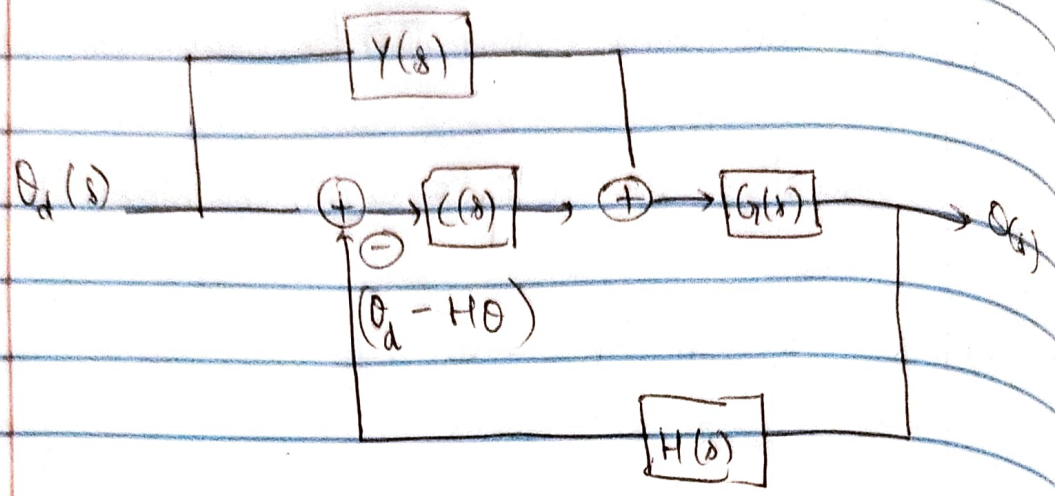
[1]



$$\therefore \left[(\theta_d - H(s)\theta)\cdot C(s) + \theta_d(s)\cdot Y(s)\right]\cdot G(s) = \theta(s)$$

$$\Rightarrow \left[\theta_d(s)\cdot C(s) - \theta C(s)H(s) + \theta_d \cdot Y(s)\right] G(s) = \theta(s)$$

$$\Rightarrow \theta(s)\left[1 + CHG\right] = \theta_d(C + Y)G$$

$$\therefore \frac{\theta(s)}{\theta_d(s)} = \frac{C(s)\cdot G(s)}{1 + C(s)H(s)G(s)} + \frac{Y(s)\,G(s)}{1 + C(s)H(s)G(s)}$$

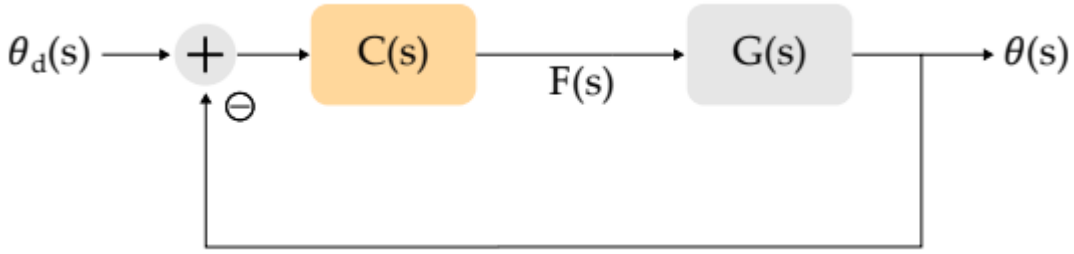
or

$$\boxed{\frac{\theta(s)}{\theta_d(s)} = \frac{CG}{1+CHG} + \frac{YG}{1+CHG}}$$

2 Stability

Imagine that you purchased three separate 1-DoF robots. Each system comes with its own links, joint, motor, and motor controller, and some of the systems are acting in strange ways. Your job is to design controllers that will result in closed-loop stability.

