Design and Development of closed loop PID controller in MATLAB SIMULINK to control the pitch angle of tail plane.

Technical Report · July 2021

DOI: 10.13140/RG.2.2.28494.59203

Khizer Ahmed
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

Abdul Kareem
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

Atif Ahmed
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

Muhammad Daniyal
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

System Controls and Dynamics CEP (ME-464)

Submitted By:

Khizer Ahmed	2018568
Abdul Kareem	2018904
Ahmad Abdur Rehman	2017036
Muhammad Atif Ahmed	2017247
Muhammad Daniyal	2017255
Muhammad Fahad Saeed	2017256
Muhammad Haroon Arshad	2017263
Muhammad Hassan Saleem	2017273
Muhammad Ibrahim Khan	2017276

Submitted To:

Dr. S. Mushtaq Ahmad

Contents

Abstract

The study conducted in this manuscript focuses on the modeling of the yaw characteristics of a 1-DOF tale plane aircraft system. The system modeled in this study is done in efforts to optimize PID controller parameters to establish Yaw stability of the Tale plane system. The system's transfer functions were derived from the characteristic equations of the 1-DOF tailplane motions and was modeled in Simulink. Using Simulink augmented space, the system response to rudder deflection-step inputs was plotted and analyzed for various controller gain values such that the system stability characteristics could be observed. The PID controller was designed against several system requirements including open and closed loop stability, disturbance response, response time and overshoot; an iterative process allowed a system to be developed with a PID controller that, in real-time, performed on test-rig and the tunes PID controller satisfied all designed criteria.

Problem Statement

A two degree-of-freedom tailplane exhibits vertical and lateral dynamics similar to that of a typical aircraft as shown in Figure 1 below. The tail-plane is mounted in front of a wind tunnel that allows variation in air speed. The rig is also instrumented, and computer interfaced that allows real-time acquisition of pitch and yaw data as well as enables the user to actuate the elevator and rudder via servomotors. The case study objectives are:

Objectives

1. To develop detailed mathematical model for pitch position in time domain from the provided experimental data.
2. To develop the transfer function between pitch angle and elevator deflection (input), while ensuring system gain is modeled too.
3. To design closed loop controllers in MATLAB/Simulink simulation environment that satisfies the following performance criteria:
    a. Open loop and Closed-loop stability analysis.
    b. Commanded reference signal tracking.
    c. The controller must be able to reject the disturbance
    d. Reasonably fast response as compared to open loop response without actuator saturation or system becoming unstable.
    e. Percentage overshoot of less than 30% and improved settling time as compared to open loop response.
4. Finally, evaluate the real-time performance on test rig based on the tuned PID gains satisfying desired performance criteria.

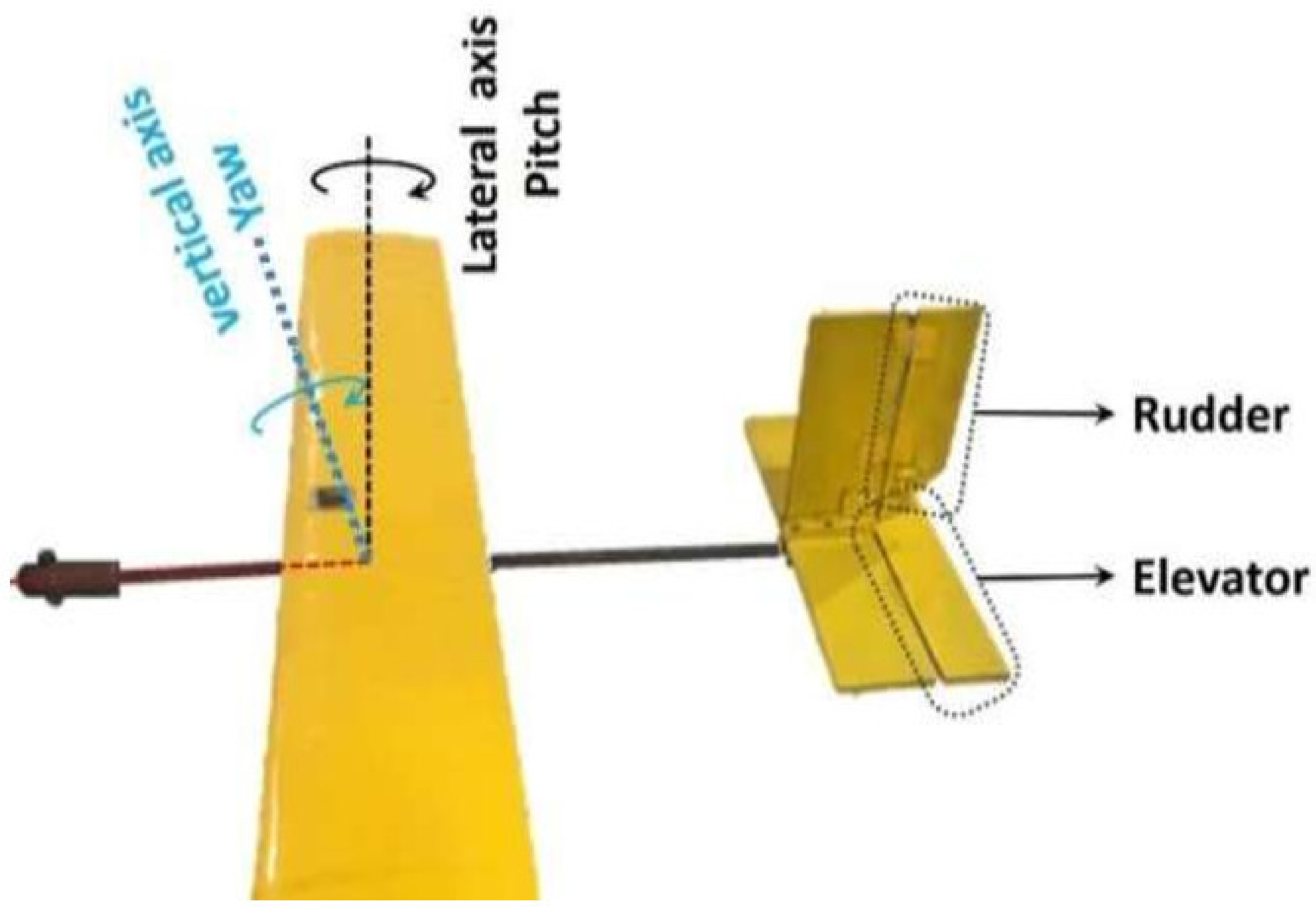


Figure 1: 2 Degree of Freedom Tailplane

Literature review

Airplane flight control frameworks comprise of essential and auxiliary frameworks. The basic control structure, which includes the ailerons, elevator, and rudder, is required to safely control an airplane during flight. Airplane control frameworks are designed to provide enough response to control inputs while maintaining a natural feel. The controls, in general, feel delicate at low speeds, and the aeroplane reacts slowly to control inputs. At higher speeds, the controls become increasingly firm, and the airplane's reaction time becomes faster. Autopilot is a computer-controlled fly system that keeps an airplane in level flight or on a predetermined path. It's usually coordinated by the pilot, although it could also be linked to a radio route signal.

Many papers were evaluated for the Tailplane literature review's control. Almost every report recommended using a PID controller to fly the plane. The software utilized, MATLAB, was the second similarity discovered. Different microcontrollers were used to carry out the practical execution. By directing the control inputs, the PID controller can reduce the estimation of a mistake. PID controllers can provide control activity tailored to specific flight requirements. PID tuning can be done in a variety of ways, according to the literature.

According to another paper, the airplane dynamic model was created using the Airlib module in MATLAB. A Simulink structure was created using this airplane model. Angle was given as an input to the Simulink model. After determining linearization based on the activity point and the framework's minor utilization, the first step was to structure the PID controller using MATLAB sisotool. The bolstered flight mode was applied to the Simulink model's information, which was the change of the lift point, using a PID control framework. MATLAB sisotool was used to determine the exchange work of the PID controller after linearization based on the activity point and obtained framework's minimal execution. The Simulink model's information, which is the change of the aileron point, was used to create a PID control structure for each flying mode. Traditional processes, such as the **Root-Locus technique**, or more modern methods based on improvement with hereditary computations can be used to obtain the controller's settings. When compared to other ways, using the Root-Locus technique for PID tuning is very straightforward and useful since it displays where the open-circle posts and zeros should be changed so the reaction meets the framework execution details. In any event, the disadvantage of this method was that it required the use of a direct model, which was merely an estimate of the UAV's complicated components. The gains are determined by two parameters: the definite increase and the duration of swaying that occurs when a definite addition is made.

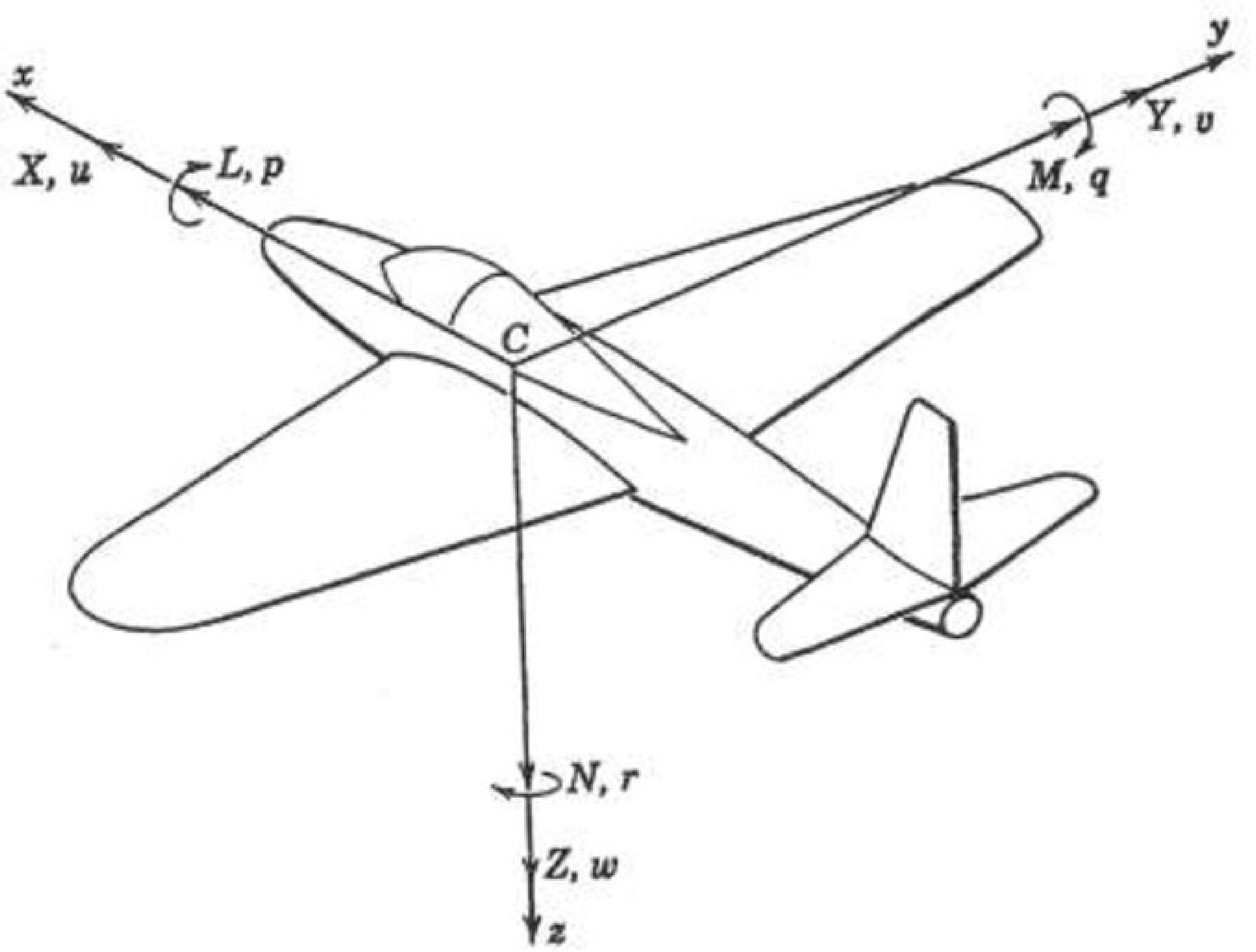
Aircraft Equation:

Nomenclature:

Forces
X Axial Force Net Force in the positive x-direction
Y Side Force Net Force in the positive y-direction
Z Normal Force Net Force in the positive z-direction

Moments
L Rolling Moment Net Moment in the positive p-direction
M Pitching Moment Net Moment in the positive q-direction
N Yawing Moment Net Moment in the positive r-direction

Rotations
p is rotation about the x-axis.
q is rotation about the y-axis.
r is rotation about the z-axis.



