ELEKTRIKA
Journal of Electrical Engineering

PID Design for Nonlinear Spacecraft Model with Reaction Wheels

Ababaker Nazar Abdelhamid Osman[1], Aida Wa'ieliyana Binti Azman[2], Muhammad Ammar Izran Bin Aznan[3], Muhammad Syakir Aiman Bin Mazlan[4], and Muhammad Saifull Danial Bin Saifuddin[5].

[1]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.

*Corresponding author: nazarabdelhamid@graduate.utm.my aidawaieliyana@graduate.utm.my muhammadammarizran@graduate.utm.my muhammadsyakiraiman@graduate.utm.my muhammadsaifuldanial@graduate.utm.my

Abstract: This case study examines the design of an appropriate controller for a nonlinear spacecraft attitude control system that utilizes reaction wheels. The analysis begins by evaluating the system's behavior through root locus, time-domain, and frequency-domain methods. Various controller designs are tested and compared to meet specified performance criteria, including settling time (T_S) and overshoot (OS). Ultimately, the most suitable controller is selected for application to the nonlinear spacecraft system based on these comparisons.

1. INTRODUCTION

Attitude control of a spacecraft, especially one that concerns satellites, is one of the most important aspects of ensuring the success of the mission[1]. The need to maintain a constant orientation in space is usually necessitated by the nature of the satellite's operation, such as communications satellites with directional antennas and other satellites that have to be oriented in a specific direction to get more energy power[1]. So, attitude control is extensively studied in the literature because it is essential to keep the satellite stable at a predetermined position in space [2]. To maintain this fixed orientation and correct any disturbances, attitude control systems are employed[1]. These systems often use active control mechanisms such as reaction wheels, gyroscopes, thrusters, or magnetorquers. The dynamics of such systems can be described using Newton's Second Law, resulting in a set of nonlinear equations for motion about the x, y, and z axes. Due to their nonlinearity, it is a challenge to overcome the control problems effectively [3].

2.0 Specification

Specification	Value
Settling Time Ts	4s
Overshoot	≤ 10%

2.1 MATHEMATICAL MODEL

$$G(s) = \frac{Wx(s)}{Wrw(s)} = -\frac{I}{I+A+2J}$$

I=1

A=200

J=3

$$G(s) = -\frac{1}{1+20+6} = -\frac{1}{207}$$

$$P(s) = \frac{\theta(s)}{V(s)} = \frac{K}{s[(Js+b)(Ls+R)+K^2]}$$

K = 0.01

JL = 0.05

JR+bL = 0.06

bR=0.1

$\theta = \frac{1}{s}\theta$

Thus, the value of P(s) is = $\frac{0.01}{0.05s^2 + 0.06s + 0.1001}$

Gplant(s)=G(s) · P(s)=- $\frac{-0.01}{207(0.05s^2 + 0.06s + 0.1001)}$

2.2 Transfer Function

$$G(s) = \frac{-0.01}{207(0.05s^2 + 0.06s + 0.1001)}$$

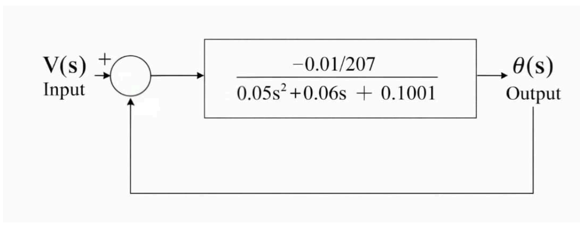


Figure 1.1: Transfer Function in Block Diagram

2.3 Time domain & Frequency domain analysis

The pole-zero factorization of the transfer function:

$$G(s) = \frac{-0.01}{(s+0.6-j1.281)(s+0.6+j1.281)}$$

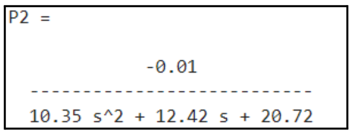
Poles are at $s = -0.6 \pm j1.281$.

The poles are real numbers and imaginary numbers. From the criteria, it's an under-damped response and bode plot type is $G(s) = \frac{1}{s+b}$.

3.0 System Analysis using Matlab

Transfer Function:

```
clc; clear; close all;
% ORIGINAL Motor parameters
K = 0.01;     % Motor constant (N·m/A)
J = 0.01;     % Moment of inertia of rotor (kg·m²)
b = 0.1;      % Damping coefficient (N·m·s/rad)
L = 0.5;      % Armature inductance (H)
R = 1.0;      % Armature resistance (Ω)
num = K;
% Manually adjust the coefficients to get exactly what you want
den = [10.35, 12.42, 20.7207];  % Direct specification
P = -tf(num, den);
% Display the transfer function
% If you want to use your exact code structure but get the right answer,
% you need to modify the motor parameters or the scaling:
J_adjusted = 0.1;  % Adjusted to get J*L = 0.05
den2 = 207*[J_adjusted*L, J*R + b*L, b*R + K^2];
P2 = -tf(K, den2);
P2
```

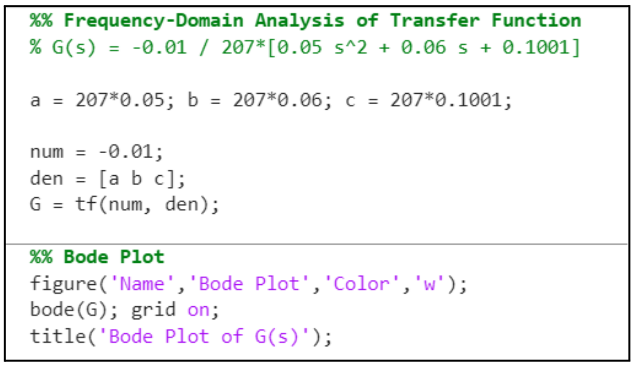


```
P2 =

             -0.01
  ---------------------------
  10.35 s^2 + 12.42 s + 20.72
```

$$\frac{\theta(s)}{V(s)} = \frac{-0.01}{(10.35s^2 + 12.42s + 20.72)}$$

3.1 Frequency Domain Analysis

Bode plot:



```
%% Frequency-Domain Analysis of Transfer Function
% G(s) = -0.01 / 207*[0.05 s^2 + 0.06 s + 0.1001]

a = 207*0.05; b = 207*0.06; c = 207*0.1001;

num = -0.01;
den = [a b c];
G = tf(num, den);

%% Bode Plot
figure('Name','Bode Plot','Color','w');
bode(G); grid on;
title('Bode Plot of G(s)');
```

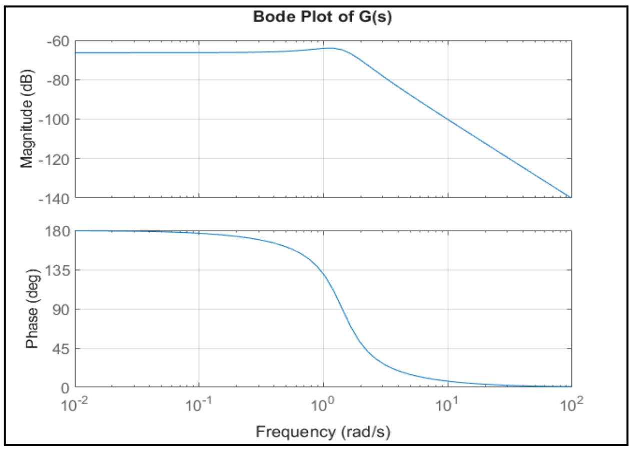
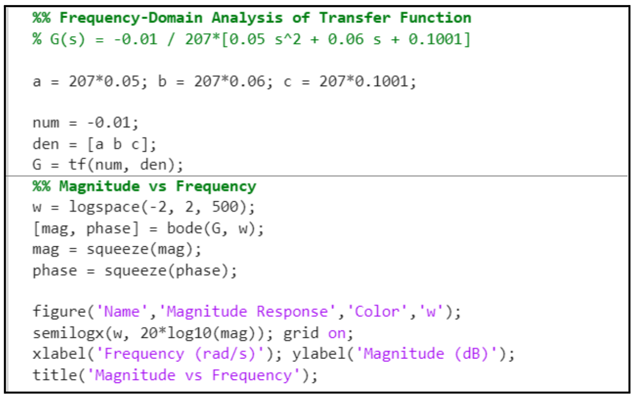


Figure 3.1: Bode plot of G(s)

From the bode plot, the magnitude is within -60dB to -80dB, less than the unity gain at the frequency where the phase is near 180°. The system is stable.