Throughout this problem assume that you can directly measure θ. Use the block diagram above as a reference when determining closed-loop stability.




```
syms C G theta_d theta real

eqn = G*(C*(theta_d - theta)) == theta;
eqn = solve(eqn, theta)/theta_d
```

```
eqn =
```

$$\frac{C\,G}{C\,G+1}$$

```
[~,denom] = numden(eqn)
```

```
denom = C G + 1
```

2.1 (5 points)

The first 1-DoF robot is a mass-damper:

$$f(t) = m\ddot{\theta} - b\dot{\theta} \tag{2}$$

Design a controller that results in closed-loop stability.

```
syms theta m b s F real

laplacian = m*s^2 - b*s
```

```
laplacian = m s^2 - b s
```

```
g = 1/laplacian
```

```
g =
```

$$-\frac{1}{b\,s - m\,s^2}$$

```
c = 7*b*s + 4 % Choosing a controller
```

```
c =
```

$$7\,b\,s + 4$$

```
eqn = subs(denom, [C,G], [c,g]) == 0
```

```
eqn =
```

$$1 - \frac{7\,b\,s + 4}{b\,s - m\,s^2} = 0$$

```
eqn = subs(eqn, [m,b], [1,1])
```

```
eqn =
```

$$1 - \frac{7\,s + 4}{s - s^2} = 0$$

```
verification = round(vpasolve(eqn, s), 4) % Both are real negative values
```

```
verification =
```

$$\begin{pmatrix} -5.2361 \\ -0.7639 \end{pmatrix}$$

2.2 (10 points)

The second 1-DoF robot has plant dynamics:

$$G(s) = \frac{1}{s(s-3)(s-5)} \tag{3}$$

Design a controller that results in closed-loop stability.

```
laplacian = (s*(s-3)*(s-5))
```

```
laplacian =
```

$$s\,(s-3)\,(s-5)$$

```
g = 1/laplacian
```

```
g =
```

$$\frac{1}{s\,(s-3)\,(s-5)}$$

```
c = 20*s^2 + 19 % Choosing a controller
```

```
c =
```

$$20\,s^2 + 19$$

```
eqn = subs(denom, [C,G], [c,g]) == 0
```

```
eqn =
```

$$\frac{20\,s^2+19}{s\,(s-3)\,(s-5)}+1=0$$

```
verification = round(vpasolve(eqn, s), 4) % Both are negative values that lie on
the left side of the plane which is correct in controls
```

```
verification =
```

$$\begin{pmatrix} -10.7712 \\ -0.6144+1.1775\,\mathrm{i} \\ -0.6144-1.1775\,\mathrm{i} \end{pmatrix}$$

2.3 (5 points)

The third 1-DoF robot is a mass-spring-damper:

$$f(t)=5\ddot{\theta}+2\dot{\theta}-15\theta \tag{4}$$

Design a controller that places both poles at $s=-5$. **Aside –** When both poles of a mass-spring-damper are equal negative real numbers, the system is *critically damped*.

```
laplacian = 5*s^2+ 2*s - 15
```

```
laplacian =
```

$5\,s^2+2\,s-15$

```
g = 1/laplacian
```

```
g =
```

$$\frac{1}{5\,s^2+2\,s-15}$$

```
c = 48*s + 140 % Choosing a controller
```

```
c =
```

$48\,s+140$

```
eqn = subs(denom, [C,G], [c,g]) == 0
```

```
eqn =
```

$$\frac{48\,s+140}{5\,s^2+2\,s-15}+1=0$$

```
verification = solve(eqn, s) % Verification that the poles are -5
```

```
verification =
```

-5

3

3.1 For the rigid system,

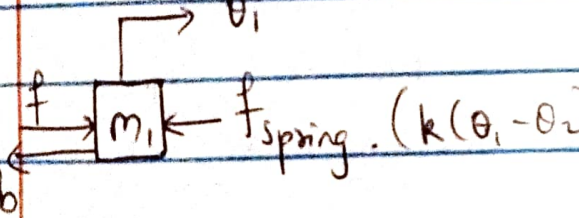
$$f = (m_1 + m_2)\ddot{\theta}_2 + b\dot{\theta}_2$$

For the compliant system;

We draw the free body diagram;

For first block mass m_1, keeping (assuming) m_2 is stationary.