Aircraft equation
Fig no:01 Aircraft Body Frame

From the Newton's Laws we know that:

$$F = \sum Fi = m.\frac{dv}{dt} \qquad (i)$$

If $F = [Fx\ Fy\ Fz]$ and $V = [u\ v\ w]$,
then;

$$Fx = m\frac{du}{dt},\ Fy = m\frac{dv}{dt} \text{ and } Fz = m\frac{dz}{dt}.$$

Using Calculus, this concept can be extended to rigid bodies by integration over all particles.

$$\sum Mi = \frac{d}{dt}H \qquad (ii)$$

Where H = $\int (r \times v)dm$ is the angular momentum.
Angular momentum of a rigid body can be found as:

$H = I\ \omega_I$

Where $\omega_I = [p \quad q \quad r]^T$ is the angular rotation vector of the body about the center of mass.

- ω_I is defined in an Inertial Frame.

Coordinate Rotations:

There are 3 basic rotations an aircraft can make:

- Roll (Φ) = Rotation about x-axis
- Pitch (θ) = Rotation about y-axis
- Yaw (Ψ) = Rotation about z-axis

Equation of Forces:

As we know that our model is being examined in inertial frame therefore calculating the forces in the Earth frame on an aircraft can be derived as:
We know that from equation (i):

$$\sum F = m.\left(\frac{dV}{dT}\right)_E$$

Then, according to the principle of relative motion:

$$m\left(\frac{dV}{dt}\right)_E = m.\left(\left(\frac{dv}{dt}\right)_B + \omega_{B/E} \times V_B\right) \qquad (iii)$$

where subscript E is for Earth Frame and Subscript B is for Body Frame
converting the above equation in matrix form we get

$$m\left(\frac{dV}{dt}\right)_E = m.\left(\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} U \\ V \\ W \end{bmatrix}\right)$$

converting the matrix in a skew-symmetric matrix to make the cross product possible;

$$m\left(\frac{dV}{dt}\right)_E = m.\left(\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} + \begin{bmatrix} 0 & -r & -q \\ r & 0 & p \\ q & p & 0 \end{bmatrix} \times \begin{bmatrix} U' \\ V' \\ W' \end{bmatrix}\right) \quad (iv)$$

$$\begin{bmatrix} Fx \\ Fy \\ Fz \end{bmatrix} = m.\left(\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} + \begin{bmatrix} 0 & -r & -q \\ r & 0 & p \\ q & p & 0 \end{bmatrix} \times \begin{bmatrix} U' \\ V' \\ W' \end{bmatrix}\right) \quad (v)$$

Now, after simplification we get:

$$Fx = m.\left(\frac{dU}{dt} + q.W - r.V\right) \quad (vi)$$

$$Fy = m.\left(\frac{dV}{dt} + r.U - p.W\right) \quad (vii)$$

$$Fz = m.\left(\frac{dW}{dt} + p.V - q.U\right) \quad (viii)$$

Hence, Fx, Fy, Fz are the total forces on the aircraft in Earth Frame.

Equation of Moments:

Now, again deriving the moment equation for Earth Frame:

$$M = \left(\frac{dH}{dt}\right)_E$$

Again, from the principle of relative motion:

$$\left(\frac{dH}{dt}\right)_E = \left(\frac{dH}{dt}\right)_B + (\omega_{B/E} \times H_B)$$

Now, solving for $\frac{dH}{dt}_B$;

$H_B = I.\,\omega_B$

Since, I is the Inertia Matrix, therefore;

$$H_B = \begin{bmatrix} Ixx & Ixy & Ixz \\ Iyx & Iyy & Iyz \\ Izx & Izy & Izz \end{bmatrix} . \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

As, we know that in this case, we have $Ixy = Iyx = Iyz = Izy = 0$. Therefore the matrix becomes;

$$H_B = \begin{bmatrix} Ixx & 0 & Ixz \\ 0 & Iyy & 0 \\ Izx & 0 & Izz \end{bmatrix} . \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Differentiate w.r.t time:

$$\frac{dH}{dt} = \left(I.\frac{d\omega}{dt}\right) = \left(I.\begin{bmatrix} p' \\ q' \\ r' \end{bmatrix}\right)$$

$$\frac{dH}{dt} = \left(\begin{bmatrix} Ixx & 0 & Ixz \\ 0 & Iyy & 0 \\ Izx & 0 & Izz \end{bmatrix} . \begin{bmatrix} p' \\ q' \\ r' \end{bmatrix}\right)$$

$$\frac{dH}{dt} = \begin{bmatrix} Ixx.p' + Ixzr' \\ Iyy.q' \\ Izz.r' + Izx.p' \end{bmatrix}$$

Now, substituting the values in $\left(\frac{dH}{dt}\right)_E$.

$$\frac{dH}{dt} = \begin{bmatrix} Ixx.p' + Ixzr' \\ Iyy.q' \\ Izz.r' + Izx.p' \end{bmatrix} + \begin{bmatrix} 0 & -r & -q \\ r & 0 & p \\ q & p & 0 \end{bmatrix} . \begin{bmatrix} Ixx & 0 & Ixz \\ 0 & Iyy & 0 \\ Izx & 0 & Izz \end{bmatrix} . \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} Ixx.p' + Ixzr' \\ Iyy.q' \\ Izz.r' + Izx.p' \end{bmatrix} + \begin{bmatrix} 0 & -r & -q \\ r & 0 & p \\ q & p & 0 \end{bmatrix} . \begin{bmatrix} Ixx & 0 & Ixz \\ 0 & Iyy & 0 \\ Izx & 0 & Izz \end{bmatrix} . \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

After simplification, we get:

$$L = Ixx.p' + Ixzr' + q'(r'(Izz - Iyy) + p.Ixz) \qquad (ix)$$

$$M = Iyy.q' + p.r(Ixx - Ixz) + (r^2 - p^2).Ixz \qquad (x)$$

$$N = Izz.r' + Ixz.p' + p.q(Iyy - Ixx) - q.r.Ixz \qquad (xi)$$

So, these L, M, N are the moments of the aircraft in the Earth Frame.

Equation of Rotation:

As we know that the aircraft rotates about three axis of rotations, generating the angle Φ, θ, Ψ respectively, which is also known as roll, pitch and yaw. So for calculating the rotation of aircraft in Earth Frame. We know that:

$$R(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & -\sin(\Phi) \\ 0 & \sin(\Phi) & \cos(\Phi) \end{bmatrix}, \ R(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \text{ and } R(\Psi) = \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where $R(\Phi), R(\theta)$ and $R(\Psi)$ are Roll, Pitch and Yaw angle.

Now, Relating the angular velocity to changes in angle. So we have

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \frac{d\Psi}{dt} \\ 0 \\ 0 \end{bmatrix} + R(\Psi). \begin{bmatrix} 0 \\ \frac{d\theta}{dt} \\ 0 \end{bmatrix} + R(\Psi).R(\theta). \begin{bmatrix} 0 \\ 0 \\ \frac{d\Phi}{dt} \end{bmatrix} \qquad \text{(xii)}$$

substituting the values of the matrices:

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \frac{d\Psi}{dt} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 0 \\ \frac{d\theta}{dt} \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} . \begin{bmatrix} 0 \\ 0 \\ \frac{d\Phi}{dt} \end{bmatrix}$$

After simplifying, we get:

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\Phi) & \cos(\theta).\sin(\Phi) \\ 0 & -\sin(\theta) & \cos(\theta).\cos(\Phi) \end{bmatrix} . \begin{bmatrix} \Phi' \\ \theta' \\ \Psi' \end{bmatrix} \qquad (xiii)$$

Now, solving for p, q and r, we get the relation of angular velocity with rotation angle:

$$p = \Phi' - \Psi' \sin(\theta) \qquad (xiv)$$

$$q = \theta'^{\cos(\Phi)} + \Psi'\cos(\theta)\sin(\Phi) \qquad (xv)$$

$$r = \Psi'\cos(\theta)\cos(\Phi) - \theta'\sin(\Phi) \qquad (xvi)$$

So as we know that in the given model we are only dealing with the Yaw angle therefore Rolling angle and Pitch angle are zero i-e;

$$\Phi = \theta = 0$$

So, the above equations become:

$p = 0$ (xvii)

$q = 0$ (xviii)

$r = \Psi'$ (xix)

Equation (xix) can be written as:

$$r = \frac{d\Psi}{dt} \qquad (xx)$$

Integrate equation (xx) w.r.t time:

$\int r \ dt = \Psi$

$$\boxed{\Psi = \int r \ dt}$$

Where r is the Yaw rate of the aircraft.

Mathematical modeling

From the figure provided, following values were obtained:

