ELL225 Control Engineering

T Pratham - 2020EE10559
Geetigya Joshi - 2020EE10493
Sampan Manna - 2020EE10547
Afreen Haider - 2020EE10463
Kanad Pandey - 2020EE30250

February 11, 2023

1 Introduction

Bicycles have immense utility in many spheres of modern life, be it transportation, exercise, or relaxation. Because of their interesting dynamic behaviour, bicycles have intrigued scientists since the 19^{th} century. Under static conditions, bicycles are unstable, just like an inverted pendulum. However, under certain conditions of forward motion, the bicycle can become stable, the same way an inverted pendulum can be stabilized by moving the horizontal contact. This was proven in the model proposed by Whipple (1899) and Carvallo (1900), where the equations of motion were linearized around the upright vertical equilibrium point were derived. Due to the development of technology such as computers software for numerical simulations (such as MATLAB), the bicycle model has received much attention in recent years too.

2 Model Description

We consider the bicycle to be made of 4 rigid parts: two wheels, one frame, and one front fork with handlebars. The rider is modelled as a point mass which can move laterally with respect to the frame. The rider also occupies the major portion of the mass of the entire system, and its main job is to apply the input torque on the handles. To avoid complicating the model, we ignore the effects of other parts such as brakes, chains and pedals.

In this simplified model, we assume three degrees of freedom: the roll angle $\phi(t)$, the steering angle $\delta(t)$, and the speed of the bicycle $v(t)$.

3 Salient Features of the Carvallo Whipple Model

- We break down the motion of the bicycle into two independent types of motions: longitudinal and lateral, so that their dynamics can be treated independently.
- The longitudinal motion concerns the propulsion, acceleration and braking of the bicycle with the rolling resistance and the aerodynamic drag forces acting on the bicycle.
- A single degree of freedom is sufficient to describe the forward motion.
- If the lateral tyre slip is also assumed to be zero, the lateral motion can be described by **two degrees of freedom**, the roll angle of the rear frame, denoted by ϕ, and the steering angle, denoted by δ.

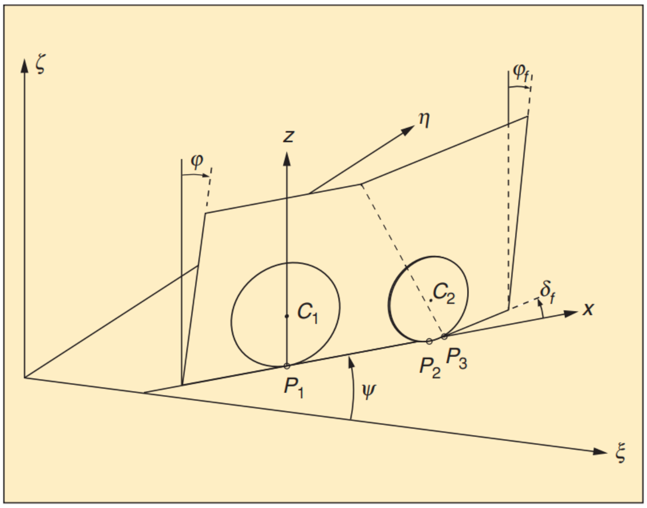

Figure 2. *Coordinate systems. The orthogonal system $\xi\eta\zeta$ is fixed to inertial space, and the ζ-axis is vertical. The orthogonal system xyz has its origin at the contact point of the rear wheel with the $\xi\eta$ plane. The x axis passes through the points P_1 and P_3, while the z axis is vertical and passes through P_1.*

- The linearised equations for the lateral motion if the bicycle is moving at a constant forward velocity v constitute the Carvallo–Whipple model, which have the structure:

$$\mathbf{M}\ddot{\mathbf{q}} + v\mathbf{C_1}\dot{\mathbf{q}} + (v^2\mathbf{K_2} + g\mathbf{K_0})\mathbf{q} = \mathbf{f} \tag{1}$$

  where,

  - $\mathbf{q} = [\phi, \delta]^T$ is the vector of degrees of freedom,
  - $\mathbf{f} = [T_\phi, T_\delta]^T$ is the vector of applied generalised forces,
  - $v\mathbf{C_1}$ is the non-symmetric velocity sensitivity matrix that is linear in the velocity,
  - $v^2\mathbf{K_2} + g\mathbf{K_0}$ is the non-symmetric stiffness matrix that consists of a part that is quadratic in the velocity and a symmetric part that linearly depends on the acceleration of gravity g.