$\because (\theta_2 - \theta_1) < 0$, spring is contracted



$$\therefore m_1\ddot{\theta}_1 = f - k(\theta_1 - \theta_2) - b\dot{\theta}_1$$

$$\Rightarrow f = m_1\ddot{\theta}_1 + b\dot{\theta}_1 + k(\theta_1 - \theta_2)$$

Converting to Laplace :-

$$F(s) = M_1\theta_1 s^2 + b\theta_1 s + k(\theta_1 - \theta_2) \quad \text{---(1)}$$

3.2 (a) We have characteristic equation :-

$$30s^2 + s + 10k_p = 0 \quad \text{(compted in MATLAB)}$$

$$\Rightarrow s^2 + \frac{s}{30} + \frac{k_p}{3} = 0$$

$$\frac{k_p}{3} > 0$$

$\therefore k_p > 0$ for system to be stable.

(b) From the MATLAB, characteristic eqn. we get

$$(2s^4 + 0.2s^3 + 300s^2 + 10s + k_p) = 0$$

$$\Rightarrow s^4 + 0.1s^3 + 150s^2 + 5s + \frac{k_p}{2} = 0$$

Using Routh-Howith form :-

$a_3, a_2, a_1, a_0 > 0$ & $[a_3 a_2 a_1 > a_1^2 + a_3^2 a_0]$

$$\Rightarrow (0.1 \times 150 \times 5) > 25 + 0.01 \times 50 k_p$$

$$\Rightarrow k_p < 100$$

$\therefore$ Range $K_p \Rightarrow \{K_p < 100\}$ or $\{0, 100\}$

Second case, where m_1 is stationary.

$\therefore \theta_2 - \theta_1 > 0$, spring is elongated.

$\therefore$

Note :-

{Elongation} = $\{x_2 - x_1\}$

{Contraction} = $\{x_1 - x_2\}$

$\therefore \quad m_2 \ddot{\theta}_2 = -k(\theta_2 - \theta_1)$

$\Rightarrow$ **Laplace :-**

$$m_2 \theta_2 s^2 = -k(\theta_2 - \theta_1) \quad \text{——— (II)}$$

Solving further on matlab, by isolating θ_1 from eqn (II) and plugging it into eqn (I) :-

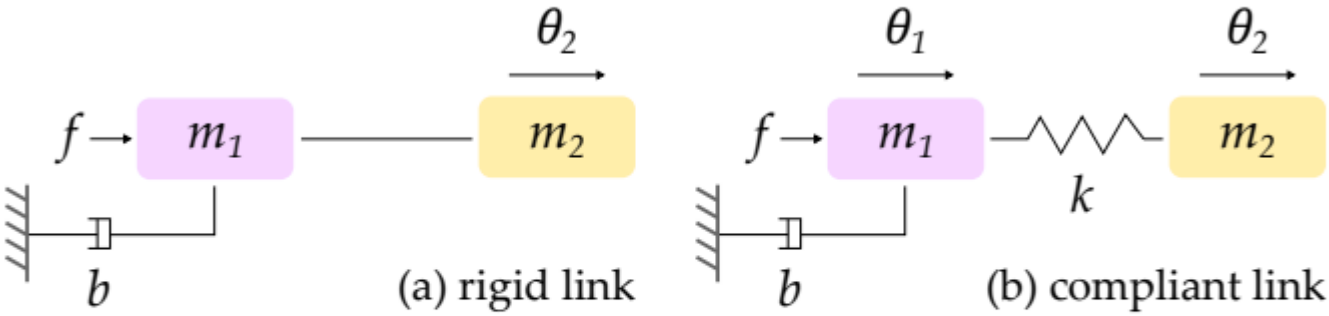
$$G_{\theta_2} = \frac{\theta_2(s)}{F(\theta)}$$

$\Rightarrow$

$$G_{\theta_2} = \frac{k}{s(bk + bm_2 s^2 + m_1 m_2 s^3 + km_1 s + km_2 s)}$$

Verification : (Provided in MATLAB in the end)

3 Compliant Joints



We often introduce mechanical compliance to make the robot soft and safe during interaction. The drawing above compares a rigid link and (Left) and a compliant link (Right). Here m_1 is the motor mass and m_2 is the link mass. For rigid systems we have a rigid connection between motor output and the link, and the total mass is $m_1 + m_2$. For compliant systems we introduce a spring k between the motor output and the link. In both systems we control the actuator force f to regulate θ_2, the position of the link.

```
syms F s m1 m2 theta_1 theta_2 k b real
```

3.1 (5 points)