- Peak Angle: -13.14°
- Steady State Angle: -11.52°
- Peak time: 2.05s

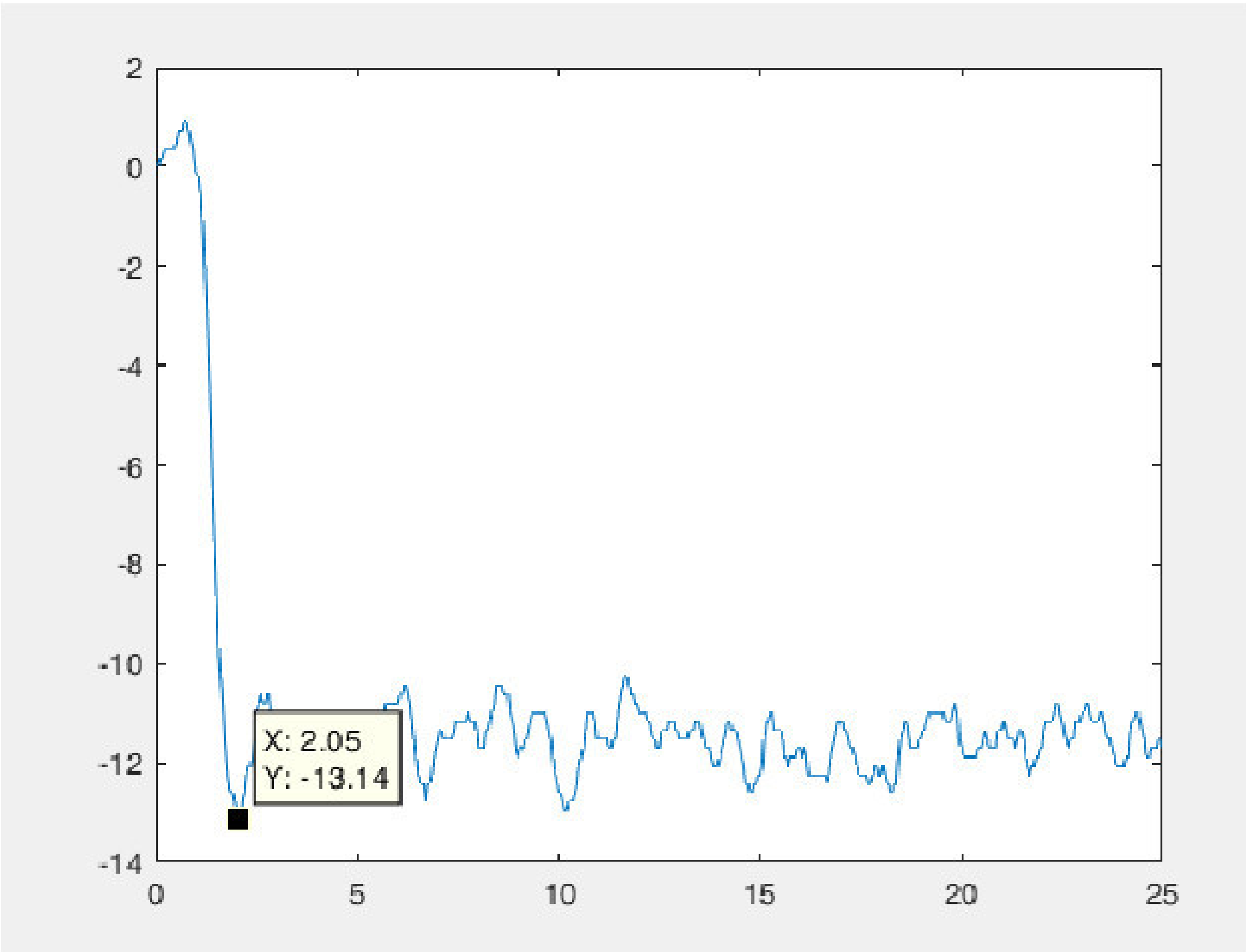


Figure 2: Graph showing Peak Angle = -13.4

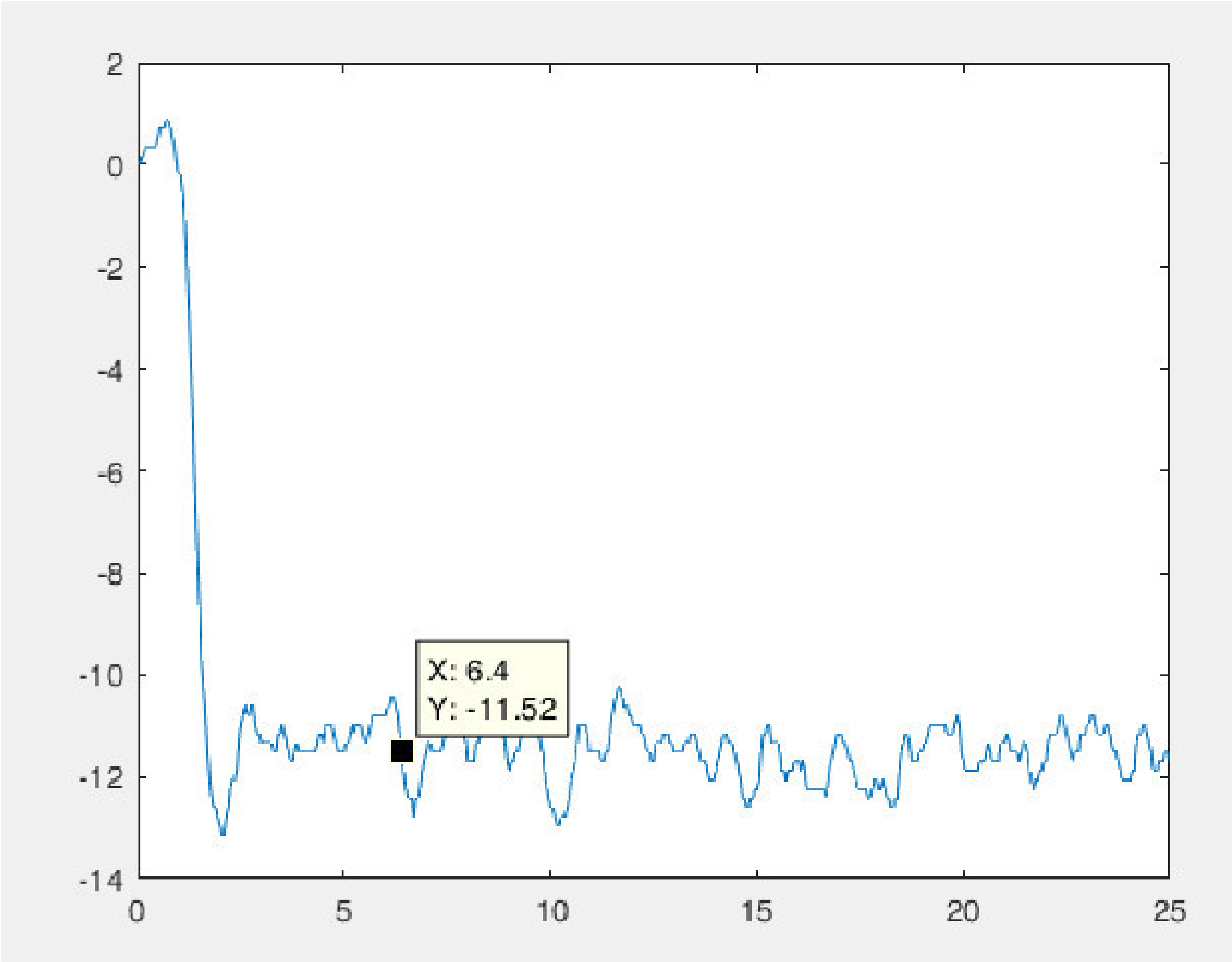


Figure 3: Graph showing Steady State Value = -11.52

$$Maximum\ Overshoot = \frac{Peak - Steady\ state}{Steady\ state}$$

$$= \frac{-13.14 - (-11.52)}{-11.52}$$

$$= 0.140625$$

$$Maximum\ Overshoot = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$

$$0.140625 = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$

$$\ln(0.140625) = \ln(e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}})$$

$$-1.961659 = -\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}$$

$$\zeta = \sqrt{\frac{(-1.961659)^2}{(-1.961659)^2 + \pi^2}}$$

$$\zeta = 0.5296$$

Calculate the natural frequency value , peak time T_P= 2.05s

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$$

$$\omega_n = \frac{\pi}{(2.05)\sqrt{1-(0.5296)^2}}$$

$$\omega_n = 1.8066 \ rad/s$$

The Second order differential is given by:

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = k\,u(t)$$

Where θ = Pitch Angle

Substituting values

$$\ddot{\theta} + 1.9136\dot{\theta} + 3.2638\theta = k\,u(t)$$

Applying Laplace transform,

$$Y(s)(s^2 + 1.9136s + 3.2638) = k\,u(t)$$

Input= u(t) = -10°

Apply Final value Theorem:

$$(3.2638)(-11.52) = k\,(-10)$$

$$k = 3.76$$

Transfer function:

$$\frac{Y(s)}{U(t)} = \frac{3.76}{s^2 + 1.9136s + 3.2638}$$

Transfer Function

Code

```
sys = tf([-37.6],[1 1.9136 3.2638]);
[y,t] = step(sys,8);
step(sys)
```

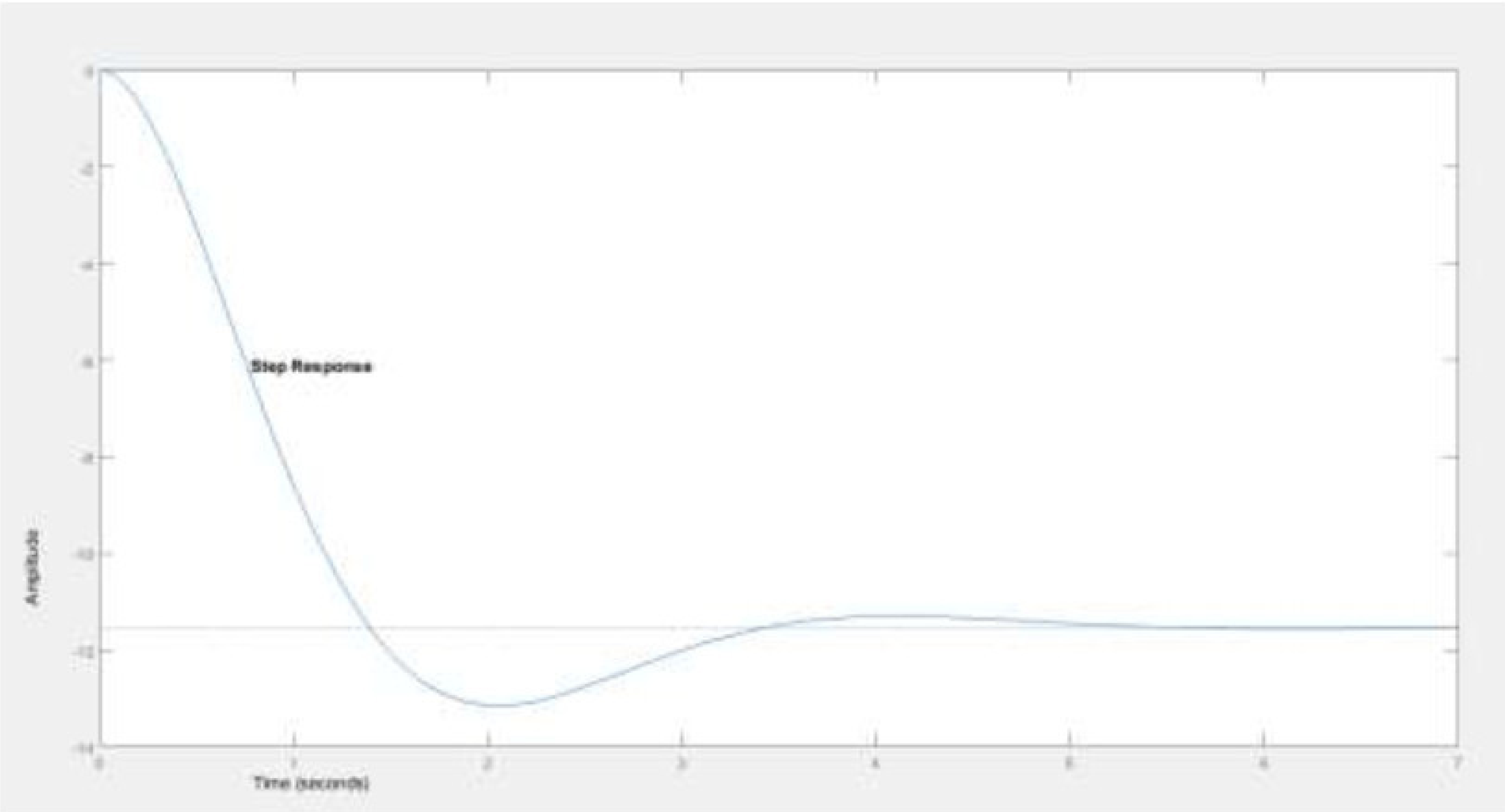


Figure 4: Plot of Transfer function

After modelling the plant using the experimental data, the following step response was plotted on the same graph as the experimental data to verify our model. As it can be seen, our modelled plant is very similar to the experimental data. As obtained from the graph, the open loop transfer function has a settling time of 3.5 seconds.

Root Locus:

In control theory, root locus analysis is a graphical method for examining how the roots of a system change with variation of a certain system parameter, commonly a gain within a feedback system. This is a technique used as a stability criterion in the field of classical control theory developed by Walter R. Evans which can determine stability of the system. The root locus plots the poles of the closed loop transfer function in the complex s-plane as a function of a gain parameter.

P Controller

$$G(s)H(s) = \frac{3.76}{s^2 + 1.9136s + 3.2638}$$

MATLAB Code

```
sysGH=tf([3.76],[1 1.9136 3.2638])
rlocus(sysGH)
sgrid
rlocfind
```