4 State Space model

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$
$$y = \mathbf{C}\mathbf{x} + \mathbf{D}u$$

where,

- $\mathbf{x} = [\phi, \delta, \dot{\phi}, \dot{\delta}]^T$
- $u = T_\delta$ (steering torque is the input)
- y = ϕ (roll angle is the output)

In order to obtain the parameter matrices M, C, K_0 and K_1 involved in the Carvallo-Whipple model, we will be using certain standard parameters that characterize a standard bicycle, including parameters describing wheel base ($b = 1.00m$), trail ($c = 0.08m$), head angle ($\lambda = 70^o$) wheel radii ($R_{rw} = R_{fw} = 0.35m$) and the parameters describing the mass and moment of inertia.

	Rear Frame	Front Frame	Rear Wheel	Front Wheel
Mass m $[kg]$	12	2	1.5	1.5
Center of Mass				
x$[m]$	0.439	0.866	0	b
z$[m]$	0.579	0.676	R_{rw}	R_{fw}
Inertia Tensor				
J_{xx} $[\frac{kg}{m^2}]$	0.476	0.08	0.07	0.07
J_{xz} $[\frac{kg}{m^2}]$	-0.274	0.02	0	0
J_{yy} $[\frac{kg}{m^2}]$	1.033	0.07	0.14	0.14
J_{zz} $[\frac{kg}{m^2}]$	0.527	0.02	J_{xx}	J_{xx}

These parameters give us the values of the matrices as follows:

$\mathbf{M} = \begin{bmatrix} 6.00 & -0.472 \\ -0.472 & 0.152 \end{bmatrix}$ $\mathbf{C} = \begin{bmatrix} 0 & -5.84 \\ 0.436 & 0.666 \end{bmatrix}$

$\mathbf{K_0} = \begin{bmatrix} -91.72 & 7.51 \\ 7.51 & -2.57 \end{bmatrix}$ $\mathbf{K_1} = \begin{bmatrix} 0 & -9.54 \\ 0 & 0.848 \end{bmatrix}$

We then proceed to evaluate the state space parameters A,B,C, and D using the values of M, C, K_0 and K_1 as shown in the MATLAB code snippet.

5 Transfer Function

Transfer function of a state-space model, taking zero initial conditions, is given by

$$\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$$

Using the above relation, we obtain the following transfer functions for each of the cases

5.1 Case 1 ($v = 0m/s$)

$$T = \frac{0.6848s^2 - 10.9}{s^4 - 2.665 \times 10^{-15}s^3 - 32.31s^2 + 4.263 \times 10^{-14}s + 260.2}$$

Zeroes	Poles	Eigenvalues
3.9889	4.1350	-4.1350
-3.9889	3.9009	-3.9009
	-4.1350	3.9009
	-3.9009	4.1350

5.2 Case 2 ($v = 1m/s$)

$$T = \frac{0.6848s^2 + 8.473s + 2.945}{s^4 + 2.097s^3 - 27.77s^2 - 23.71s + 251.3}$$

Zeroes	Poles	Eigenvalues
-12.0149	-4.4429 + 0.4761j	3.3944 + 1.0314j
-0.3580	-4.4429 - 0.4761j	3.3944 - 1.0314j
	3.3944 + 1.0314j	-4.4429 + 0.4761j
	3.3944 - 1.0314j	-4.4429 - 0.4761j

5.3 Case 3 ($v = 6m/s$)

$$T = \frac{0.6848s^2 + 50.84s + 487.4}{s^4 + 12.58s^3 + 131.2s^2 + 1125s - 60.17}$$