Find the plant dynamics $G_1(s)$ and $G_2(s)$. Here $G_1(s)$ is the plant for the rigid 1-DoF robot and $G_2(s)$ is the plant for the compliant 1-DoF robot. Both plants should be of the form:

$$G(s) = \frac{\theta_2(s)}{F(s)} \tag{5}$$

```
% G1 evaluation
eqn = F == (m1 + m2)*s^2*theta_2 + b*s*theta_2
```

eqn = $F = \theta_2\,(m_1 + m_2)\,s^2 + b\,\theta_2\,s$

```
G1 = solve(eqn, theta_2)/F
```

G1 =

$$\frac{1}{(m_1 + m_2)\,s^2 + b\,s}$$

```
% G2 evaluation

eqn = F - b*s*theta_1 == m1*s^2*theta_1 + k*(theta_1 - theta_2)
```

eqn = $F - b\,s\,\theta_1 = m_1\,\theta_1\,s^2 + k\,(\theta_1 - \theta_2)$

```
theta_1 = simplify(solve(eqn, theta_1))
```

theta_1 =

$$\frac{F + k\,\theta_2}{m_1\,s^2 + b\,s + k}$$

```
eqn = -k*(theta_2 - theta_1) == m2*s^2*theta_2
```

eqn =

$$-k\left(\theta_2 - \frac{F + k\,\theta_2}{m_1\,s^2 + b\,s + k}\right) = m_2\,s^2\,\theta_2$$

```
G2 = simplify(solve(eqn, theta_2))/F
```

G2 =

$$\frac{k}{s\left(b\,k + b\,m_2\,s^2 + m_1\,m_2\,s^3 + k\,m_1\,s + k\,m_2\,s\right)}$$

3.2 (10 points)

Assume we measure θ_2 in real-time and the closed-loop transfer function is:

$$\frac{\theta(s)}{\theta_d(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)} \tag{6}$$

We will use a proportional controller $C(s) = k_p$. Let $m_1 = 1$ kg, $m_2 = 2$ kg, $b = 0.1$ Ns/m, and $k = 100$ N/m.

- For what range of k_p is the **rigid** robot stable?
- For what range of k_p is the **compliant** robot stable?

```
% 3.2 (a)
```

- For what range of k_p is the **rigid** robot stable?

```
syms C G theta_d theta real

eqn = G*(C*(theta_d - theta)) == theta;
eqn = solve(eqn, theta)/theta_d
```

eqn =

$$\frac{C\,G}{C\,G + 1}$$

```
[~,den] = numden(eqn)
```

den = $C\,G + 1$

```
syms kp real
```

```
c = kp;

g = G1;

eqn = subs(den, [C,G], [c,g]) == 0
```

```
eqn =
```

$$\frac{\mathrm{kp}}{(m_1 + m_2)\, s^2 + b\, s} + 1 = 0$$

We will use a proportional controller $C(s) = k_p$. Let $m_1 = 1$ kg, $m_2 = 2$ kg, $b = 0.1$ Ns/m, and $k = 100$ N/m.

```
eqn = simplify(subs(eqn, [m1, m2, b, k], [1, 2, 0.1, 100])) % Further range is
provided in the written section, this is just to get the simplified characteristic
equation with s and Kp
```

```
eqn =
```

$$30\, s^2 + s + 10\,\mathrm{kp} = 0 \wedge s \neq 0 \wedge s \neq -\frac{1}{30}$$

```
% 3.2 (b)
```

- For what range of k_p is the **compliant** robot stable?

```
c = kp;

g = G2;

eqn = subs(den, [C,G], [c,g]) == 0
```

```
eqn =
```

$$\frac{k\,\mathrm{kp}}{s\,\left(b\,k + b\,m_2\,s^2 + m_1\,m_2\,s^3 + k\,m_1\,s + k\,m_2\,s\right)} + 1 = 0$$

We will use a proportional controller $C(s) = k_p$. Let $m_1 = 1$ kg, $m_2 = 2$ kg, $b = 0.1$ Ns/m, and $k = 100$ N/m.

```
eqn = subs(eqn, [m1, m2, b, k], [1, 2, 0.1, 100]) % Further range is provided in
the written section, this is just to get the simplified characteristic equation
with s and Kp
```

```
eqn =
```

$$\frac{100\,\mathrm{kp}}{s\,\left(2\,s^3 + \frac{s^2}{5} + 300\,s + 10\right)} + 1 = 0$$

3.2 (a) We have characteristic equation :-

$$30s^2 + s + 10k_p = 0 \quad \text{(computed in MATLAB)}$$

$$\Rightarrow s^2 + \frac{s}{30} + \frac{k_p}{3} = 0$$

$$\frac{k_p}{3} > 0$$

$\therefore k_p > 0$ for system to be stable.