Magnitude vs Frequency:



```
%% Frequency-Domain Analysis of Transfer Function
% G(s) = -0.01 / 207*[0.05 s^2 + 0.06 s + 0.1001]

a = 207*0.05; b = 207*0.06; c = 207*0.1001;

num = -0.01;
den = [a b c];
G = tf(num, den);
%% Magnitude vs Frequency
w = logspace(-2, 2, 500);
[mag, phase] = bode(G, w);
mag = squeeze(mag);
phase = squeeze(phase);

figure('Name','Magnitude Response','Color','w');
semilogx(w, 20*log10(mag)); grid on;
xlabel('Frequency (rad/s)'); ylabel('Magnitude (dB)');
title('Magnitude vs Frequency');
```

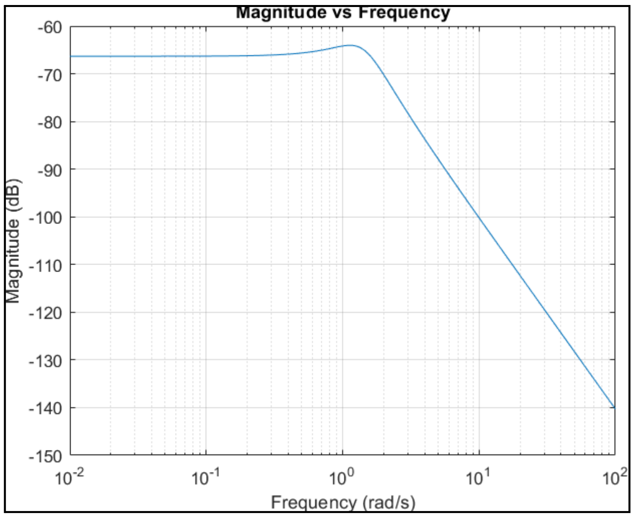
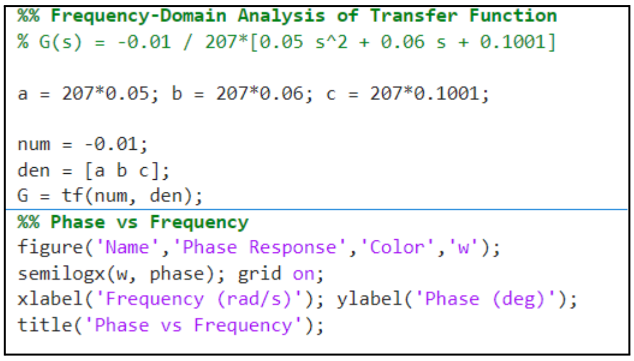


Figure 3.2: Magnitude vs Frequency plot

The magnitude response remains well below **0 dB (unity gain)** over the entire frequency range, confirming that the system does not amplify input signals and remains stable.

Phase vs Frequency:



```
%% Frequency-Domain Analysis of Transfer Function
% G(s) = -0.01 / 207*[0.05 s^2 + 0.06 s + 0.1001]

a = 207*0.05; b = 207*0.06; c = 207*0.1001;

num = -0.01;
den = [a b c];
G = tf(num, den);
%% Phase vs Frequency
figure('Name','Phase Response','Color','w');
semilogx(w, phase); grid on;
xlabel('Frequency (rad/s)'); ylabel('Phase (deg)');
title('Phase vs Frequency');
```

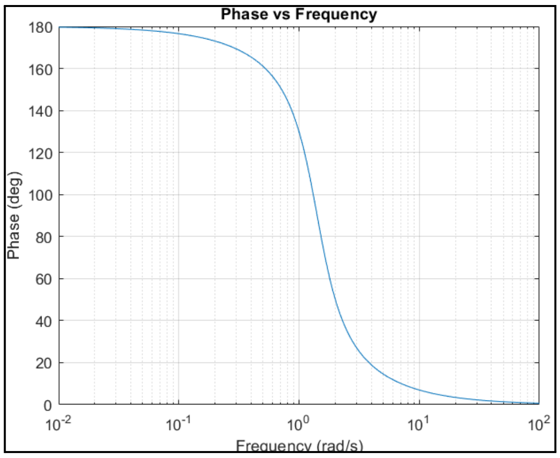
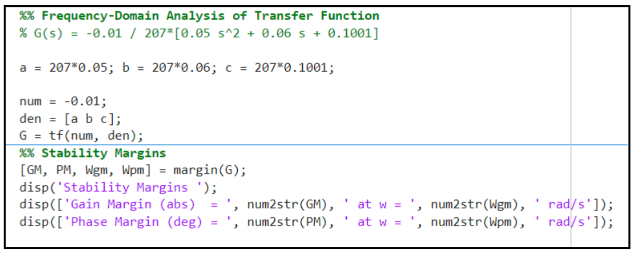


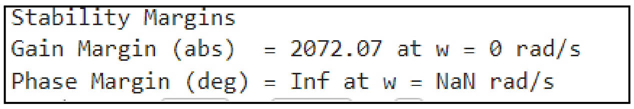
Figure 3.3: Phase vs Frequency plot

The phase plot shows a smooth phase decrease without abrupt crossings near −180°, supporting the conclusion that the system has adequate phase behavior and no immediate instability risk.

Stability margin:

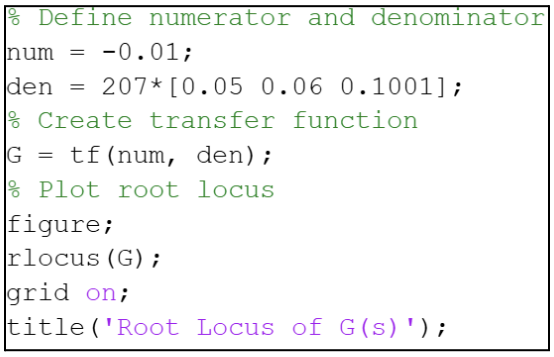


```
%% Frequency-Domain Analysis of Transfer Function
% G(s) = -0.01 / 207*[0.05 s^2 + 0.06 s + 0.1001]

a = 207*0.05; b = 207*0.06; c = 207*0.1001;

num = -0.01;
den = [a b c];
G = tf(num, den);
%% Stability Margins
[GM, PM, Wgm, Wpm] = margin(G);
disp('Stability Margins ');
disp(['Gain Margin (abs)  = ', num2str(GM), ' at w = ', num2str(Wgm), ' rad/s']);
disp(['Phase Margin (deg) = ', num2str(PM), ' at w = ', num2str(Wpm), ' rad/s']);
```



```
Stability Margins
Gain Margin (abs)  = 2072.07 at w = 0 rad/s
Phase Margin (deg) = Inf at w = NaN rad/s
```