Zeroes	Poles	Eigenvalues
-62.9271	-10.4100 + 0.0000j	-1.1127 +10.3689j
-11.3102	-1.1127 +10.3689j	-1.1127 -10.3689j
	-1.1127 -10.3689j	-10.4100 + 0.0000j
	0.0532 + 0.0000j	0.0532 + 0.0000j

6 Preliminary Analysis (Using Routh Hurwitz)

For k=1

v=0

$$\begin{array}{c|ccc} s^4 & 1 & 31.63 & 249.3 \\ s^3 & -2.665 & 42.33 & \\ s^2 & 47.51 & 249.3 & \\ s^1 & 56.314 & & \\ s^0 & 249.3 & & \end{array}$$

Since there are 2 sign changes therefore 2 roots in the right half.
Hence system is unstable.

v=1

$$\begin{array}{c|ccc} s^4 & 1 & -27.0852 & 254.245 \\ s^3 & 2.097 & -15.237 & \\ s^2 & -19.82 & 254.245 & \\ s^1 & 11.663 & & \\ s^0 & 254.245 & & \end{array}$$

Since there are 2 sign changes therefore 2 roots in the right half.
Hence system is unstable.

v=6

$$\begin{array}{c|ccc} s^4 & 1 & 131.8848 & 427.23 \\ s^3 & 12.58 & 1175.84 & \\ s^2 & 38.416 & 427.23 & \\ s^1 & 1035.936 & & \\ s^0 & 427.23 & & \end{array}$$

Since there are 0 sign changes therefore 0 roots in the right half.
Hence system is stable.

Using a proportional controller

v=0

$$\begin{array}{c|ccc} s^4 & 1 & \text{0.6848k-32.31} & \text{260.2-10.9k} \\ s^3 & -26.65 & \text{426.3-3.041k} & \\ s^2 & \text{-16.1+0.5707k} & \text{260.2-10.9k} & \\ s^1 & \frac{70.9+292.75k-1.736k^2}{0.5707k-16.1} & & \\ s^0 & \text{260.2-10.9k} & & \end{array}$$

Since there is one sign change from row 1 to 2 therefore at least one root in right half.
System is unstable.

v=1

s^4	1	0.6848k-27.77	251.3+2.945k
s^3	2.097	8.473k-27.31	
s^2	-16.46-3.35k	251.3+2.945k	
s^1	$\frac{28.38k^2+66.213k+136.7}{3.35k+16.46}$		
s^0	251.3+2.945k		

We can have system stable for only a negative range of k (-85.33 to -4.9134) which is not possible.
System is unstable.
Hence a proportional controller is not sufficient to stabilise the system.

v=6

For k=1 we already have a stable system.