(b) From the MATLAB, characteristic eqn. we get

$$(2s^4 + 0.2s^3 + 300s^2 + 10s + k_p) = 0$$

$$\overbrace{}^{a_3} \quad \overbrace{}^{a_2} \quad \overbrace{}^{a_1} \quad \overbrace{}^{a_0}$$

$$\Rightarrow s^4 + 0.1s^3 + 150s^2 + 5s + \frac{k_p}{2} = 0$$

Using Routh-Hourwitz form :-

$a_3, a_2, a_1, a_0 > 0$ & $\left[a_3 a_2 a_1 > a_1^2 + a_3^2 a_0\right]$

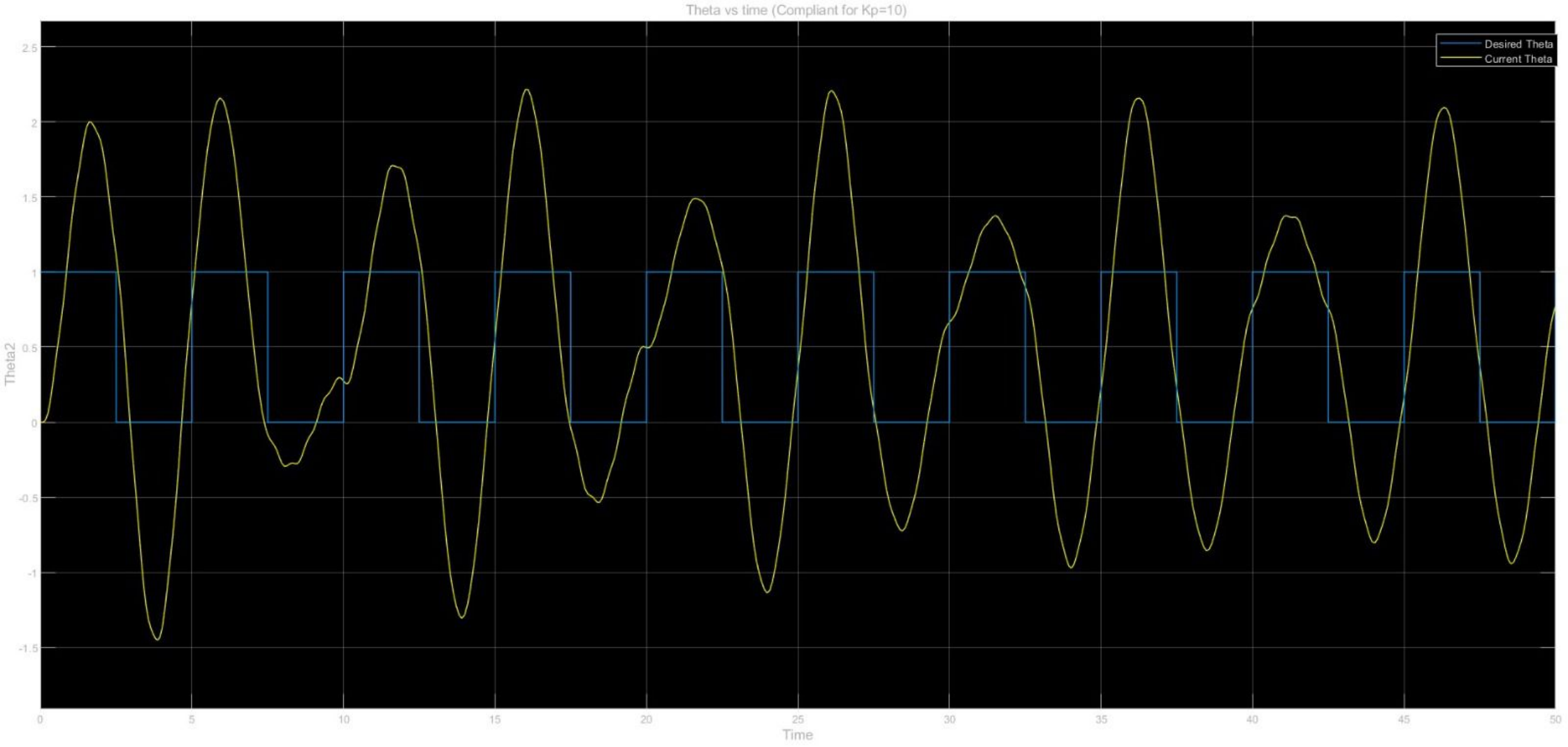
$\Rightarrow (0.1 \times 150 \times 5) > 25 + 0.01 \times 50 k_p$

