With Pitch Gain (Proportional) Controller

In this case we are using pitch gain controller with which we can change the properties of system. Table 1 shows the effect of controller on the system response. Since the block diagram for this case is shown in Fig. 7.

![Block Diagram of UFSS with Gain controller.](image)

Fig. 7 – Block Diagram of UFSS with Gain controller.

Keeping in mind Table 1, we will change the value of Kp to make our system performance better. In this case transfer function of the system will comes out as in (8);

$$\frac{\theta_y}{\theta_r} = \frac{G_1G_2HK_p}{1 + K_pHG_1G_2}$$  \hspace{1cm} (8)

By substituting values from (2), (3) and (4) in (8), we get another equation which is named as (9);

$$\frac{\theta_y}{\theta_r} = \frac{0.25K_p}{s^3 + 2.226s^2 + 0.5905s + (0.257 + 0.25K_p)}$$  \hspace{1cm} (9)
To observe the system performance with proportional controller we find the step response of the system using the MATLAB.

Fig. 8 shows that proportional controller reduced both the rise time and the steady-state error, increased the over-shoot, and decreased the settling time also. Though this response is better than first case, but it may be worse due to addition of noise. Addition of noise may make the system unstable, since the system response will no longer be stable. So to decrease percentage overshoot and to get a better response, here we will also use a derivative controller (Katsuhiko, 1990; Liang, 2009).

5. System Response with PD Controller

As we have discussed two cases, in first one we discussed the system performance without any controller and in second case we discussed the system performance using a proportional controller with which performance of system became better than first case.

![Fig. 9 – Block Diagram of UFSS using PD controller.](image)
Now to make it best we will use derivative controller along with proportional controller and here it can also be known as pitch rate sensor. Fig. 9 shows the block diagram of UFSS using PD controller (Nise, 2004; Palm, 2005).

By solving the block diagram of Fig. 9, we get transfer function which can be expressed mathematically as in (10);

\[
\frac{\theta_i}{\theta_r} = \frac{(0.25s + 0.1088)K_p}{s^3 + 3.456s^3 + (3.215 + 0.25K_d)s^2 + (0.6378 + 0.1088K_d + 0.25K_p)s + 0.6236 + 0.1088K_p}
\]

By using hit and trial method we get the values for \(K_p\) and \(K_d\). Since

\[K_p = 50; K_d = 45\]

To check the performance of the system by using PD controller, we find the step response of the system by using MATLAB command ‘step’. Fig. 10 shows the step response of the system.

So from above plot it is clear that system response is much better by using PD controller. Since every parameter is in limit like that rise time, percent overshoot, settling time, peak time every parameter has been controlled by using PD controller (Introduction: PID Controller Design).
6. System Response Using Linear Quadratic Regulator

A. Introduction to Linear Quadratic Regulator

The theory of optimal control system is related with operating a dynamic system at lowest cost. Linear Quadratic Regulator has very wide applications in automation and control system and it is one of the precise tool for automation and control system. The case where the system dynamics are described by a set of linear differential equations and the cost is described by a quadratic function is called the LQ problem. One of the main results in the theory is that the solution is provided by the linear-quadratic regulator (LQR), a feedback controller whose state equation is given below in (11) (Yang, 2005).

\[ \dot{x} = Ax + Bu \]  

Where:

\[ A = \begin{bmatrix} -0.226 & -0.1285 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]

For a continuous time system, the state-feedback law minimizes the quadratic cost function, and the equation for state feedback law is shown in (12) (Introduction: PID Controller Design; Yang, 2005)

\[ u = -Kx \]  

Where K is the most important parameter here, which is the state feedback gain, whose matrix is determined by using MATLAB which is given below;

\[ K = \begin{bmatrix} 1.0e+004 * \\ 0.0000 & 8.6989 & 0.4340 & 0.0109 \end{bmatrix} \]

Mathematically K can be derived by using equation shown in (13).

\[ K = R^{-1}(B^T S + N^T) \]  

Whereas quadratic cost function is shown in (14).

\[ J(u) = \int_{0}^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt \]  

B. Algorithm for Implementation of Linear Quadratic Regulator in MATLAB

Initially diagonal matrix is introduced by using MATLAB command ‘\texttt{diag}’. Then MATLAB command ‘\texttt{lqr}’ is used to find K, P and E, where
K, P and E are parameters to solve raccati equation. Then to get Eigen values MATLAB command ‘eig’ is used. Finally, the simulation results are obtained and then compared it with the results of PD controller.

C. Simulation Results using Linear Quadratic Regulator

Fig. 11 − System Response using LQR (Linear Quadratic Regulator).

7. Results

Figs. 10 and 11 shows the step response of system using PD controller and LQR (Linear Quadratic Regulator) respectively. Comparing the Figs. 10 and 11 it is clear that system response is stable after approximately 0.1 sec in Fig. 11, while it became stable at about more than 1 sec in Fig. 10.

8. Conclusions

In this paper we selected a closed loop (unmanned free-swimming submersible vehicle) system whose pitch is to be controlled by using two different tools. Initially, we used PD controller to control its pitch response. Therefore, adjusting the values of \( K_p \) and \( K_d \) by hit and trial method keeping in mind the Table 1, we approached to a fine step response of system. After this, we observed the simulation results by using another tool Linear Quadratic regulator which was an excellent result and we observed that response is being stable in 10th part of the second. So with this we are able to control the pitch response in just 0.1 sec. Similarly heading control system of unmanned free swimming submersible vehicle (UFSS) can also be analysed by using Linear Quadratic Regulator more precisely. As this paper leads to the pitch control of a robotic vehicle, since we can say that such type of unmanned (robotic) vehicles can easily be controlled by using linear quadratic regulator (LQR) with greater precision.

Acknowledgements. We are thank full to our teachers specially Engr. Usman Ali Khan and Dr. Faisal Qureshi & friends for helping us out in completion of this
paper. Without help of teachers & friends we would not have completed this paper in a better way. We would also like to thanks our Head of Department Dr. Hafiz Ashiq Hussain, our principal Prof. Jan Muhammad Kereio & Vice chancellor of Islamia University of Bahawalpur Dr. Mukhtar who gave us opportunity to write this paper on international platform in this University.

REFERENCES


COMPARĂRIE ÎNTRE REGULATOARE DE TIP PD ŞI LQR PENTRU SISTEMUL DE CONTROL AL TANGAJULUI DINTR-UN VEHICUL SUBMERSIBIL AUTONOM UTILIZÂND MEDIUL MATLAB

(Rezumat)

În această lucrare sunt investigate și comparate rezultatele obținute în urma implementării unor regulatoroare de tip PD și LQR pentru sistemul de control al tangajului dintr-un vehicul submersibil autonom. Mai întâi a fost dezvoltat un model al vehiculului submersibil și pe baza acestuia au fost proiectate mai multe regulatoroare în vederea obținerii celor mai bune rezultate din punct de vedere al performanțelor. În urma simulărilor efectuate în bucla închisă se poate trage concluzia că regulatorul LQR poate obține cea mai mare precizie și cel mai bun timp de răspuns în controlul tangajului unui vehicul submersibil autonom.