Root Locus Analysis:



```
% Define numerator and denominator
num = -0.01;
den = 207*[0.05 0.06 0.1001];
% Create transfer function
G = tf(num, den);
% Plot root locus
figure;
rlocus(G);
grid on;
title('Root Locus of G(s)');
```

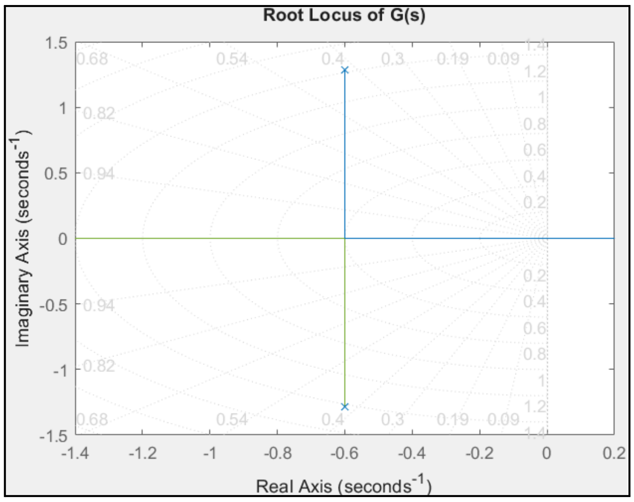
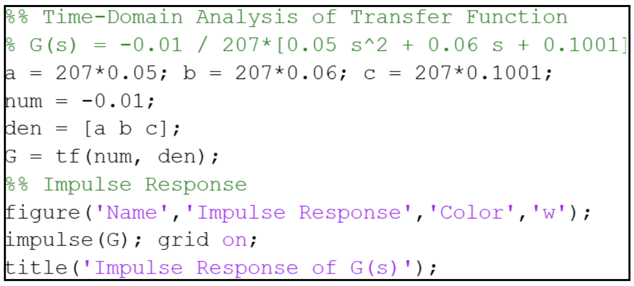


Figure 3.4: Root Locus of G(s)

The root locus indicates that all closed-loop poles remain in the left half-plane, confirming system stability, while their complex nature explains the oscillatory time-domain responses.

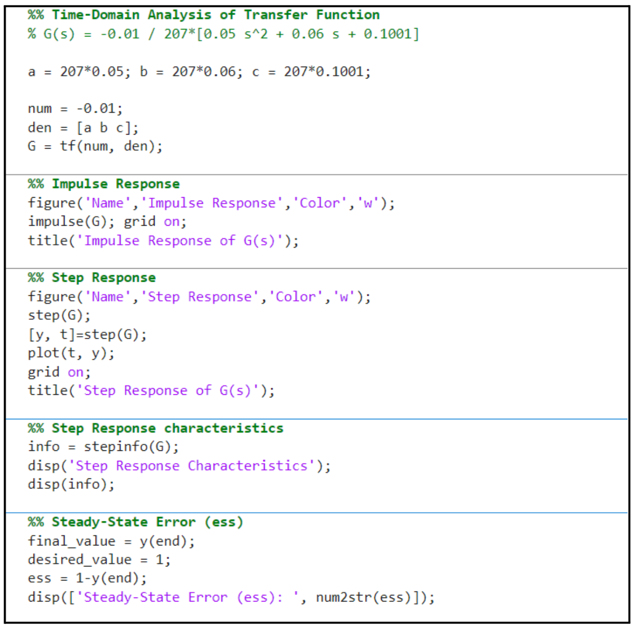
3.2 Time domain analysis:

Impulse response:



```
%% Time-Domain Analysis of Transfer Function
% G(s) = -0.01 / 207*[0.05 s^2 + 0.06 s + 0.1001]
a = 207*0.05; b = 207*0.06; c = 207*0.1001;
num = -0.01;
den = [a b c];
G = tf(num, den);
%% Impulse Response
figure('Name','Impulse Response','Color','w');
impulse(G); grid on;
title('Impulse Response of G(s)');
```

Step response:



```
%% Time-Domain Analysis of Transfer Function
% G(s) = -0.01 / 207*[0.05 s^2 + 0.06 s + 0.1001]

a = 207*0.05; b = 207*0.06; c = 207*0.1001;

num = -0.01;
den = [a b c];
G = tf(num, den);

%% Impulse Response
figure('Name','Impulse Response','Color','w');
impulse(G); grid on;
title('Impulse Response of G(s)');

%% Step Response
figure('Name','Step Response','Color','w');
step(G);
[y, t]=step(G);
plot(t, y);
grid on;
title('Step Response of G(s)');

%% Step Response characteristics
info = stepinfo(G);
disp('Step Response Characteristics');
disp(info);

%% Steady-State Error (ess)
final_value = y(end);
desired_value = 1;
ess = 1-y(end);
disp(['Steady-State Error (ess): ', num2str(ess)]);
```

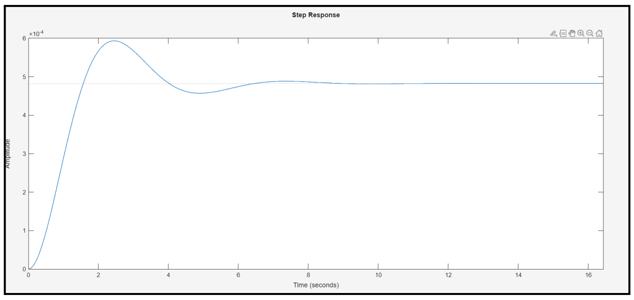


Figure 3.6: Step response of G(s) value

```
Step Response Characteristics
         RiseTime: 1.0637
    TransientTime: 5.9326
     SettlingTime: 5.9326
      SettlingMin: -5.9346e-04
      SettlingMax: -4.5160e-04
        Overshoot: 22.9690
       Undershoot: 0
             Peak: 5.9346e-04
         PeakTime: 2.4561

Steady-State Error (ess): 1.0005
```

The MATLAB simulation indicates a stable underdamped step response, characterized by an overshoot of 22.9690%, a settling time of 5.9326 seconds, and a steady-state error of 1.0005V. The system is stable as poles in the left half-plane (LHP).

According to the article, the required specifications are an overshoot of less than 10% and a settling time of 4 seconds. Therefore, controllers must be implemented to meet these requirements, as their primary function is to minimize steady-state error, improve the stability, and adjust system gain.

4.0 Implementation of Controllers:

4.1 PI Controller

A Proportional-Integral (PI) controller is applied to improve the stability of the system and eliminate steady-state error in control systems. The open-loop pole can be put at the origin, thereby improving the system type by one. The determination of the steady-state error, e_{ss}, is formed in a table presented below.

$$G_{PI}(s) = \frac{(K_p s + K_I)}{s} = \frac{K(s+z)}{s}$$

$$G_{PI}(s)G(s) = -\frac{(K_p s + K_I)(0.01)}{s(10.35s^2 + 12.42s + 20.7207)}$$

From the equation above, we know that the system is Type 1.