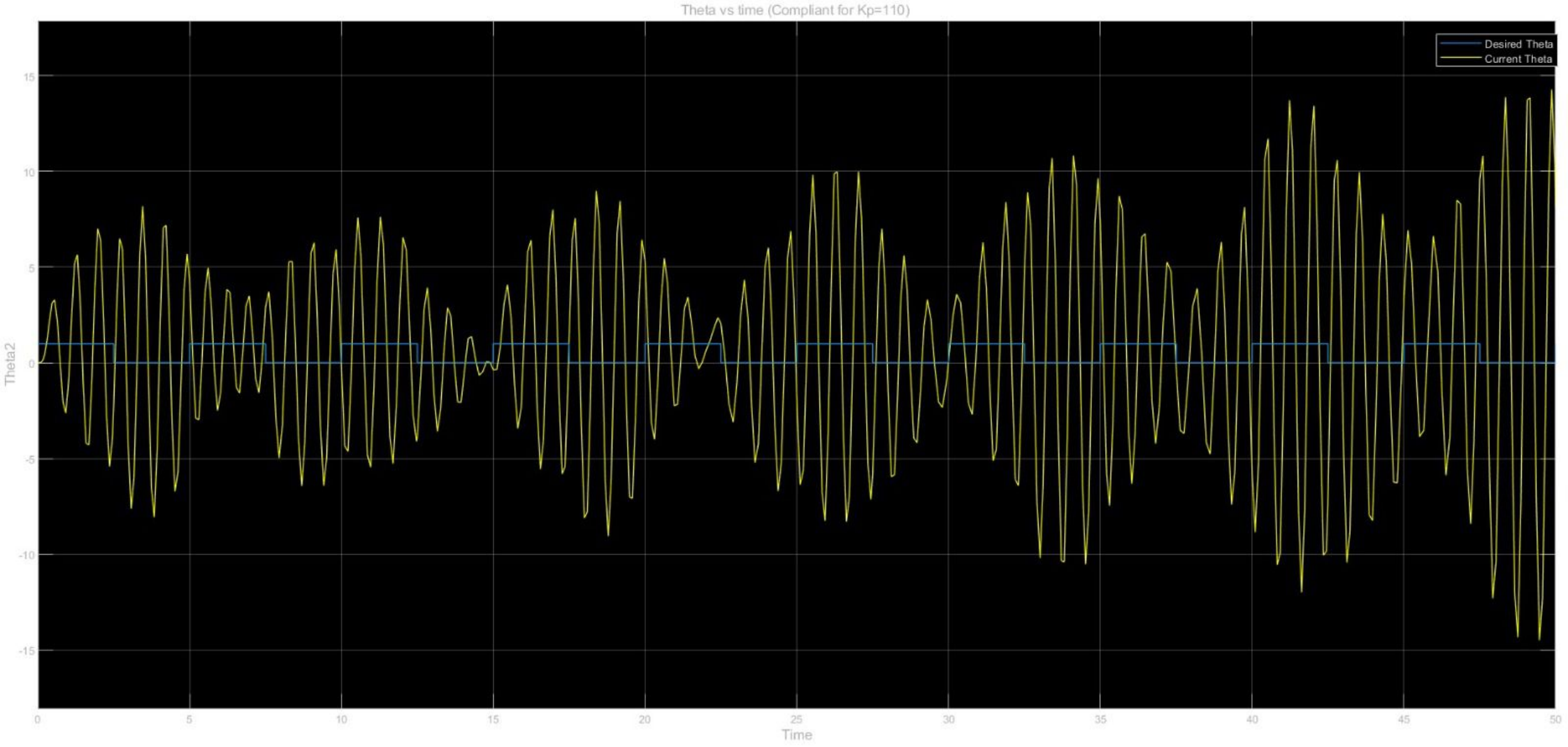
$\Rightarrow \underline{k_p < 100}$