7 Time response for the system

7.1 For velocity = 0

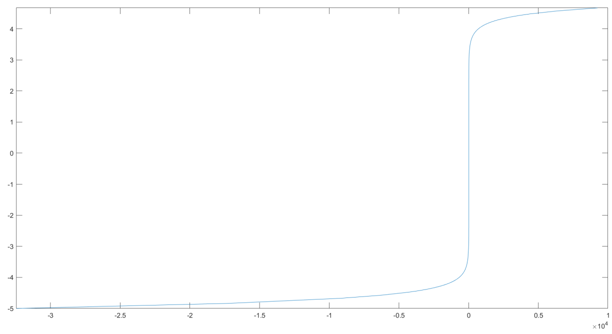

Figure 1: Zero Input response v=0

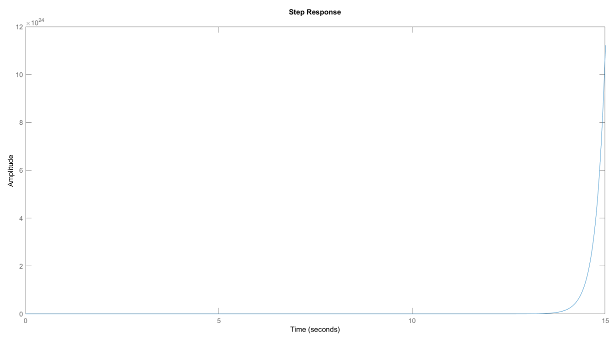

Figure 2: Step Input Response v=0

7.2 For velocity = 1

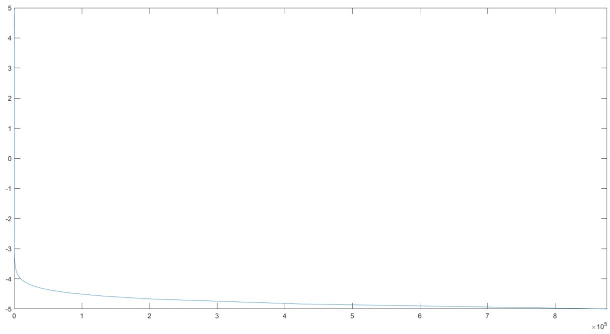

Figure 3: Zero Input Response v=1

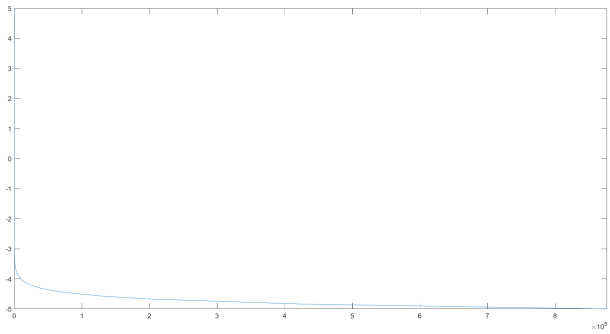

Figure 4: Step Input Response v=1

7.3 For velocity = 6

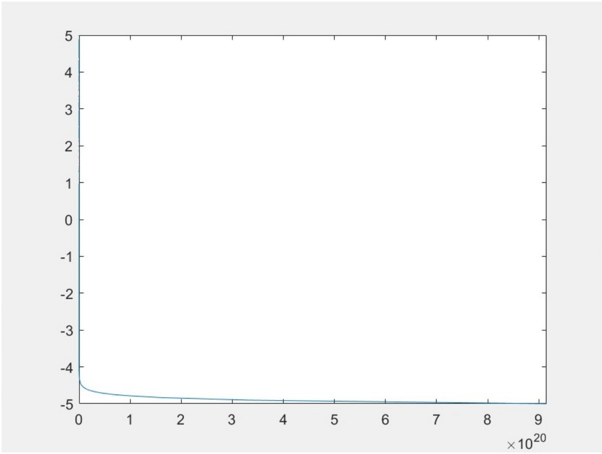

Figure 5: Zero input response v=6

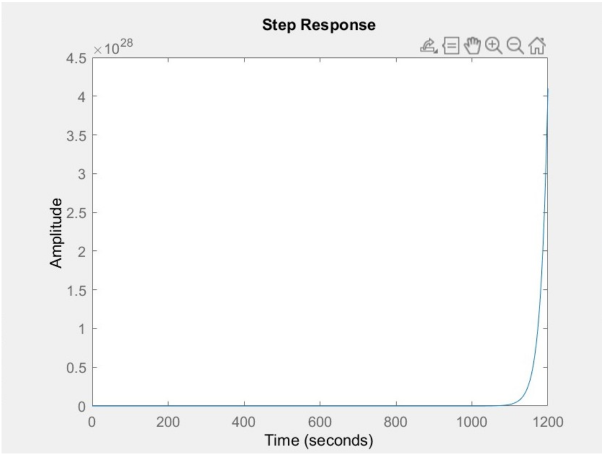

Figure 6: Step Input Response v=6

8 Uncompensated system: Nyquist Plots, Bode Plots and Root Locus

The Nyquist plot is calculated for the open loop transfer function $G(s)$ of the system.
Bode plots are done on the closed loop transfer function $T(s)$ and they are plotted against $log(w)$
Routh Hurwitz analysis is done for closed loop transfer function while Root locus is plotted for the open loop system.

8.1 For velocity = 0 ms^{-1}

Nyquist plot is given by From the plot we obtain that the system is unstable for v=0.