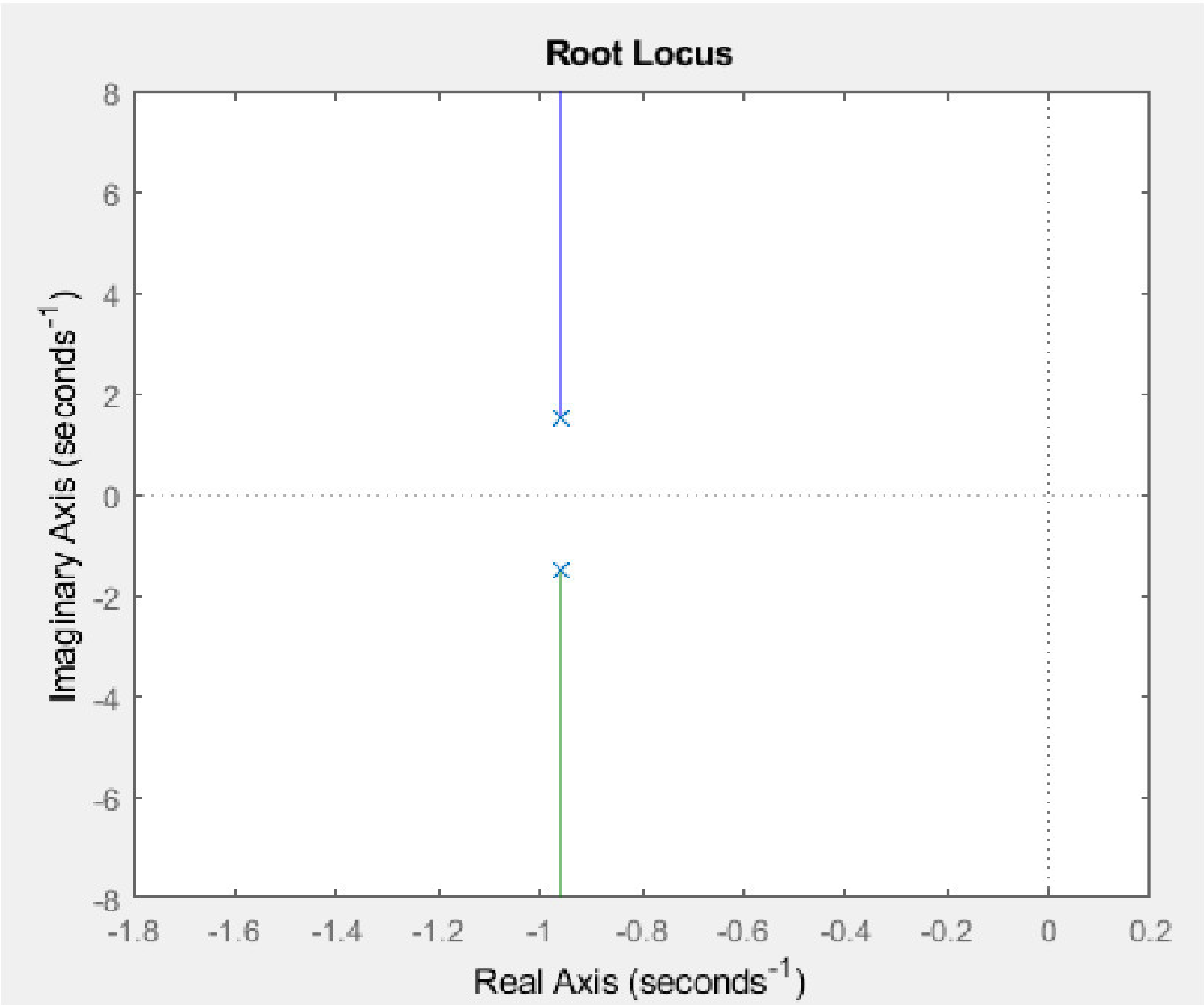


Figure 5: Root Locus of P-Controller

The root locus of the open loop function clearly shows how the poles originally existed in the 3 rd and 4th quadrant of the s-plane. Therefore, the open loop plant is stable. Moreover, it must be noted that the root locus never passes through the imaginary axis. Therefore, it never becomes marginally stable and Ziegler Nichols method cannot be applied for PID controller tuning.

This root locus is designed for proportional controller in which we assume the Kp values. This root locus clearly shows how the poles existed in third and fourth quadrant of the S-plane. Hence, the open loop plant obtained is stable. It is also noted that root locus doesn't pass through an imaginary axis and that is the reason it never becomes marginally stable and hence, Ziegler Nichols method can't be used for PID controller tuning.

PD Controller:

$$G(s)H(s) = \frac{3.76(s+20)}{s^2 + 1.9136s + 3.2638}$$

MATLAB Code

```
sysGH=tf(3.76*[1 20],[1 1.9136 3.2638])
rlocus(sysGH)
sgrid
rlocfind
```

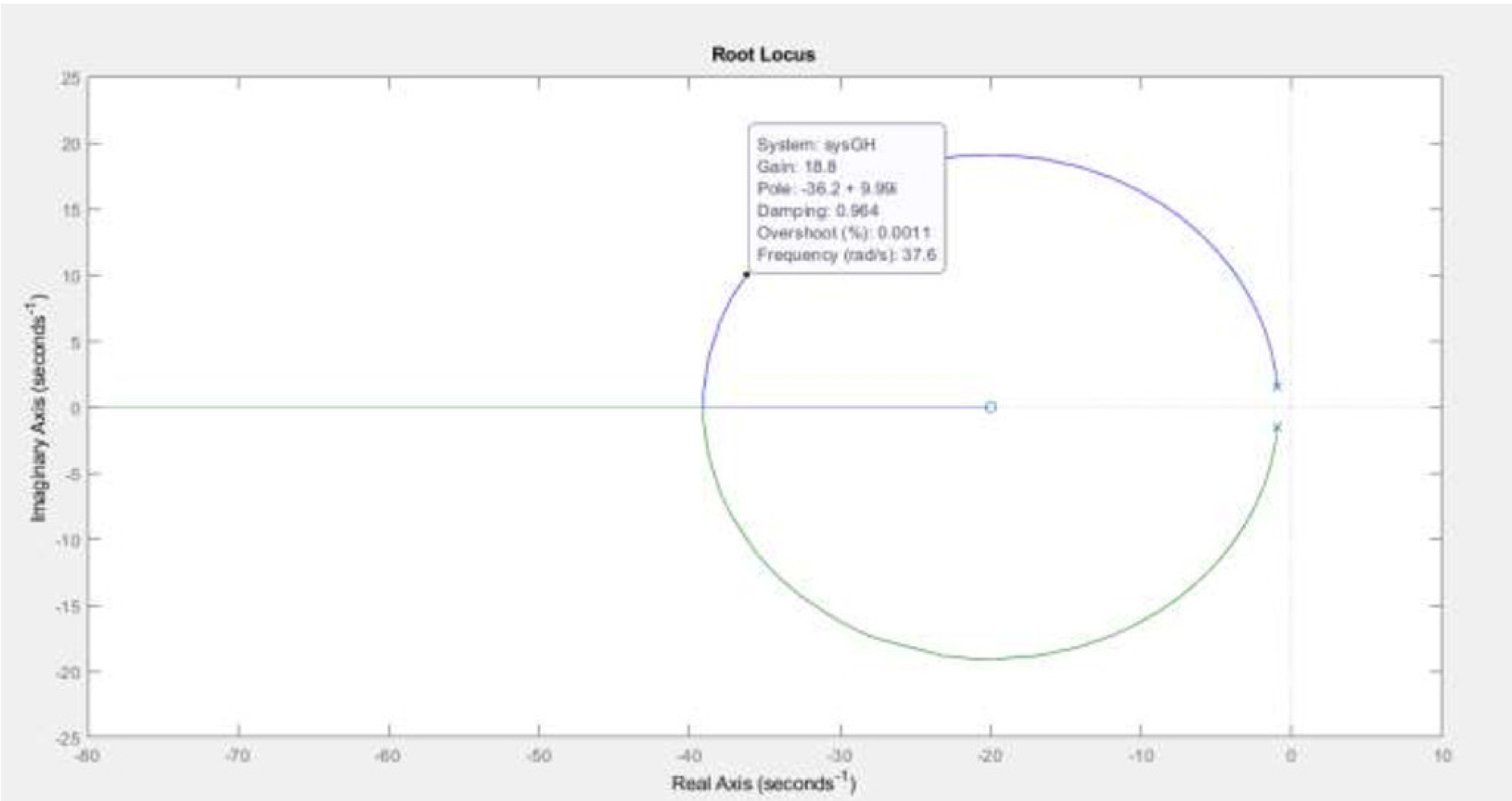


Figure 6: Root Locus of PD-Controller

Gain obtained is the value of Kd. This value is obtained by shifting the root locus center to s=-20. While the Ki value in this case is assumed in this case. We now resort to using the root locus technique to design our controller. The idea is to add poles and zeroes (equivalent to adding compensator gains) to our root locus and try to manipulate the step response, such that it meets our design requirements. To get desired characteristics, we selected different points on the root locus. The Kd, Kp and ki values vary as we move along the locus.

Simulink Model

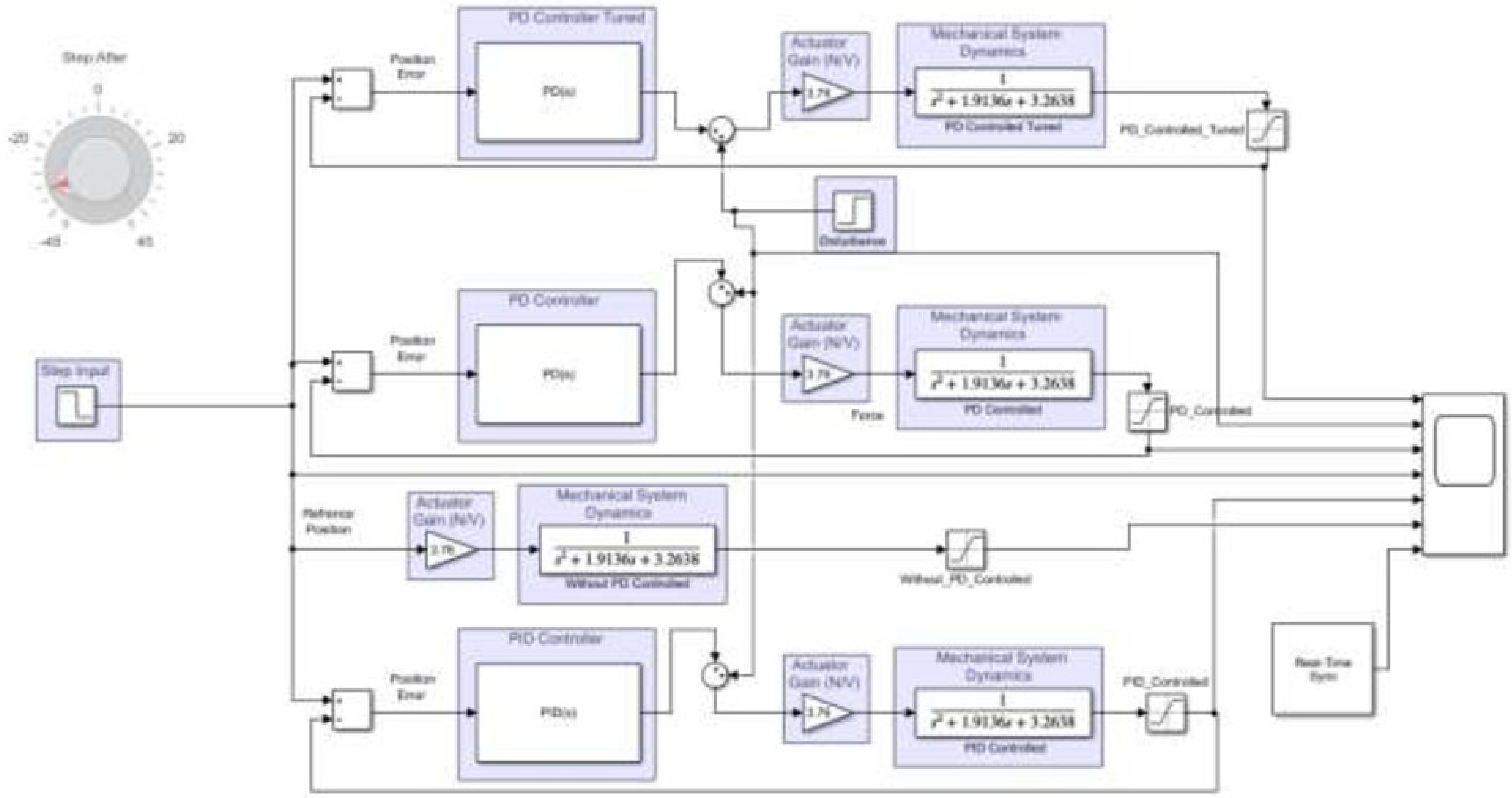


Figure 7: Simulink Model with PID Controller, PD Controller & PD Controller with Tuning

The following Simulink model was used to design the PD & PID controllers and simulate the step response. The knob controls the step input in case of real time simulation. The model also contains saturators that limits the angle from +45 to -45 degrees. The disturbance has been introduced just before the plant by applying a step input of -10 after 1 seconds. This allowed the group to also analyze the performance requirement pertaining to disturbance rejection of the system.

PD & PID Controller Values

Controller Parameters	Tuned	Block	Block
P	59.9003	376	50
I	n/a	n/a	50
D	7.0393	18.8	15
N	3176.8447	100	100

Performance and Robustness	Tuned	Block	Block
Rise time	0.0538 seconds	0.0162 seconds	0.0254 seconds
Settling time	0.342 seconds	0.139 seconds	0.0932 seconds
Overshoot	12 %	32.7 %	8.94 %
Peak	1.1	1.32	1.09
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	76.5 deg @ 27.8 rad/s	42.9 deg @ 71 rad/s	61.1 deg @ 51.8 rad/s
Closed-loop stability	Stable	Stable	Stable

Figure 8: Values of gains in Controllers with their Performance Parameters

The autotune feature in Simulink provided the following values for the PID controller. It must be noted that filter coefficient (N) is a low pass filter attached to the derivative controller in order to reject high frequency noise form the sensor.