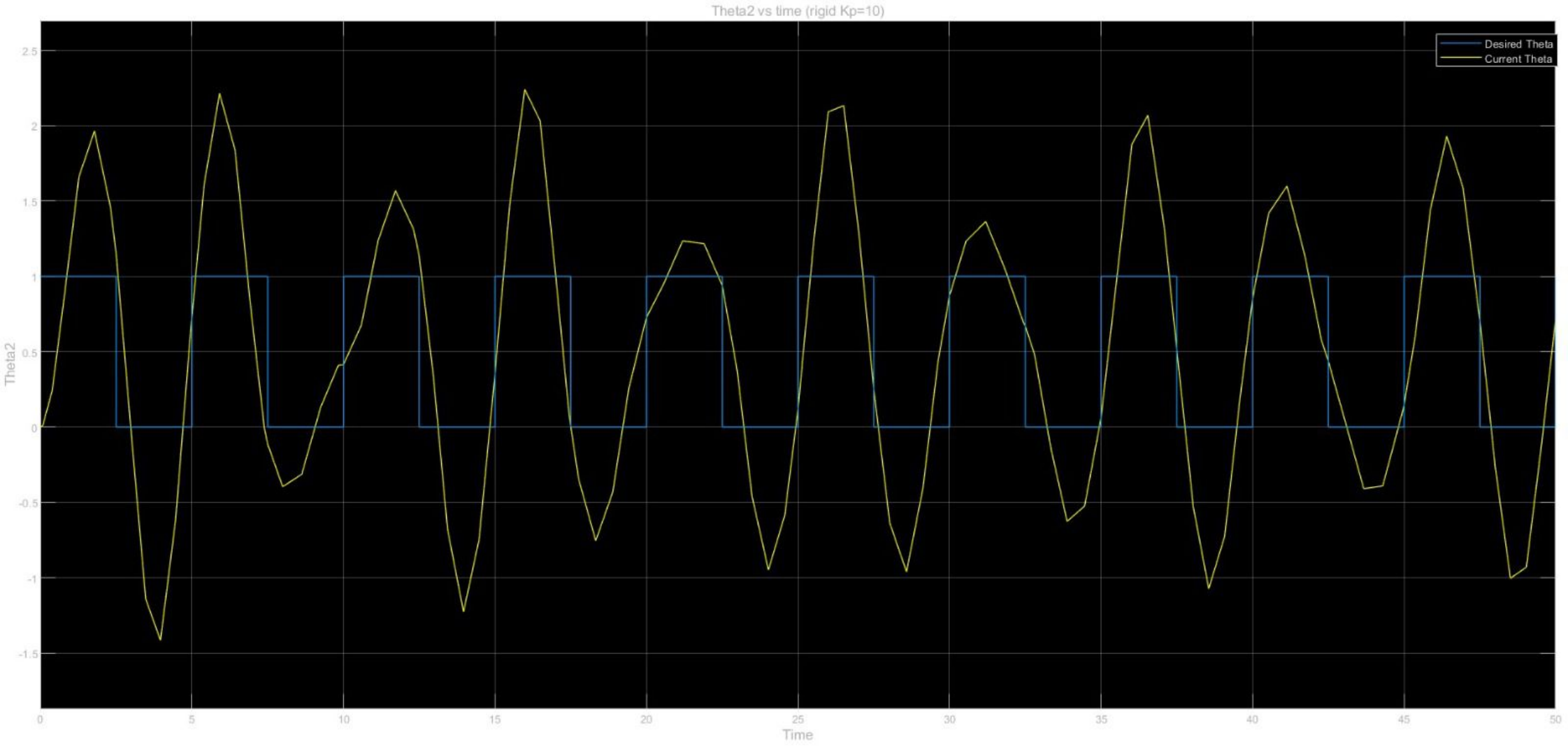
$\therefore$ Range $K_p \Rightarrow \{K_p < 100\}$ or $\{0, 100\}$