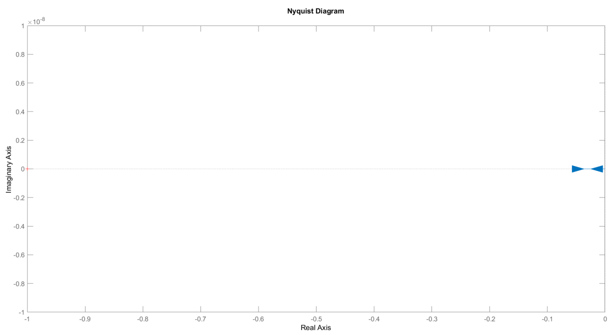

Figure 7: Nyquist plot for v=0

Bode plots are given as

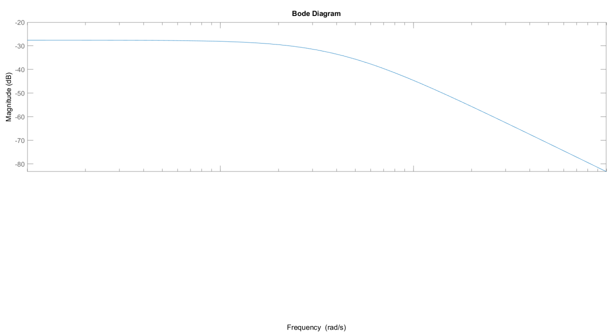

Figure 8: Bode plots for v=0

The root locus for the uncompensated system is

8.2 For velocity = 1 ms^{-1}

Nyquist plot is given by the following figure. The Nyquist plot shows the system to be unstable.

Bode plots are given as
The root locus for the uncompensated system is

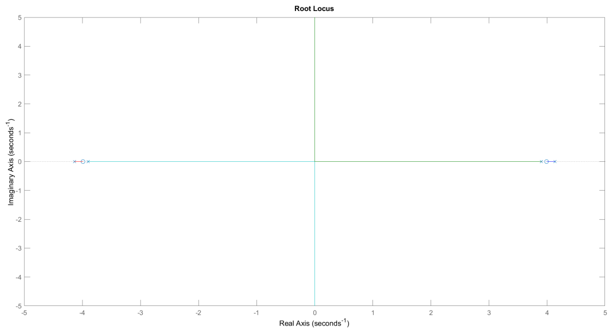

Figure 9: Root Locus for v=0

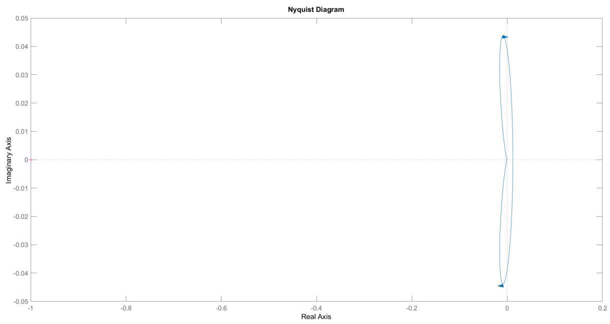

Figure 10: Nyquist plot for v=1

8.3 For velocity = 6 ms^{-1}

Nyquist plot is given by the following figure.
From the Nyquist plot, we obtain -1 clockwise rotations about -1+j0 and 1 open loop pole in the right half of the complex plane. This adds up to 0 closed loop poles for the system hence it is stable.