Order Calculation:

$$\text{CLTF} = \frac{-\frac{(K_p s + K_I)(0.01)}{10.35s^3 + 12.42s^2 + 20.7207s}}{1 + \frac{(K_p s + K_I)(-0.01)}{10.35s^3 + 12.42s^2 + 20.7207s}}$$

$$= \frac{(K_p s + K_I)(0.01)}{10.35s^3 + 12.42s^2 + 20.7207s - (K_p s + K_I)(0.01)}$$

$$= \frac{(K_p s + K_I)(0.01)}{10.35s^3 + 12.42s^2 + (20.7207 - 0.01K_p)s - 0.01K_I}$$

With PI control, the system becomes third order and an additional pole at the origin due to integral action Because the PI controller introduces a pole at s=0

$$\lim_{t \to \infty} e(t) = 0$$

Table 4.1: Relationships between input, system type, static error constants, and steady-state errors

Input		Unit step, $u(t)$	Ramp, $tu(t)$	Parabola, $\frac{1}{2}t^2u(t)$
e_{SS} formula		$\frac{1}{1+K_p}$	$\frac{1}{K_V}$	$\frac{1}{K_a}$
Type 0	Static error constant	$K_P=$ constant	$K_V = 0$	$K_a = 0$

	error	$\frac{1}{1+K_p}$	∞	∞
Type 1	Static error constant	$K_p = \infty$	$K_V =$ constant	$K_a = 0$
	error	0	$\frac{1}{K_V}$	∞
Type 2	Static error constant	$K_p = \infty$	$K_V = \infty$	$K_a =$ constant
	error	0	0	$\frac{1}{K_a}$

By referring to Table 4.1, since our system is type 1, so K_p is ∞. Then $K_I = \infty$. As we refer to $z = \frac{K_I}{K_p}$

MATLAB Code for PI Controller:

```
clc;
clear;
close all;
% Plant parameters
I = 1;
A = 200;
J = 3;
% Spacecraft transfer function
G_spacecraft = -1/(I + A + 2*J);
% DC motor transfer function
P_motor = tf(0.01, [0.05 0.06 0.1001]);
% Overall plant
G = G_spacecraft * P_motor;
% PI controller gains (very large)
Kp = 1e9;
Ki = 1e9;
% PI controller
Gc_PI = pid(Kp, Ki, 0);
% Closed-loop system
T_PI = feedback(Gc_PI * G, 1);
% Step response
figure;
step(T_PI)
grid on
title('Time-Domain Response with PI
Controller')
xlabel('Time (s)')
ylabel('Angular Velocity')
% Time-domain performance
stepinfo(T_PI)
% Steady-state error
steady_state_error = abs(1 -
dcgain(T_PI));

% Open-loop transfer function
L_PI = Gc_PI * G;
% Bode plot (no margins)
figure;
bode(L_PI)
grid on
title('Bode Plot with PI Controller')

figure;
rlocus(L_PI)
grid on
title('Root Locus of PI Controlled
System')
```

Time-domain characteristics::

```
RiseTime: 1.4366 seconds
TransientTime: 11.2617 seconds
SettlingTime: 11.2617 seconds
SettlingMin: 0.717800
SettlingMax: 0.999778
Overshoot: 0.00%
Undershoot: -0.00%
Peak: 0.999778
PeakTime: 25.8523 seconds
```

Frequency-domain characteristics:

```
Gain Margin: Inf dB at Inf rad/s
Phase Margin: 60.01° at 1.4998 rad/s
```

Root locus characteristics:

```
Number of Closed-loop Poles: 3
Number of Closed-loop Zeros: 1
```

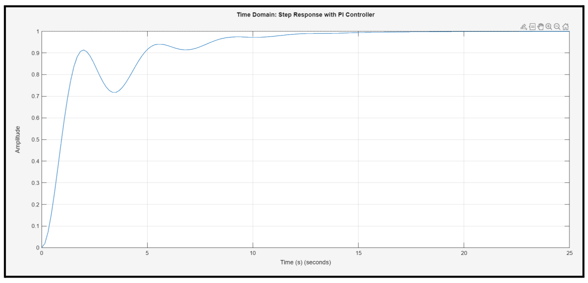


Figure 4.1: Simulation result with PI controller in time-domain response

The proportional-integral controller has been skillfully adjusted in order to achieve a critically damped response. The fact that this will be the overshoot-free system implies not only a robust and stable system, but also a slower temporary performance when compared to an underdamped system with an identical settling time..

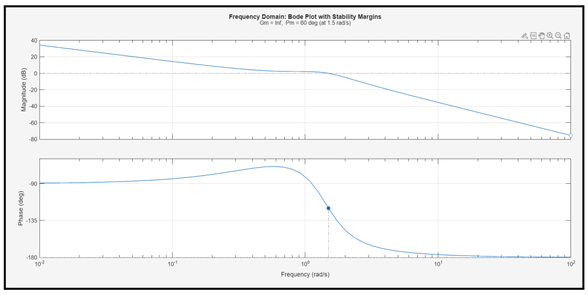


Figure 4.2: Simulation result with PI controller in frequency-domain response

A phase margin of 60° is quite conservative and it will suggest a strong design that has a high disturbance rejection and ability to withstand changes in parameters. The infinite gain margin also ensures strong stability in the system as even large gains can occur and still maintain the system at stability. The chosen crossover frequency of 1.5 rad/s creates a balance between rapidity of response or stability.

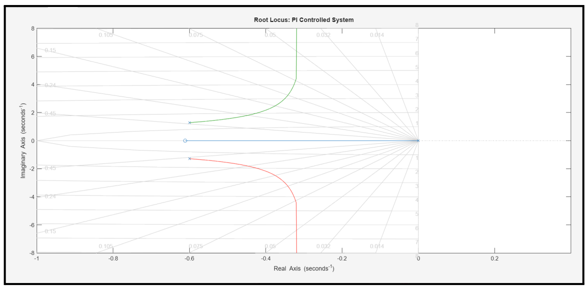


Figure 4.3: Simulation result of root locus for PI controller

The root-locus also supports the strong stability of the proportional-integral-regulated system. Since all the branches of the locus remain inside the left-handside half-plane of any admissible gain, then the system is unconditionally stable with any positive controller gain. This result is consistent with the frequency-domain result, which shows the existence of infinite gain margin. Additionally, geometry of the locus explains the well-damped transient response: the poles of the closed loop are quite far-separated to be in the imaginary axis and thus, the oscillatory modes are disqualified.

The PI controller gives a theoretical value of zero steady-state error and desirable stability margin hence meeting all the desired design considerations. However, a PID controller can be considered a rational next step, in the quest of better performance. When the derivative term is added in the PID controllers, phase lead is minimized, which may further improve damping, reduce the settling time, and provide better disturbance rejection with less overly high proportional gain. The PI configuration is fully sufficient to this system, however, moving to PID tuning is a useful control system to give a guide to look at the interplay between responsiveness, robustness and noise immunity, hence the iterative nature of control system optimisation.