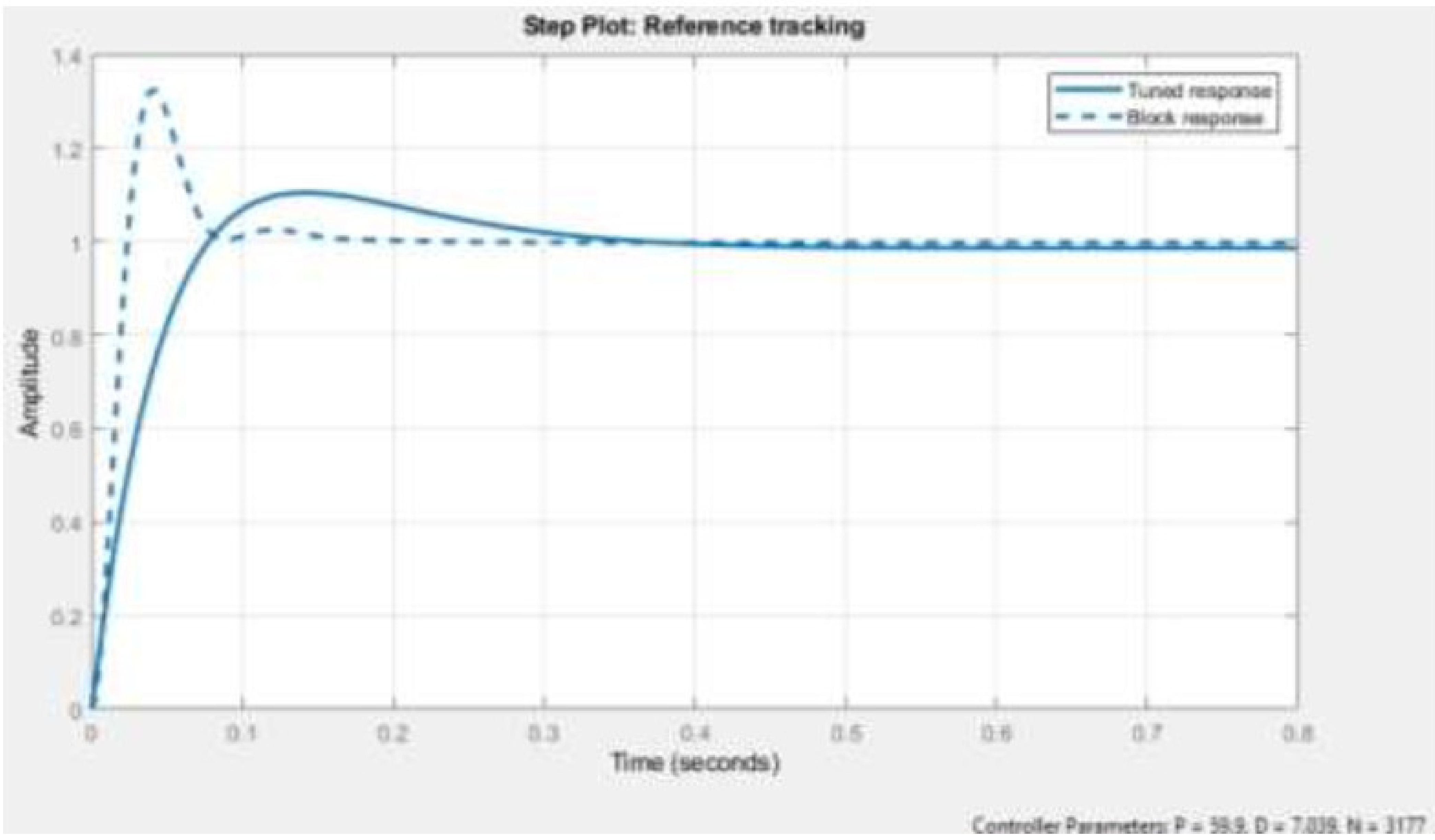


Figure 9: PD Controller vs PD Controller after Tuning

-10 Degree Step Input

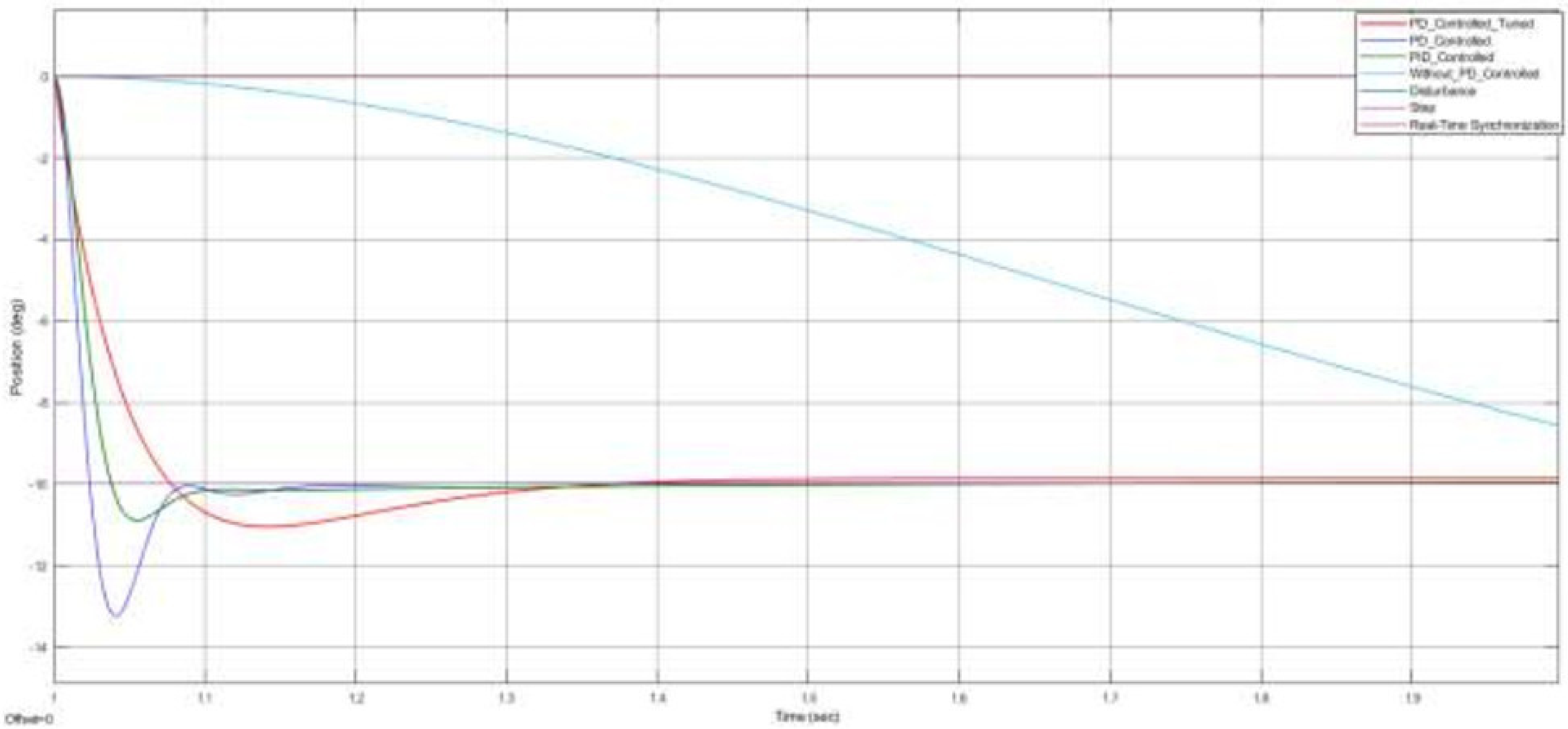


Figure 10: Response of System with -10 degree step input

This is the first plot which shows how the system responds to a step input of 13 degrees. The settling time now is around 2 seconds. Which is 52% faster response time than the open loop transfer function.

Real Time Simulation

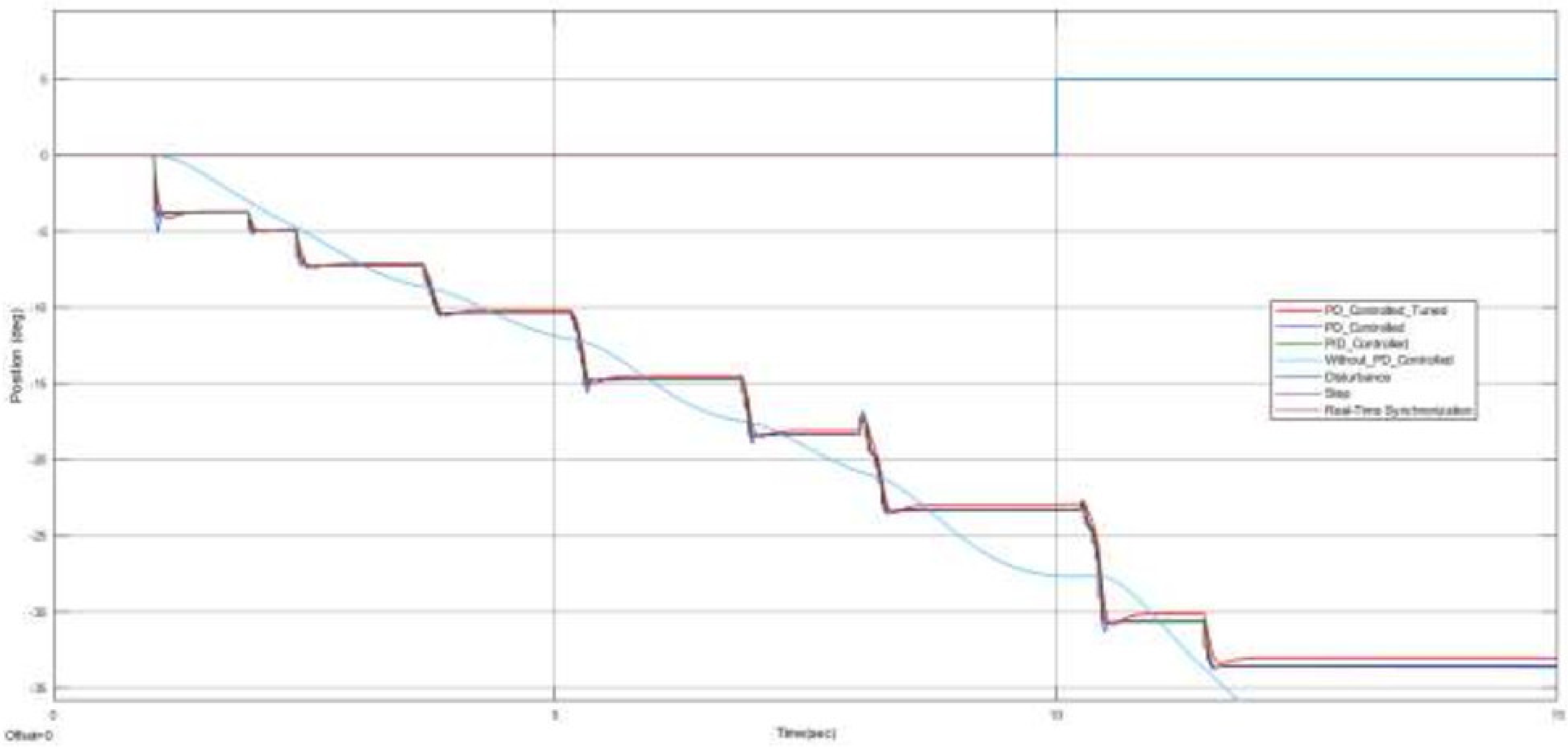


Figure 11: Response of System to Real-time Input

Stability Analysis

The stability analysis is of particular interest to control designers as it indicated the system's tendency to be stable or unstable while it is functioning. We have used the proportional gain (Kp) to find out the poles of the system. With the help of MATLAB, we were able to plot these poles and find the stability of the system.

One of the general performance requirements of a good control system is a measure of its closed loop stability. A system is said to be stable if its output is under control. Otherwise, it is said to be unstable. A stable system produces a bounded output for a given bounded input. We can classify the systems based on stability as follows.

The first type is a Stable System. A system is said to be stable if "A bounded input produces a bounded output", which basically means that the output of the system must be controlled. For a system to be perfectly stable, the poles of the systems should lie on the left side of plane that is plotted on MATLAB. This condition is true for both open and closed loop systems. These responses are usually of a decaying nature.

The second major category is the Marginally Stable System. A system is said to be marginally stable if it produces constant amplitude and frequency through a given bounded input. In MATLAB, we can term a system as such in two cases, if the poles lie on the imaginary axis, or if one of the poles is negative while the other one lies on the origin.

The third category is the Unstable System, which means the presence of "an unbounded output with a bounded input", meaning the output is not in our control. These plots lie on the right side of the MATLAB plot, and give an exponentially, or linearly increasing response.

Values of Kp were assumed and inputted into the system to see the response of the system. The graphs below display the results of each Kp value.