Bode plots are given as
The root locus for the uncompensated system is

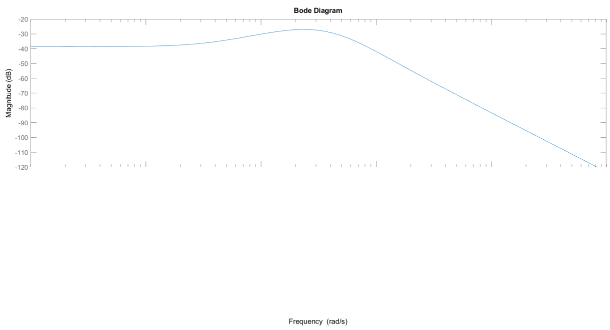

Figure 11: Bode plots for v=1

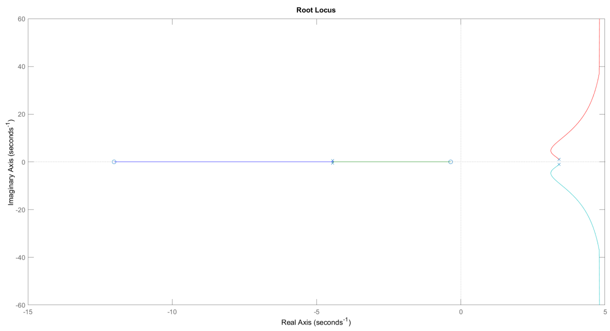

Figure 12: Root Locus for v=1

9 Compensator Design

9.1 For velocity = 0 ms^{-1}

From the Nyquist plot for velocity = 0, we obtain that the closed loop system is unstable.

9.2 For velocity = 1 ms^{-1}

From the Nyquist plot for velocity = $1ms^{-1}$, we obtain that the closed loop system is unstable. In order to stabilize the system, we use a PID controller, whilst meeting the required design specifications.

- damping ratio $\zeta = 0.6$
- settling time $\tau = 0.152s$
- steady state error to be eliminated

For this PID controller, we add a zero at $x = -33.6$ to the modified root locus to achieve transient specifications. We fix damping ratio at 0.6 and keep settling time free to change. In order to eliminate steady state error, we add an integrator (a pole at $x = 0$ and a zero very close to the origin at $x = -10^{-4}$. This modifies the root locus to form a stable closed system. Corresponding plots for the stable system are shown below.

9.3 For velocity = 6 ms^{-1}

From the Nyquist plot for velocity = $6ms^{-1}$, the closed loop system is seen to be stable. However the step response for the system settles to 1.14. Therefore we design a PI controller to control the step response output for this velocity. The required design specifications for this are

- damping ratio $\zeta = 0.6$

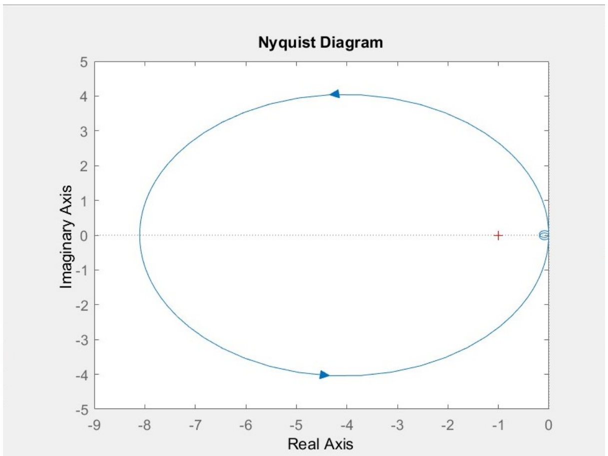

Figure 13: Nyquist plot for v=6

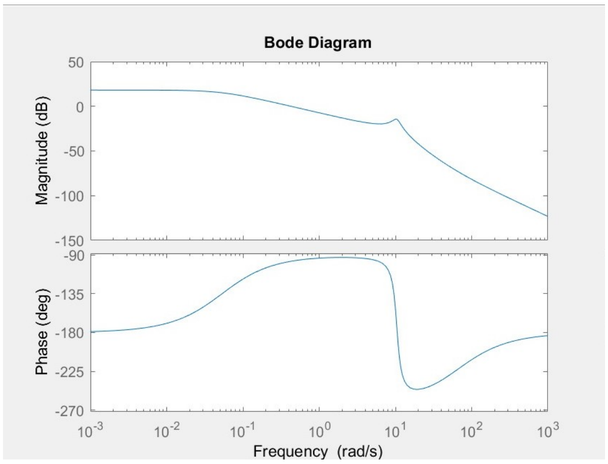

Figure 14: Bode plots for v=6

- settling time $\tau = 1s$
- steady state error to be eliminated

For this PI controller we add a pole at origin and a zero very close to the origin. This stabilizes the step response for the system as shown in the plots below.