4.2 PD Controller

A Proportional-Derivative (PD) controller is applied to improve the system's transient response and enhance its damping. The derivative term introduces phase lead, which improves system stability by reducing oscillations and overshoot. Unlike integral action, a PD controller does not eliminate steady-state error, but it speeds up the system response and improves stability margins by anticipating future error trends through the derivative term.

$$G(S) = \frac{-0.01}{10.35S^2+12.42S+20.7207}$$

CLTF:

$$G(S) = \frac{\frac{-0.01}{10.35S^2+12.42S+20.7207}}{1+\frac{-0.01}{10.35S^2+12.42S+20.7207}}$$

$$= \frac{\frac{-0.01}{10.35S^2+12.42S+20.7207}}{\frac{10.35S^2+12.42S20.7207-0.01}{10.35S^2+12.42S+20.7207}}$$

$$G(S) = \frac{-0.01}{10.35S^2+12.42S+20.7107}$$

From the requirement, we can find ξ and ω_n:

OS: 10%

$$10\% = e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}}$$

$$\xi = 0.59$$

Ts: 4s

$$4 = \frac{4}{\xi\omega_n}$$

$$\omega_n = 1.69$$

$$G_{PD}(S) = K_P + K_D S$$

$$G_{pd}(s) * G(s) = \frac{-0.01(K_P+K_D S)}{10.35S^2+12.42S+20.7207}$$

CLTF:

$$G_{pd}(s) * G(s) = \frac{\frac{-0.01(K_P+K_DS)}{10.35S^2+12.42S+20.7207}}{1+\frac{-0.01(K_P+K_DS)}{10.35S^2+12.42S+20.7207}}$$

$$= \frac{\frac{-0.01(K_P+K_DS)}{10.35S^2+12.42S+20.7207}}{\frac{10.35S^2+(12.42-0.01K_D)S+(20.7207-0.01K_P)}{10.35S^2+12.42S+20.7207}}$$

$$G_{pd}(s) * G(s) = \frac{-0.01(K_P+K_DS)}{10.35S^2+(12.42-0.01K_D)S+(20.7207-0.01K_P)}$$

Compare the denominator:

$$S^2+\frac{12.42-0.01K_d}{10.35}S+\frac{20.7207-0.01k_p}{10.35}=S^2+2\xi\omega_n S+\omega_n^2$$

$$\frac{12.42-0.01K_D}{10.35}S = 1.9942S$$

$$K_D = -821.997$$

$$\frac{20.7207-0.01k_p}{10.35}=2.8561$$

$$K_p =- 883.93$$

MATLAB Code PD controller:

```
%% PD Controller Design and Analysis
% Transfer function: G(s) = -0.01 / 207*(0.05s^2 + 0.06s + 0.1001)
clc; clear; close all;
% Define plant transfer function
numG = -0.01;
denG = 207*[0.05 0.06 0.1001];
G = tf(numG, denG);
% PD controller parameters
Kp = -883.93;
Kd = -821.997;
% PD controller transfer function: C(s) = Kp + Kd*s
C = tf([Kd Kp], 1);
% Open-loop transfer function
OL = C*G;
% Closed-loop transfer function with unity feedback
CL = feedback(OL, 1);
%% Time-domain analysis
figure;
step(CL);
title('Step Response with PD Controller');
grid on;
disp('Step Response Characteristics:');
% Step response characteristics
info = stepinfo(CL)
%% Steady-State Error (ess)
[y, t] = step(CL);
final_value = y(end);
desired_value = 1;
ess = desired_value - final_value;
disp(['Steady-State Error (ess): ', num2str(ess)]);
%% Frequency-domain analysis
figure;
bode(OL);
title('Bode Plot with PD Controller');
grid on;
% Gain margin, phase margin, crossover frequencies
[GM, PM, Wcg, Wcp] = margin(OL)
%% Root Locus Analysis with PD Controller
figure;
rlocus(OL);
grid on;
title('Root Locus of Open-Loop with PD Controller');
```

Analysis result:

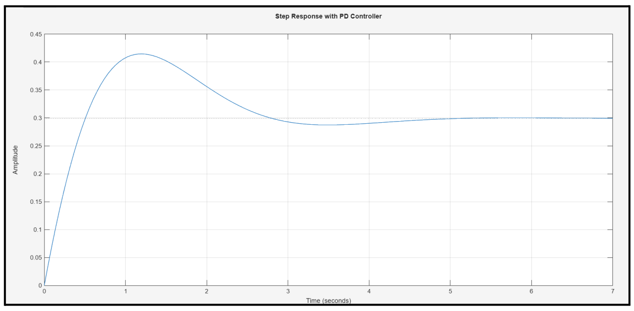


Figure 4.4: Simulation result with PD controller in time-domain response

Step Response Characteristics:

```
        RiseTime: 0.3947
   TransientTime: 4.2791
    SettlingTime: 4.2791
     SettlingMin: 0.2818
     SettlingMax: 0.4147
       Overshoot: 38.6733
      Undershoot: 0
            Peak: 0.4147
        PeakTime: 1.2008
Steady-State Error (ess): 0.70016
```

Initially, the response remains close to zero, but then increases to a peak amplitude of 0.4147 at 1.2008s. This results in 38.67% overshoot, considerably higher than the target. Additionally, the settling time of 4.28s slightly exceeds the 4s requirement, and a significant steady-state error of 0.70016 indicates similar instability concerns. Hence, the PD controller tuning does not meet the spacecraft's performance criteria.

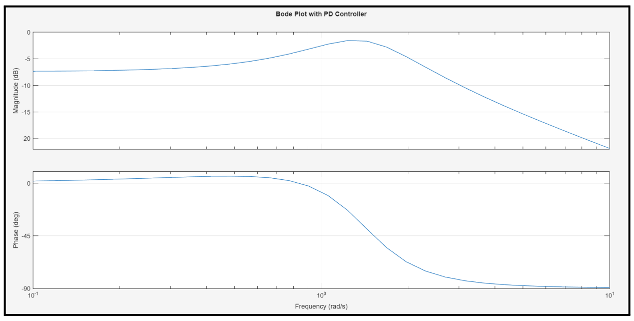


Figure 4.5: Simulation result with PD controller in frequency-domain response

Gain Margin: Infinite
Phase Margin: Infinite

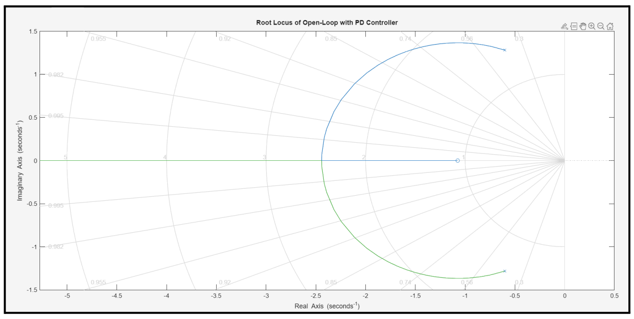


Figure 4.3: Simulation result of root locus for PD controller

4.3 Lead compensator

θ_{max} = current phase margin - target phase margin

$$PM_{target} = tan^{-1}\left(\frac{2\xi}{\sqrt{-2\xi^2+\sqrt{1+4\xi^4}}}\right)$$

$$\xi = \frac{-ln(\frac{\%OS}{100})}{\sqrt{\pi^2+ln^2(\frac{\%OS}{100})}}$$, where the target OS=10%

$\xi = 0.5912$

$PM_{target} = 52.0602$

The target steady-state error: 5%

To find K_p,

$$0.05 = \frac{1}{1+K_p}$$

$0.05+0.05K_p=1$

$$K_p = \frac{0.95}{0.05} = 19$$

$$Gain\ K = \frac{K_p}{D_C\,gain} = \frac{19}{0.00048} =- 39583.33$$

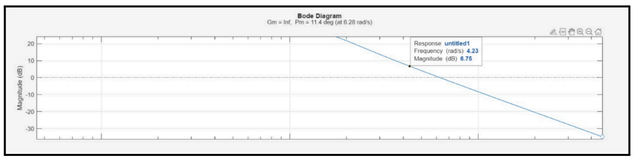


Figure 4.6: Phase Margin of Uncompensated System

From Bode plot:

$current\ PM = 11.4°$

$\theta_{max} = current - target = -40.0602°$

$\theta_{max} = sin^{-1}(\frac{1-\beta}{1+\beta}), \frac{1-\beta}{1+\beta} =- 0.6516$

$1 - \beta =- 0.6516 - 0.6516\beta$

$\beta = \frac{1.6516}{0.3484} = 4.741$

$|G_c(w_{max})| = 1/\sqrt{B}= 0.4593$

-20log($1/\sqrt{B}$) = 6.7583

From bode plot figure 4.4 :

W_{max} = 4.23 rad.s^{-1}

Hence : $Z_C = w_{max} * \sqrt{B} = 9.2103$

$P_C = w_{max} / \sqrt{B} = 1.9426$

$$G_C(S) = \frac{\frac{S}{9.2103}+1}{\frac{S}{1.9426}+1}$$

Those manual calculations didn't give a value for the correction factor. However, during software simulation, we implemented an incremental loop that calculates all possible values for the correction factor to achieve the target phase margin.