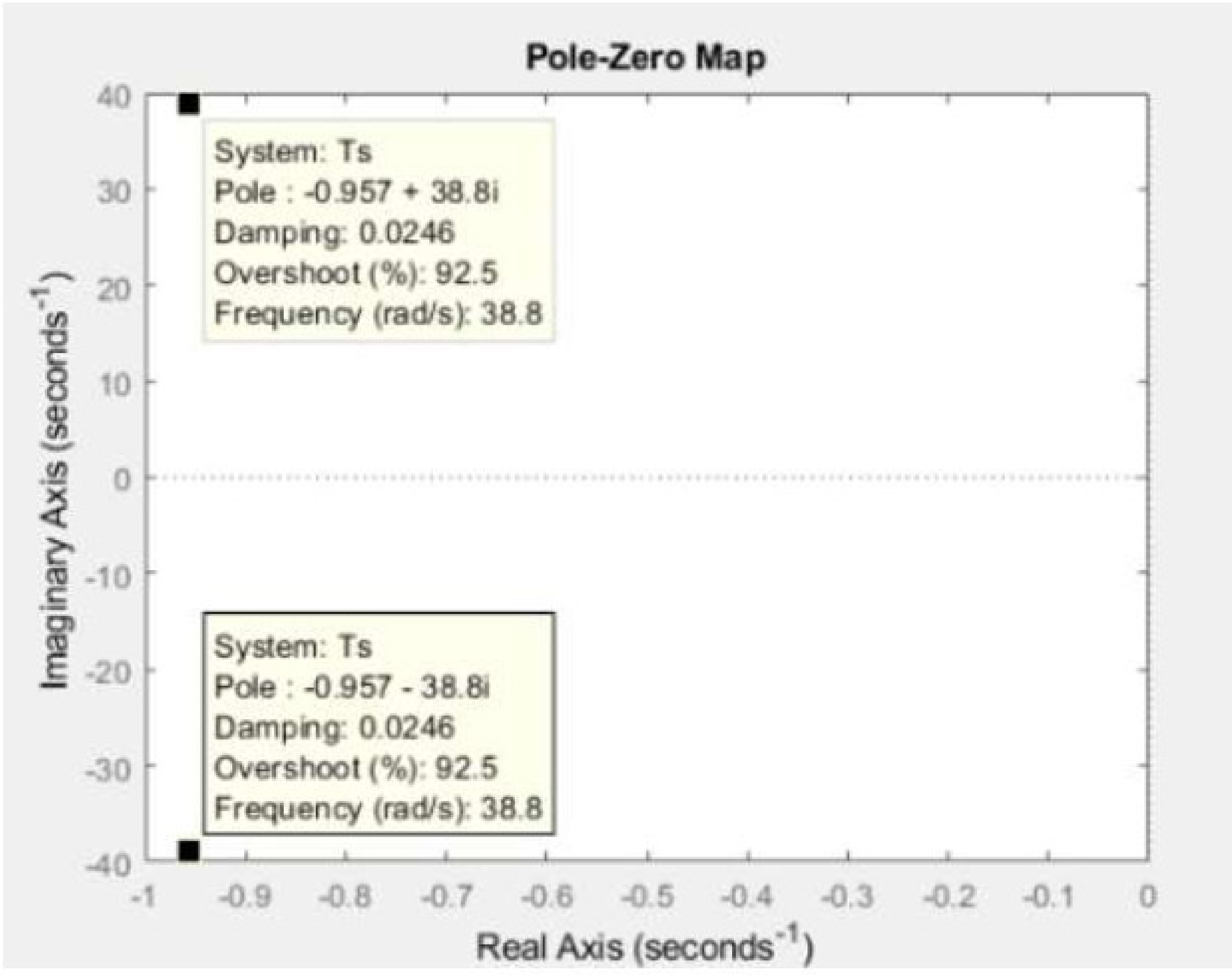


Figure 12: Pole-Zero Map of Kp=400

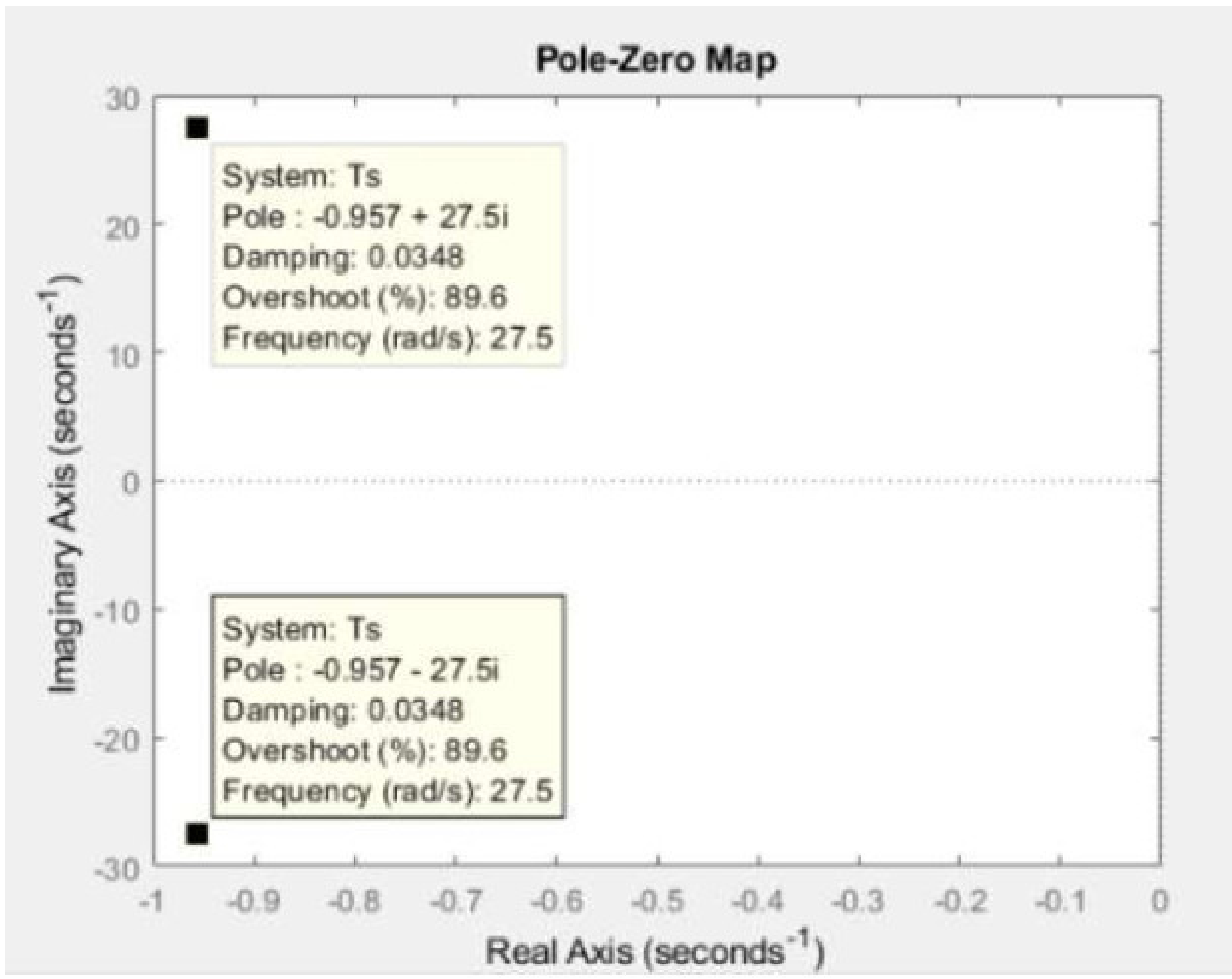


Figure 13:: Pole-Zero Map of Kp=200

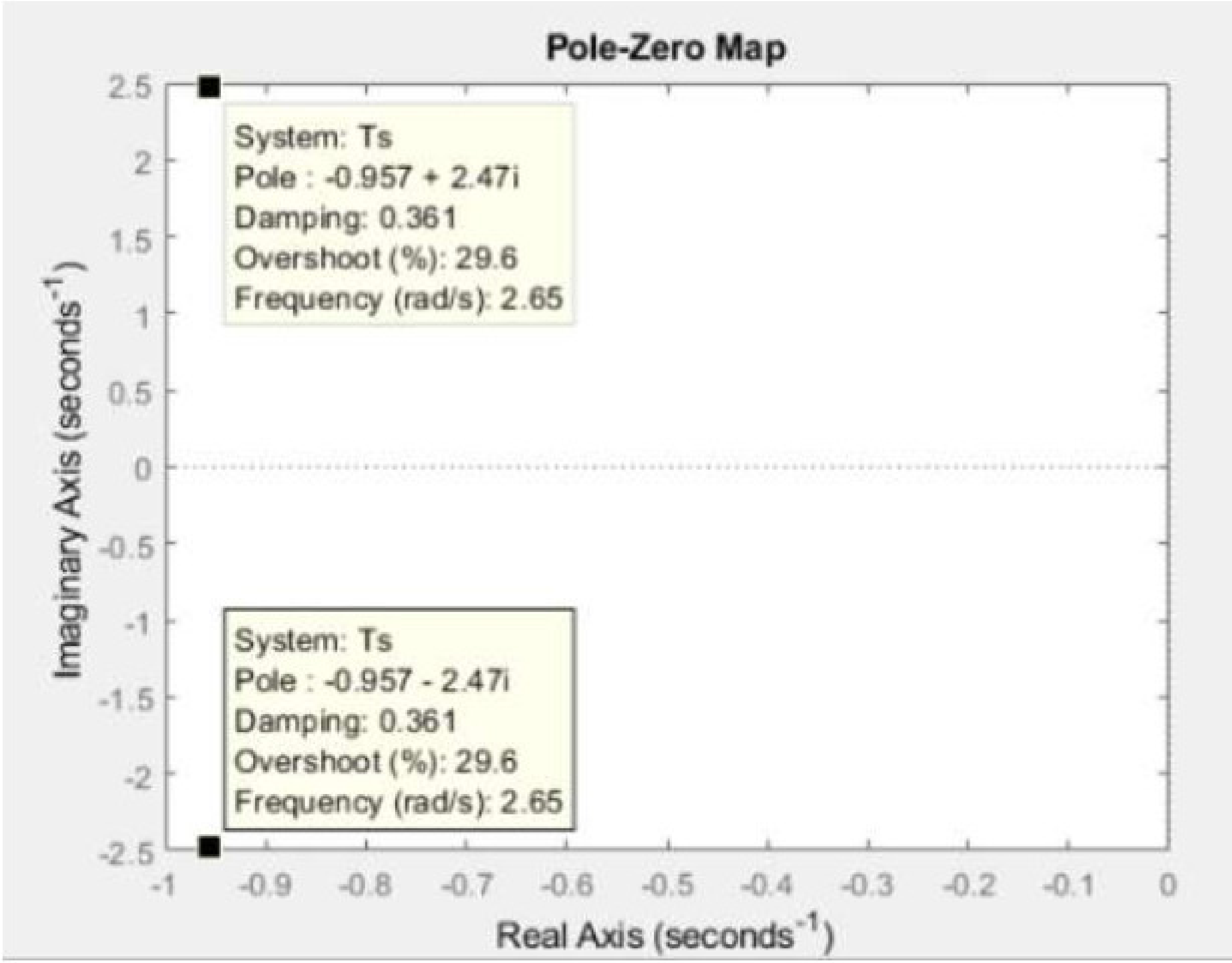


Figure 14:: Pole-Zero Map of Kp=1

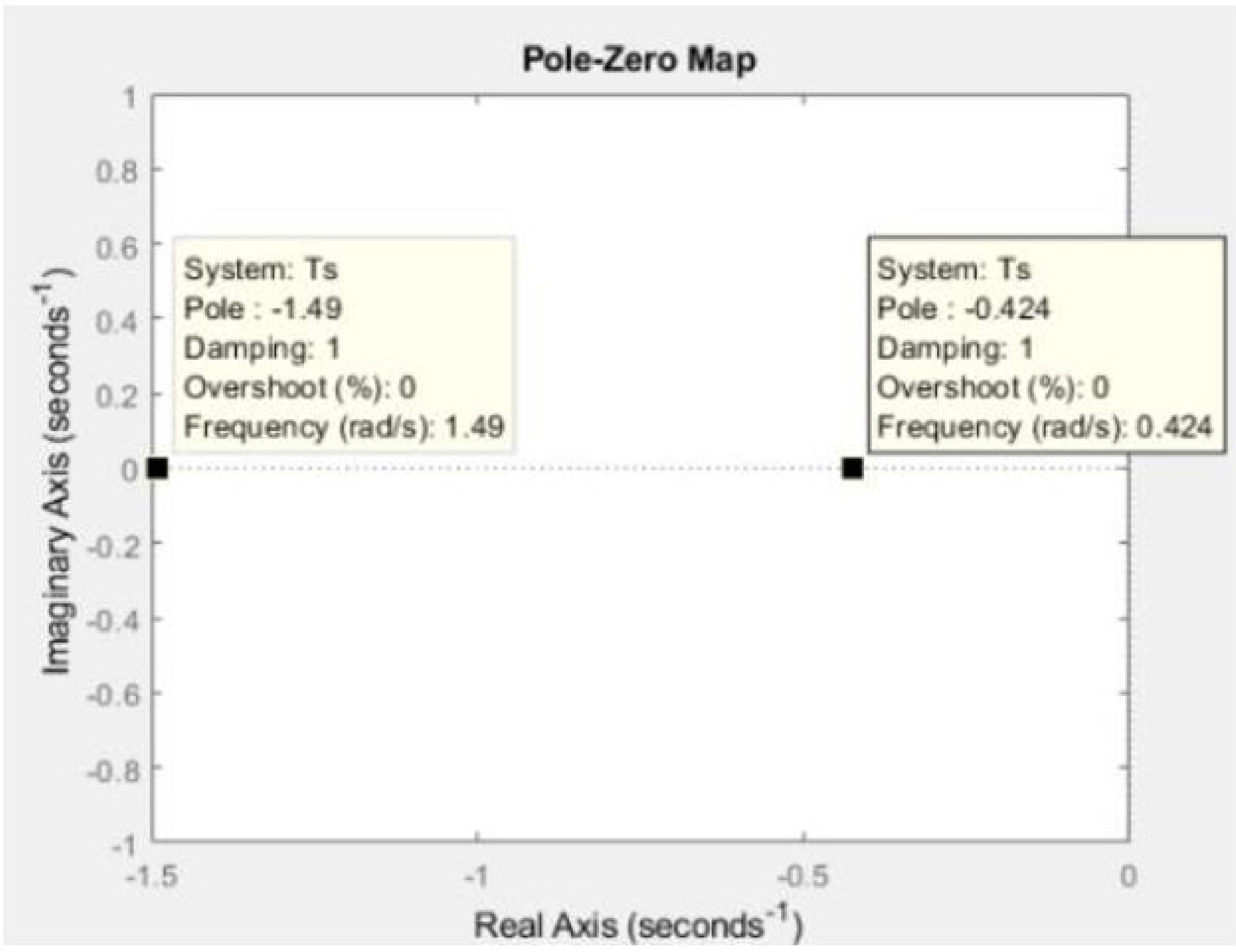


Figure 15: Pole-Zero Map of Kp=-0.7

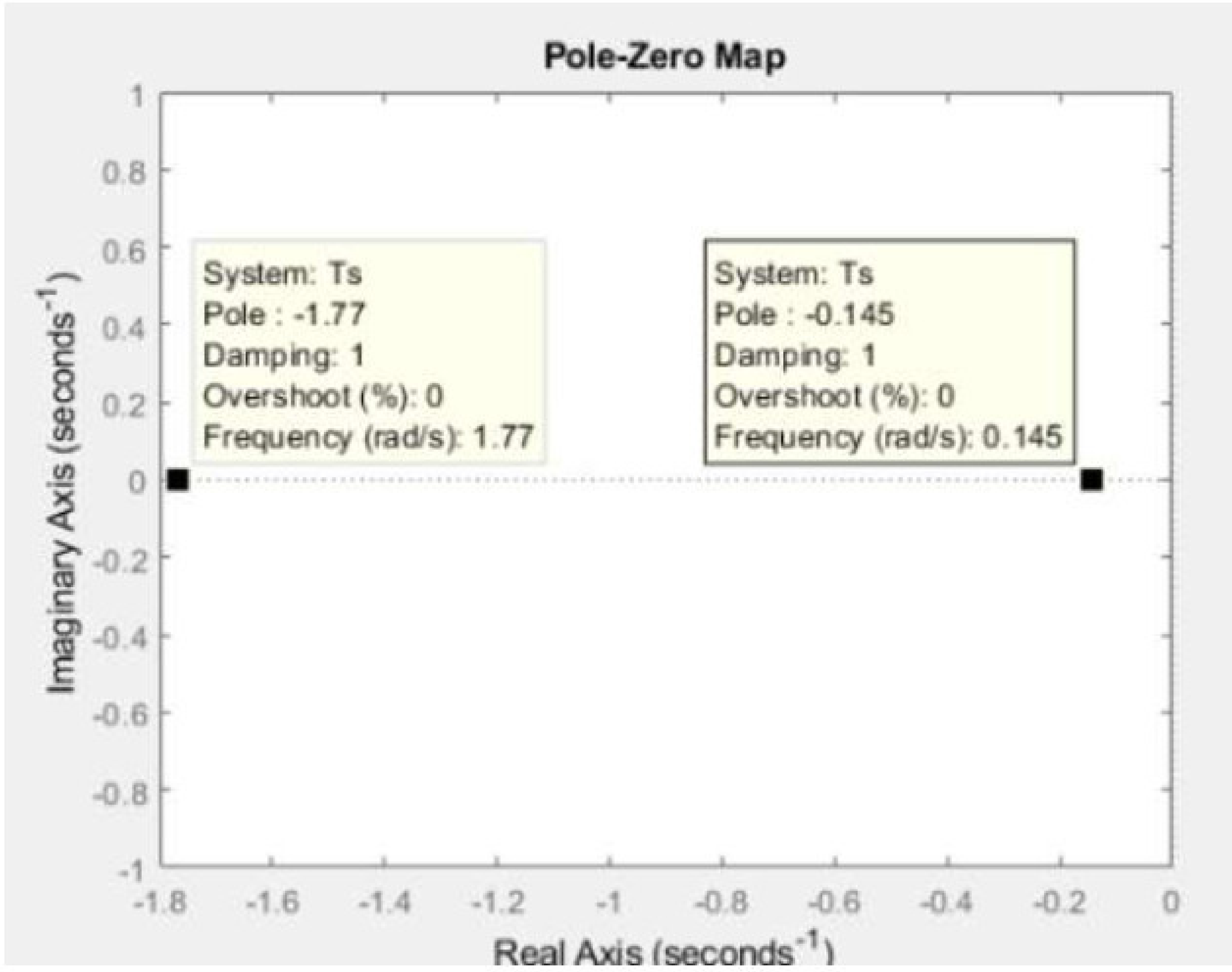


Figure 16: Pole-Zero Map of Kp=-0.8

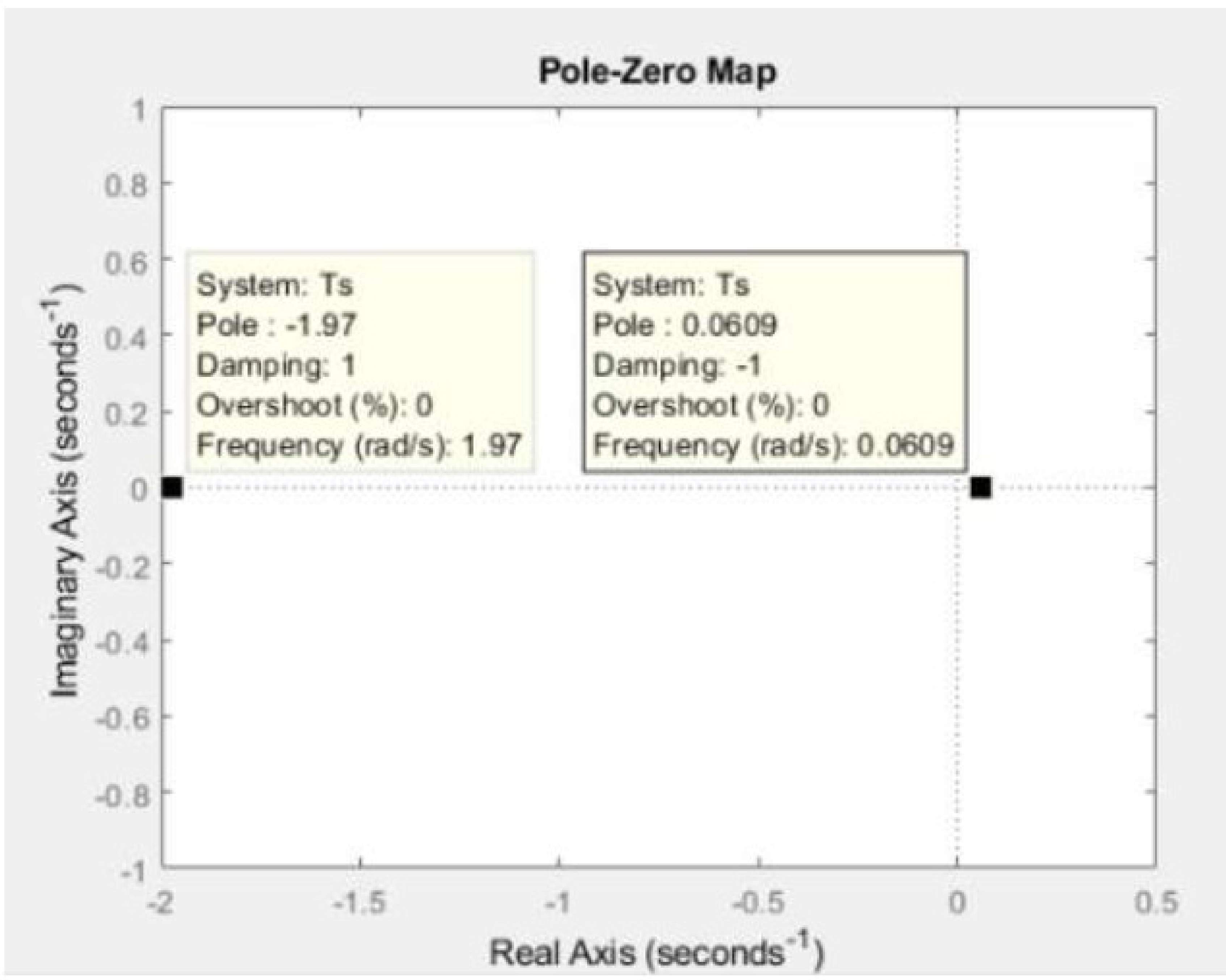


Figure 17: Pole-Zero Map of Kp=-0.9

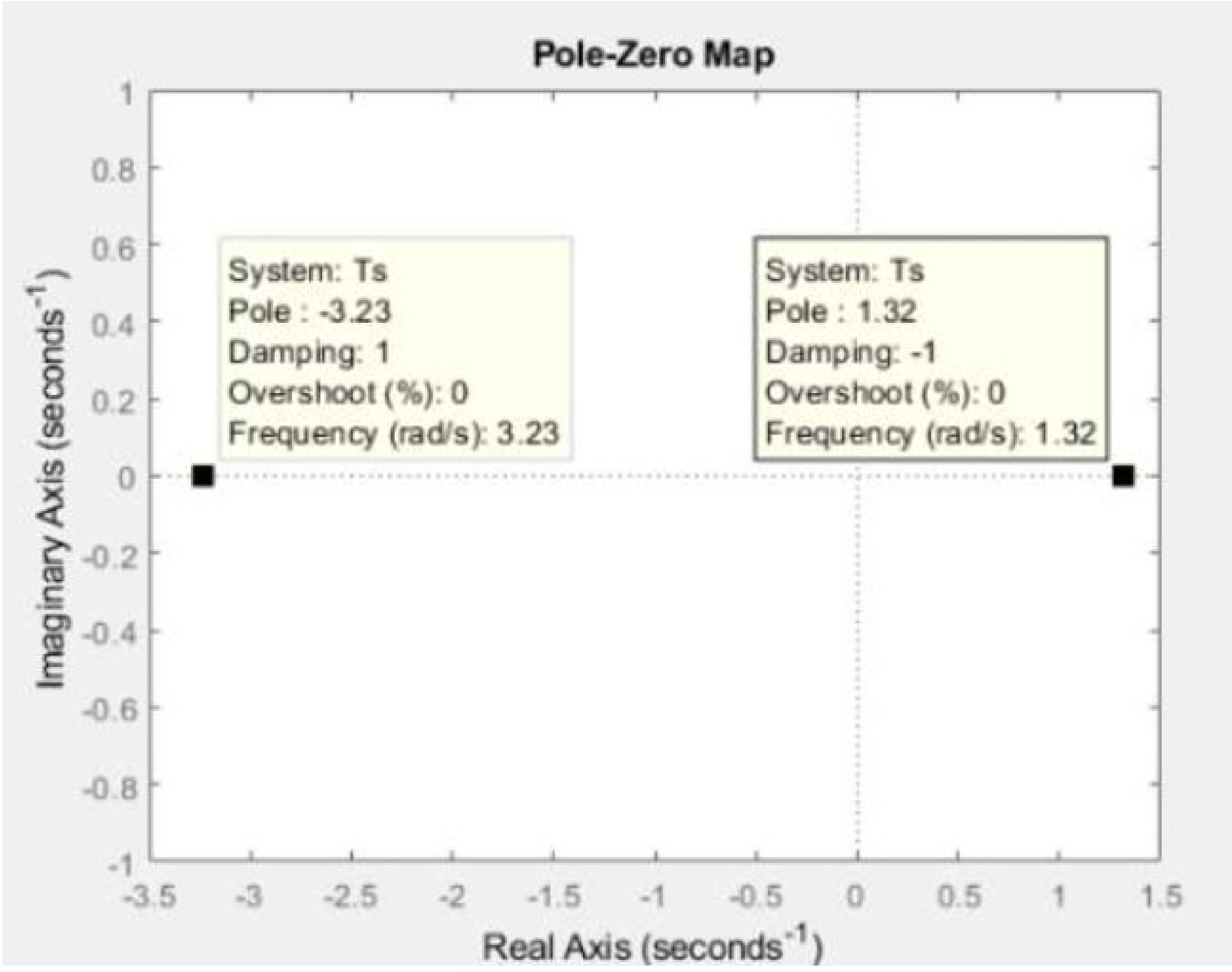


Figure 18: Pole-Zero Map of Kp=-2

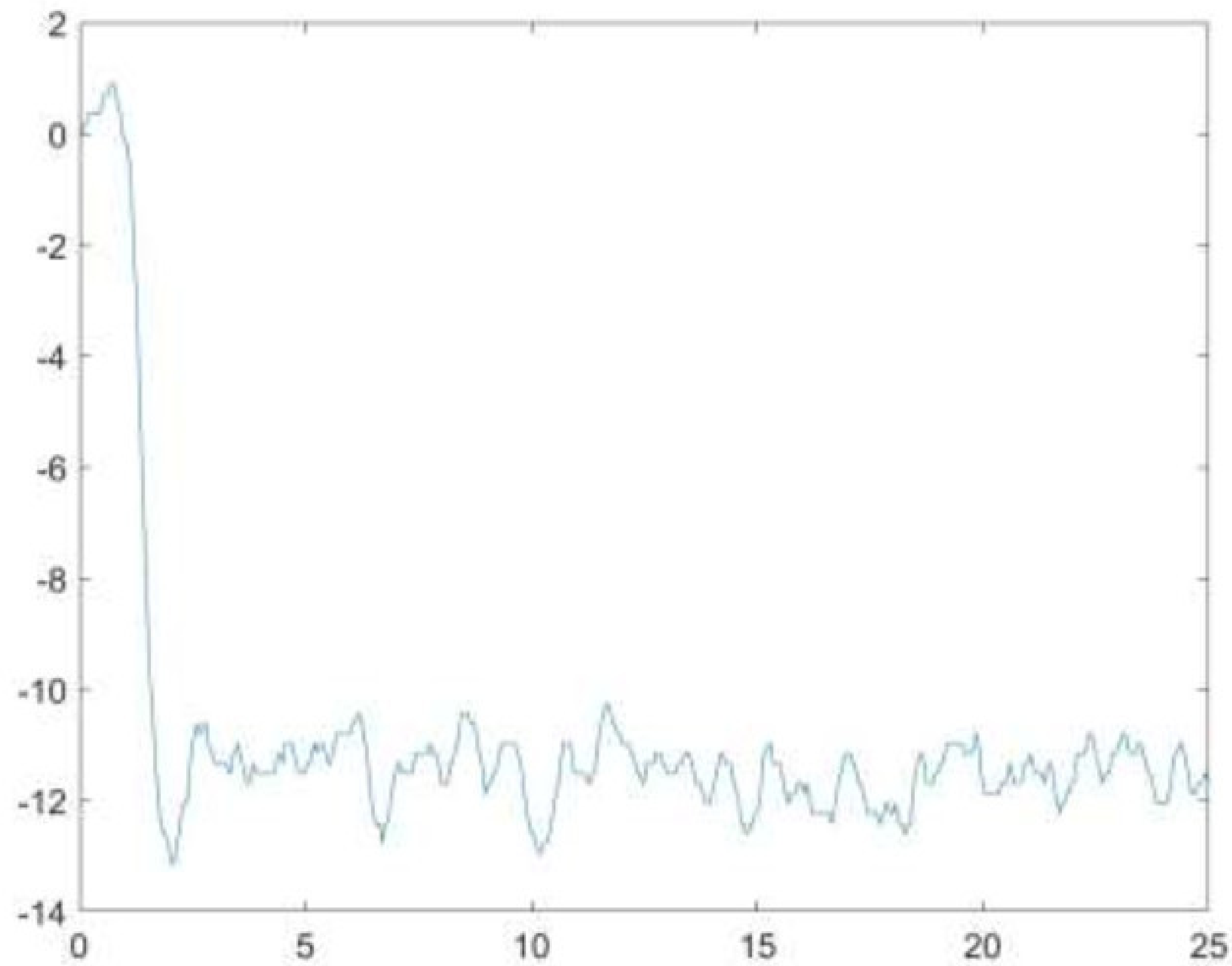
Based on the analysis performed to test the stability of the designed control system by varying kp values and determining the pole values, it was found that the controller designed is stable for kp values greater than 1. Between 1 and -0.9 kp value, the poles are not present in the imaginary axis but still negative (on the left axis) nonetheless and therefore, the system is said to be marginally stable. Below -0.9, however, the system shifts entirely to the right or positive axis and therefore is no longer stable.

Control Gain (Kp)	Closed Loop Poles	Stability Status
400	-0.957±38.8i	Stable
200	-0.957±27.5i	Stable
1	-0.957±2.47i	Stable
-0.7	-1.49, -0.424	Marginally Stable
-0.8	-1.77, -0.145	Marginally Stable
-0.9	-1.97, 0.0609	Un-Stable