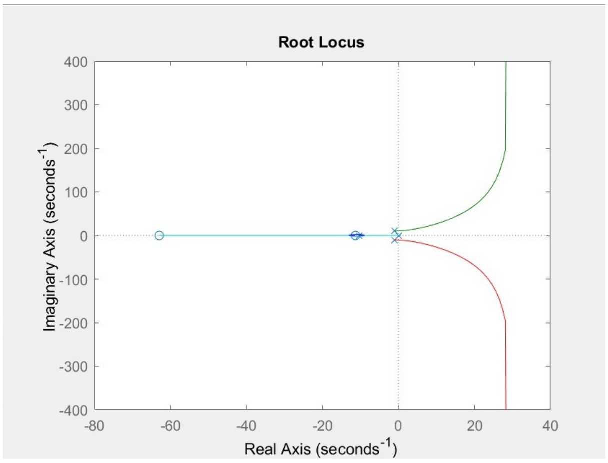

Figure 15: Root Locus for v=6

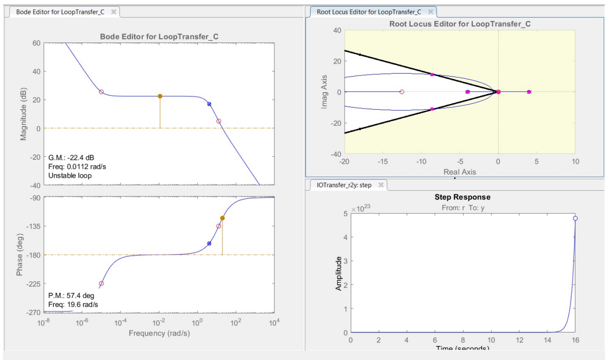

Figure 16: Bode Plots and Root Locus after designing a compensator, system still unstable

10 MATLAB script

```
prompt1='velocity ';
syms s v t;
v=input(prompt1);

M = [
      6, -0.472;
      -0.472, 0.152;
];
C = [
      0, -5.84;
      0.436, 0.666;
];
K0 = [
      -91.72, 7.51;
      7.51, -2.57;
];
K2 = [
      0, -9.54;
      0, 0.848;
];

M_inv = inv(M);

A = [
   0, 0, 1, 0;
```

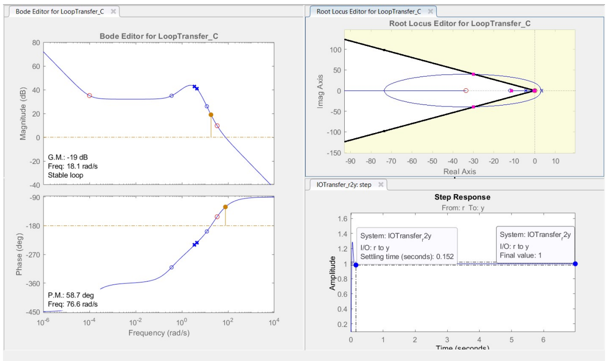

Figure 17: Bode Plot and Root Locus for v=1 after applying the designed compensator

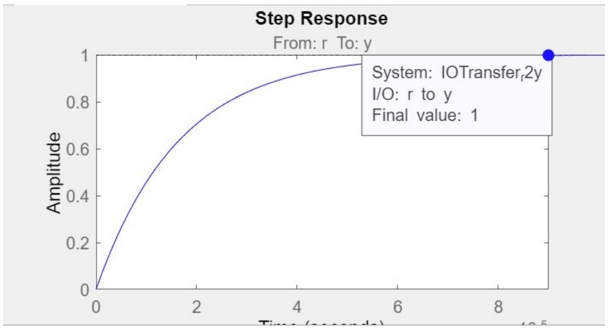

Figure 18: Stable Step Response

```
  0, 0 ,0, 1;
  -1*M_inv*(K0+v*v*K2), -v*M_inv*C
];

B = [
     0;
     0;
     M_inv(1,2);
     M_inv(2,2)
];

C = [1, 0, 0, 0];
D = 0;

[V,d]=eig(A)

I=eye(4);
tfun=C*(inv((s*I-A)))*B+D;
display(vpa(simplify(tfun)));
[b,a]=ss2tf(A,B,C,D)          %state space to tranfer func
[z,p] = tf2zp(b,a)             %transfer func to poles zeroes
sys = tf((b),(a))
%tfun and sys are same, one calculated directly , one manually%


figure('Name', 'Step Response');
step(sys);        %plotting the step response%

figure('Name', 'Nyquist Plot');
nyquist(sys);

figure('Name', 'Root Locus');
```