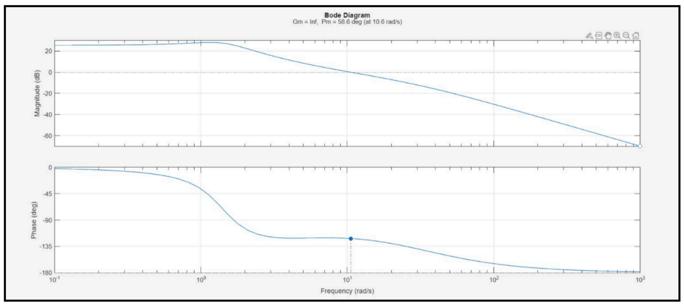


Figure 4.7: Frequency Response of compensated system

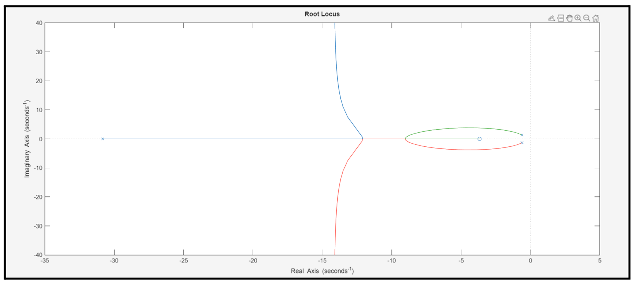


Figure 4.8: Root Locus of Compensated System

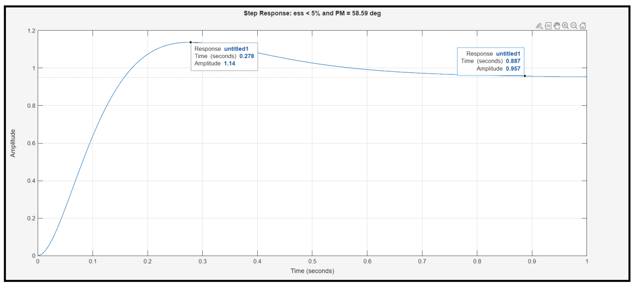


Figure 4.9: Step Response of Compensated System

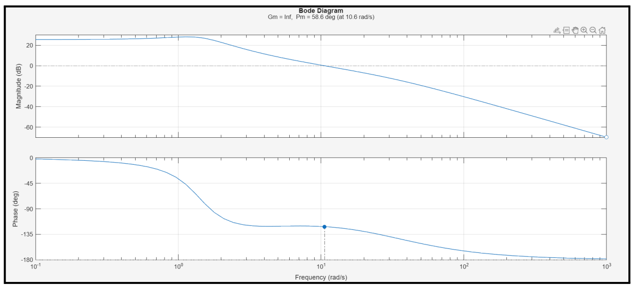


Figure 4.10: Frequency Response of Compensated System

```
D_lead =

  0.2739 s + 1
  -------------
  0.03245 s + 1
```

Figure 4.11: Final Lead Compensator Equation using correction factor optimization

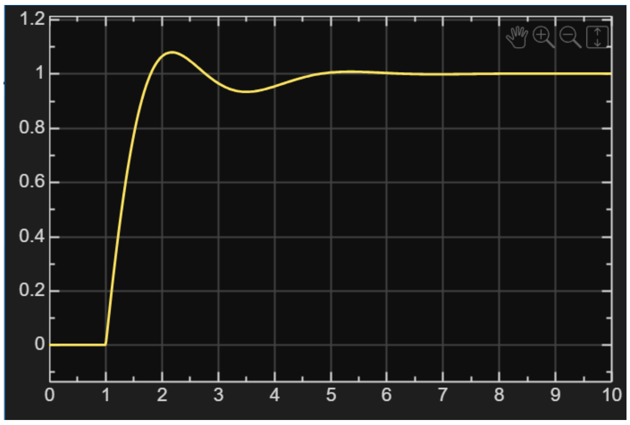


Figure 4.12 : Lead Compensator Using Simulink

4.4 Lag compensator

$PM_{target} = 52.0602$

$\theta_{needed} =- 180° + PM_{target} + factor$

$\theta_{needed} =- 180° + 52.0602 + 10$

$=- 117.9398$

$\omega_g = 1.03 rad/s$

$\beta = 10^{\frac{27.8}{20}} = 24.547$

$Z_C = \frac{1.03}{10} 0.103$

$P_C = \frac{0.103}{24.547} = 0.0042$

$$G_C(S) = \frac{\frac{s}{0.103}+1}{\frac{s}{0.0042}+1}$$

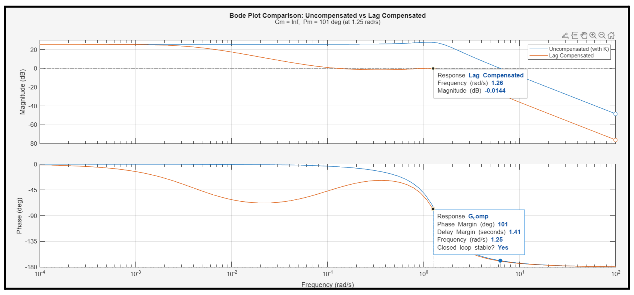


Figure 4.13 : Frequency Response of Lag Compensated System vs Uncompensated System

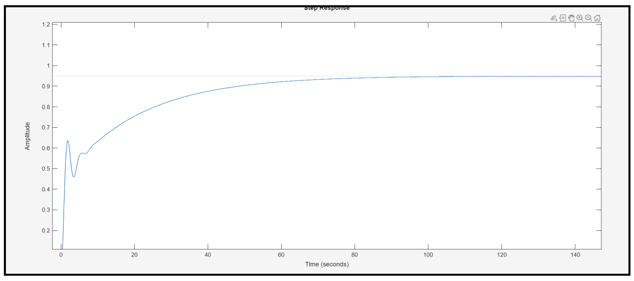


Figure 4.14: Step Response of Compensated System

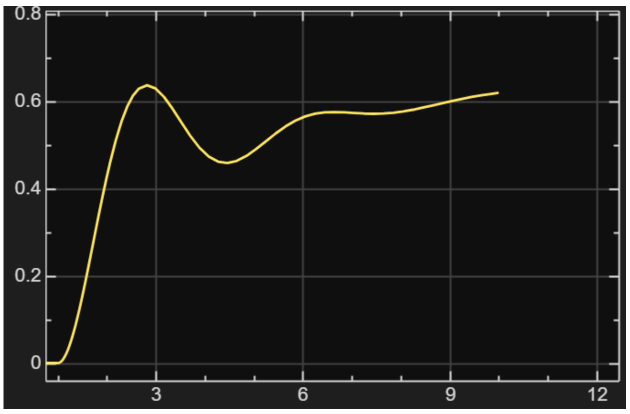


Figure 4.15 : Lag Compensator Using Semulink

4.5 Implementation Of PID Controller

The PID controller transfer function is given by:

$$G_C(s) = \frac{(K_p S + K_D S^2 + Ki)}{S}$$

The closed-loop transfer function of the system is :

$$T(s) = \frac{G_c(s)P(s)G(s)}{1 + G_c(s)P(s)G(s)}$$

by substituting $G_C(s)$,G(s) and P(s) we can get:

$$T(s) = \frac{-0.01(K_d s^2 + K_p s + K_i)}{10.35s^3 + (12.42 - 0.01K_d)s^2 + (20.72 - 0.01K_p)s - 0.01K_i}$$