-2	-3.23, 1.32	Un-Stable

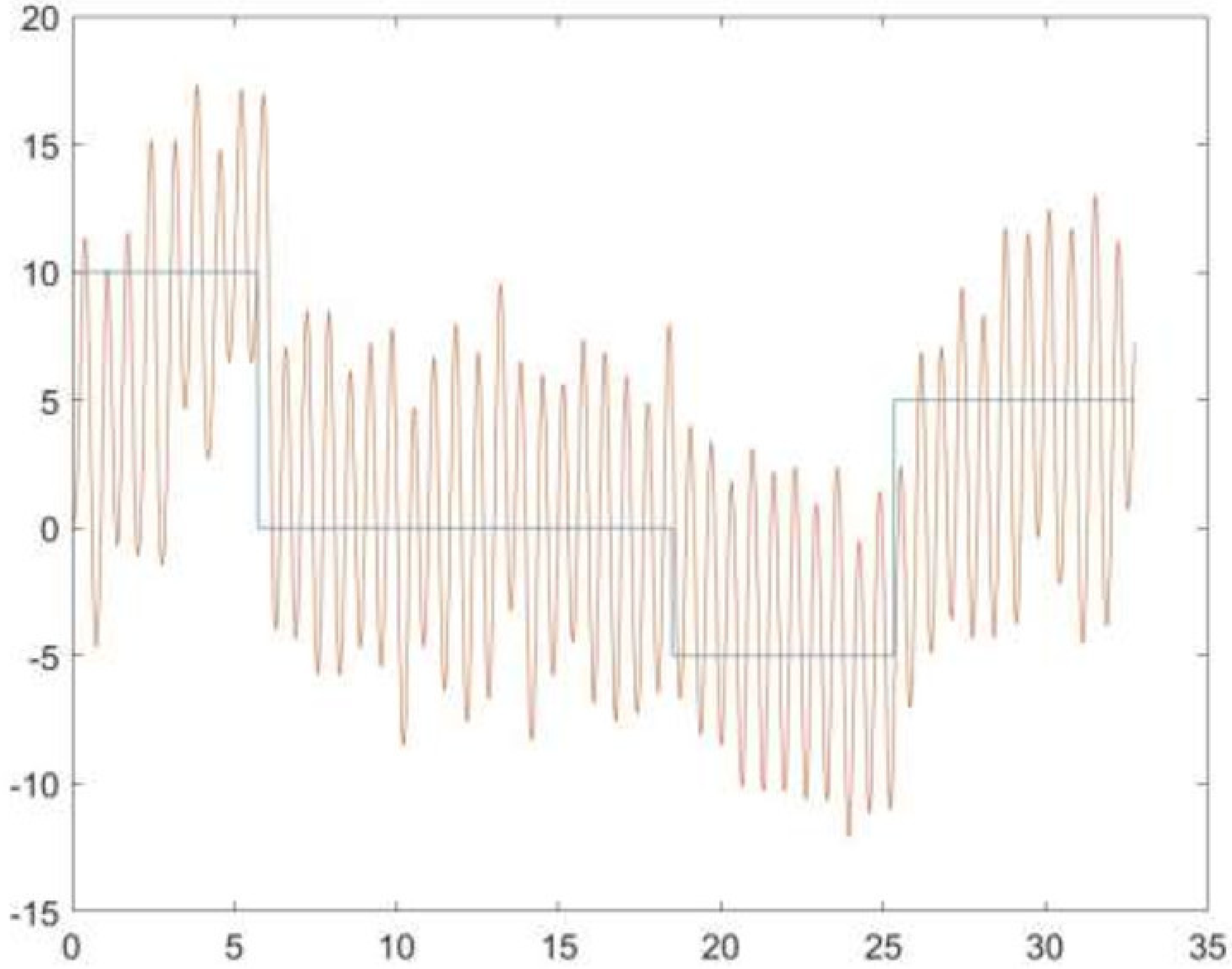
Conclusion:

The system's transfer functions were derived from the characteristic equations of the 1-DOF tailplane motions and was modeled in Simulink. Using Simulink augmented space, the system response to rudder deflection-step inputs was plotted and analyzed for various controller gain values such that the system stability characteristics could be observed. The PID controller was designed against several system requirements including open and closed loop stability, disturbance response, response time and overshoot; an iterative process allowed a system to be developed with a PID controller that, in real-time, performed on test-rig and the tunes PID controller satisfied all designed criteria. Based on the analysis performed to test the stability of the designed control system by varying kp values and determining the pole values, it was found that the controller designed is stable for kp values greater than 1. Between 1 and -0.9 kp value, the poles are not present in the imaginary axis but still negative (on the left axis) nonetheless and therefore, the system is said to be marginally stable. Below -0.9, however, the system shifts entirely to the right or positive axis and therefore is no longer stable. We check it for open loop data and closed loop Implementation Process