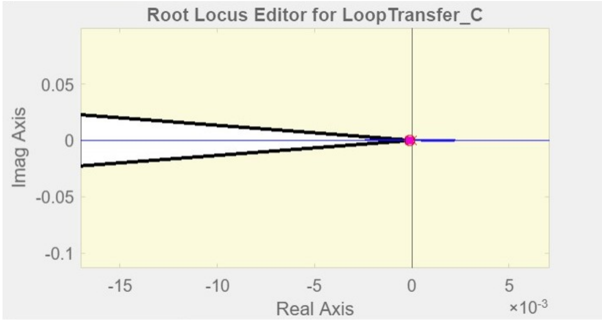

Figure 19: Stable Root Locus

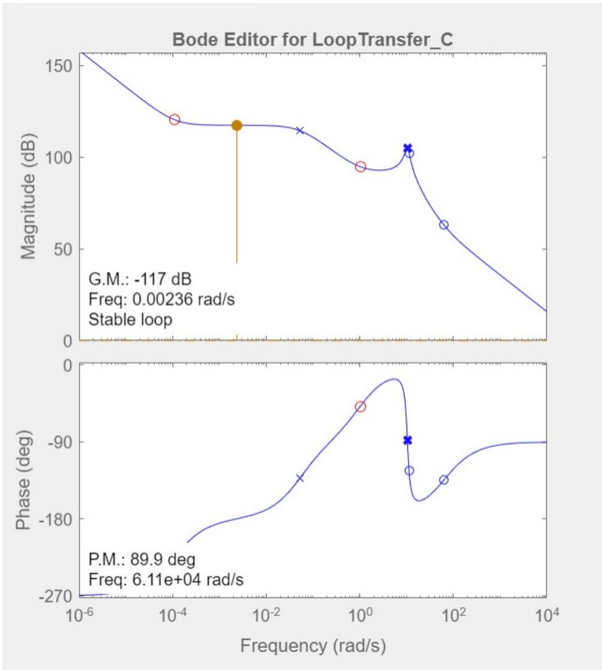

Figure 20: Stable Bode Plots

```
rlocus(sys);

figure('Name', 'Bode Plot');
bode((sys))

x_0 = [0;0;0.001;0];
x(t)=V*expm(d*t)/V*x_0; %trying to plot the zero input response, cant find a command to do it directly%
y(t)=C*x(t);
figure('Name', 'Zero Input Response');
fplot(y(t),t)
```

Listing 1: Script

```
syms v
A=[0 0 1 0;
   0 0 0 1;
   13.67 0.225-1.319*v*v -0.164*v -0.552*v
   4.857 10.81-1.125*v*v 3.621*v -2.388*v
]
e=eig(A)
z=simplify(e)
```

```
%v=linspace(0, 10)
k=0;
fplot (k,v);
for i=1:length(e)
 y = real((e(i)));
 fplot(y,[0,10])
 grid on
 xlabel('velocity (m/s)')
 ylabel('real part of eigen value')
 hold on
end
```

Listing 2: Script

11 Contribution

- Geetigya Joshi: State Space Modelling, evaluation of the transfer function, and corresponding zeroes, poles and eigenvalues, time response
- Kanad Pandey: Evaluation of zeroes, poles, and eigenvalues, and documentation, and type-setting.
- T. Pratham: PID Compensator design, time response
- Afreen Haider: PID Compensator design, Nyquist Plot, Bode Plot
- Sampan Manna: Preliminary Analysis using Routh Tables, Mechanical model description