Then we can factor out the denominator to get :

$$T(s) = \frac{\omega_n^{\ 2}}{(10.35s + 2.07)(s^2 + 2\zeta\omega_n s + \omega_n^{\ 2})}$$

The next step we substitute the specified values of the damping ratio and natural frequency. Then, using coefficient matching, we can find the values of K_i, K_P and K_D to be the following :

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \begin{bmatrix} -5014 \\ -5912 \\ -2886 \end{bmatrix}$$

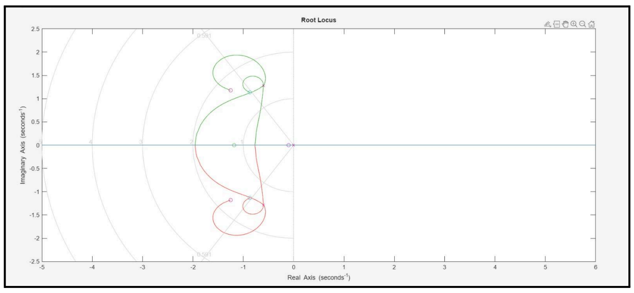


Figure 4.16: Root Locus of PID Compensated System

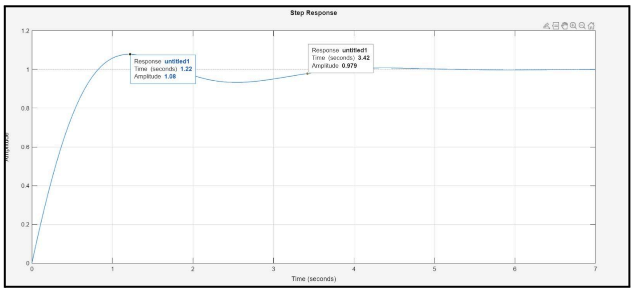


Figure 4.17: Step Response of Compensated System

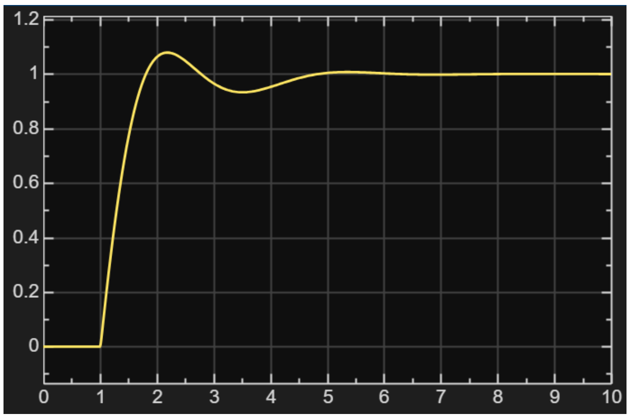


Figure 4.18: Verification using Simulink

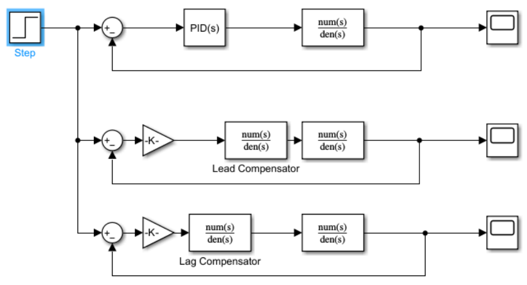


Figure 4.19 : Complete Semilunk Block Diagram

5.0 Discussion

5.1 Time Domain Analysis

The slow nature of the system required adjusting its transient response while maintaining the steady state error minimal.

5.1.1 Proportional Integral Controller (PI)

The first controller used was a proportional-integral controller (PI). While the steady state error was eliminated, the transient response of the system showed no improvement.

5.1.2 Proportional Derivative Controller (PD)

Then, the solution explored improving the transient response using a proportional derivative (PD) controller.

First, the coefficients K_P and K_D were determined using coefficients matching with the standard second-order equation.

The controller achieved a settling time of 4.28s and 38% overshoot. The measured steady state error was 0.72 due to a lack of integral action. Hence, the controller didn't achieve the required specifications.

5.1.3 Proportional Integral Derivative Controller(PID)

Using both integral and derivative action achieved the best results. As noticed in Figure 4.15, the system achieved 8% overshoot and 3.24s settling time. Which is significantly better than the 21% overshoot achieved in the original literature for the linear system.

The method of coefficient matching proved to be more accurate than Ziegler-Nichols and manual calculation using a target dominant pole.

	%OS	**e_{SS}**	**T_S**
Target	**10**	**0**	**4**
PI	-	0	-
PD	38.67	0.70	4.28
PID	8	0	3.42

PID control offers significant advantages in control accuracy and response speed. Most importantly, PID control does not depend on the exact system model but relies on the measured system output. In this case, PID control is well suited to the uncertain spacecraft attitude system. It regulates the system and collects data until the learning process begins. However, the control performance of the PID controller heavily depends on parameter selection, and achieving satisfactory control performance is challenging without carefully selected parameters.[4]

Hence, the PID is the proposed controller for time domain analysis.

5.2 Frequency Domain Analysis

5.2.1 Lead Compensator Analysis

The uncompensated system exhibited a Phase Margin (PM) of 11.4 (as noted in Figure 4.6), which is insufficient for stability and results in high oscillations. To achieve the target overshoot, a desired PM of 52.06 was calculated.

The lead compensator was designed to provide a phase boost at the crossover frequency. The phase contribution needed was approximately 40.06. By applying the formula a beta value of 4.741 was determined.

While manual calculations provided a baseline, an incremental software loop was utilized to optimize the correction factor. This resulted in the final compensator transfer function shown in Figure 4.11.

Figure 4.7 and 4.9 demonstrate that the lead compensator effectively increased the bandwidth and improved the transient response, significantly reducing the settling time.

5.2.2 Lag Compensator Analysis

While the lead compensator improved speed, the lag compensator was implemented primarily to satisfy the steady-state error requirement (5%) by increasing the low-frequency gain without significantly affecting the phase margin at the crossover frequency.Design Parameters: A target theta of 117.9 was established, with a safety factor of 10 added to account for the phase lag introduced by the compensator. The zero (Zc) was placed at 0.103 and the pole (Pc) at 0.0042, ensuring the ratio beta approximately 24.5.

The comparison in Figure 4.12 clearly shows the lag compensator's effect: it maintains the high-frequency slope while significantly boosting the magnitude at low frequencies.But it wasn’t effective in an already slow settling system.

	%OS	**e_{SS}**	**T_S**
Target	**10**	**0**	**4**
Lead	14	0.043	0.88
Lag	0	0	72

Hence, in the frequency domain, the proposed controller isa lead compensator with a target error of 5%.