Open Loop Data Result



Closed Loop Implementation Responses



References:

1. Stengel, R. (2004). Flight Dynamics. Princeton University Press, Princeton, NJ
2. S.Jisha and R.B. Aswin 'Pitch Control of Flight System using Dynamic Inversion and PID Controller' International Journal of Engineering Research & Technology (IJERT), Vol. 4 Issue 07, July-2015
3. Kluever, C. A. (2015). *Dynamic Systems: Modeling, Simulation, and Control* (1st ed.). Wiley.
4. Akyürek, Şeyma & Özden, Gizem & Atlas, Emre & Kasnakoğlu, Coşku & Kaynak, Ünver. (2016). Design of a Flight Stabilizer System and Automatic Control Using HIL Test Platform. International Journal of Mechanical Engineering and Robotics Research Vol. 5, No. 1, January 2016 10.18178/ijmerr.5.1.77-81
5. S. Jagannathan 'PID Based Rudder Controller' International Journal of Engineering Research and Development, Volume 4, Issue 4 (October 2012)
6. S.Jisha and R.B. Aswin 'Pitch Control of Flight System using Dynamic Inversion and PID Controller' International Journal of Engineering Research & Technology (IJERT), Vol. 4 Issue 07, July-2015
7. Yasir M. Khan, Sarvat Mushtaq Ahmed, Mohsin Ali, Masroor Khan, " Flight dynamics and parametric modelling of a 2-DOF lab aircraft," Proceedings of institutions of Mechanical engineers part G Journal of Aerospace Engineering, issue july 2018.