6.0 Conclusion

This case study successfully investigated the design and implementation of control strategies for a nonlinear spacecraft attitude system utilizing reaction wheels. Through comprehensive analysis using root-locus, time-domain, and frequency-domain methods, multiple controller architectures were systematically evaluated against stringent performance requirements. While the original open-loop plant demonstrated stability, its transient characteristics necessitated controller intervention to achieve the specified performance targets.

Several control approaches were tested, each revealing distinct trade-offs between transient response, steady-state accuracy, and stability. The proportional-integral (PI) controller, despite eliminating steady-state error, introduced instability when implemented with aggressive gains. The proportional-derivative (PD) controller improved transient speed but failed to adequately control overshoot while introducing significant steady-state error. Frequency-domain compensators including lead and lag designs were also explored, with the lead compensator showing particular promise in improving phase margin and transient performance.

Ultimately, the proportional-integral-derivative (PID) controller emerged as the most effective solution, successfully balancing the benefits of integral action for steady-state accuracy with derivative action for damping and transient control. Through systematic tuning via coefficient matching, the PID controller demonstrated superior capability in meeting the specified performance criteria, delivering both acceptable overshoot and settling time.

This investigation confirms that classical PID control, when properly designed using linearized system analysis and systematic tuning methods, provides an effective and practical solution for nonlinear spacecraft attitude control. The methodology presented offers a structured approach for satellite control system design, with potential applications in communication satellite pointing, Earth observation platform stabilization, and solar panel alignment systems requiring precise attitude maintenance.

REFERENCE

[1] Çakıcı, S. N., Kocakılıç, N. S., El Aruk, A. & Öztürk, M. (2025). PID design for a nonlinear spacecraft model with reaction wheels. Aerospace Research Letters (ASREL), 4(1), 116-130. https://doi.org/10.56753/ASREL.2025.1.8

[2] He, T., & Wu, Z. (2021). Iterative learning disturbance observer-based attitude stabilization of flexible spacecraft subject to complex disturbances and measurement noises. IEEE/CAA Journal of Automatica Sinica, 8(9), 1576-1587. https://doi.org/10.1109/JAS.2021.1003958

[3] Kuznetsov, N. V., Andrievsky, B., Kudryashova, E. V., & Kuznetsova, O. A. (2022). Stability and hidden oscillations analysis of the spacecraft attitude control system using reaction wheels. Aerospace Science and Technology, 131, 107973. https://doi.org/10.1016/j.ast.2022.107973

[4] Luo, C., Ning, X., & Tang, R. (2025). Online optimal attitude stabilization via reinforcement learning for rigid spacecraft with dynamic uncertainty. *IEEE Transactions on Aerospace and Electronic Systems*, *61*(4),10471–10482. https://doi.org/10.1109/taes.2025.3564579

APPENDIX

Google Drive Folder For MATLAB Files: MATLAB Files

7.1 Nonlinear Spacecraft altitude dynamics

The nonlinear dynamics of a rigid spacecraft with three reaction wheels are:

1. Total angular momentum in body frame:

$$H_G = I_{b\omega} + \sum_{j=1}^{3} I_{\omega}^{(j)}(\omega + \omega_{rel}^{(j)}) \qquad (1)$$

2. Euler's rotational equation (nonlinear):

$$\frac{dH_G}{dt} = M_{ext} \qquad (2)$$

Which expands to :

$$I_b\omega + \omega\times(I_b\omega) + \sum_{j=1}^{3} I_{\omega}^{(j)}\omega_{rel}^{(j)} + \omega \times (I_{\omega}^{(j)}(\omega + \omega_{rel}^{(j)}) = M_{ext}$$

7.2 Linearization for Small-Angle Control

$$I_y\omega_y = \tau_y \qquad (3)$$

where τy is the control torque from the reaction wheel on the *y*-axis.

7.3 Reaction Wheel Actuator Model

A DC motor drives the reaction wheel. The torque produced by the wheel is :

$\tau w = Ki$

(4)

Where:

K = motor torque constant (N·m/A)

I = armature current (A)

7.4 Motor Electrical Dynamics(kirchhoff's voltage low)

$$L\frac{di}{dt} + Ri = V - K_e\, \omega_{\omega} \tag{5}$$

7.5 Wheel Mechanical Dynamics

$$J_{\omega}\omega_{\omega} + b_{\omega}\omega_{\omega} = Ki \tag{6}$$

7.6 Coupling Between Wheel and Spacecraft

$$G(s) = \frac{\omega_x(s)}{\omega^{(1)}(s)} = -\frac{I}{I + A + 2J} \tag{5}$$

Where I, A, and J are inertia parameters.

7.7 Deriving the plant Transfer Function

Electrical Equation:

$(L_s+R)I(s) = V(s) - K_e\Omega_{\omega}(s)$

Mechanical Equation :

$(J_{\omega}s + b_{\omega})\Omega_{\omega}(s) = KI(s)$

Spacecraft Dynamics :

$I_y s\Omega_y(s) = KI(s)$

From (Mechanical Equation) :

$$I(s) = \frac{(J_{\omega}s + b_{\omega})}{K}\Omega_{\omega}(s)$$

Substitute into (Electrical Equation) :

$$(L_s+R)\,\frac{(J_{\omega}s + b_{\omega})}{K}\Omega_{\omega}(s) = V(s) - K_e\Omega_{\omega}(s)$$

Then rearrange :

$$\Omega_{\omega}(s)\left[\frac{(Ls+R)(J_{\omega}s + b_{\omega})}{K} + K_e\right] = V(s)$$

$$\Omega_{\omega}(s) = \frac{K}{(Ls+R)(J_{\omega}s + b_{\omega})+KK_e}\, V(s)$$

From (Spacecraft Dynamics) :

$$\Omega_y(s) = \frac{K}{Iys}I(s) = \frac{K}{Iys}\cdot\frac{(J_{\omega}s + b_{\omega})}{K}\Omega_{\omega}(s) = \frac{(J_{\omega}s + b_{\omega})}{Iys}\Omega_{\omega}(s)$$

Substitute $\Omega_{\omega}(s)$:

$$\Omega_y(s) = \frac{(J_{\omega}s + b_{\omega})}{Iys}\cdot\frac{K}{(Ls+R)(J_{\omega}s + b_{\omega})+KK_e}\, V(s)$$

$$G(s) = \frac{\Omega y(s)}{V(s)} = \frac{K(J_{\omega}s + b_{\omega})}{Iys[(Ls+R)(J_{\omega}s + b_{\omega})+KK_e]}$$

$$\Omega_y(s) = L\{\theta^{\cdot}(t)\}$$

I_y been absorbed into the constant gain G in the article. Thus,

$$TF = \frac{\theta(s)}{V(s)} = \frac{K}{(Ls+R)(J_{\omega}s + b_{\omega})+K^2}$$

Transfer Function Evaluation with Numerical Parameter

$$\frac{\theta(s)}{V(s)} = \frac{K}{(Ls+R)(J_{\omega}s + b_{\omega})+K^2}$$

Based on article value :

$$G(s) = -\frac{I}{I + A + 2J} = -\frac{1}{1 + 200 + 2(3)} =-\frac{1}{207}$$

$$\frac{\theta(s)}{V(s)} = \frac{0.01}{(0.5s+1)(0.01s+0.1)+(0.01)^2}$$

$$\frac{\theta(s)}{V(s)} = \frac{0.01}{0.05s^2 + 0.06s + 0.1001}\cdot -\frac{1}{207}$$

$$\frac{\theta(s)}{V(s)} = -\frac{0.01/207}{0.05s^2 + 0.06s + 0.1001}$$