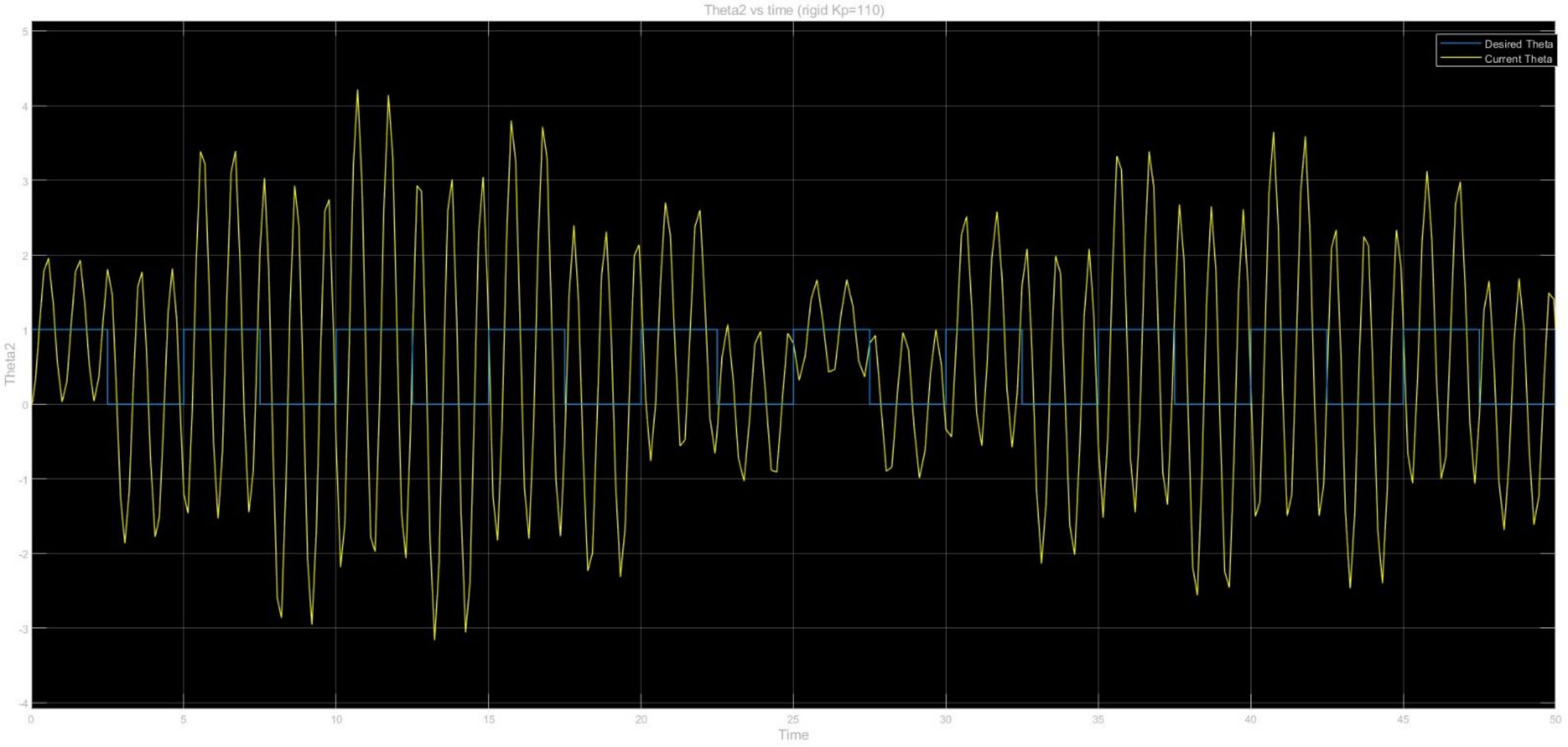
3.3 Pulse generator plots for Rigid and Compliant systems for both $K_p = 10$ and $k_p = 110$

(Mentioned in Title Label)


Theta vs time (Compliant for Kp=10)
Desired Theta
Current Theta
Theta2
Time
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25
30
35
40
45
50


Theta vs time (Compliant for Kp=110)
Desired Theta
Current Theta
Theta2
Time
15
10
5
0
-5
-10
-15
0
5
10
15
20
25
30
35
40
45
50


Theta2 vs time (rigid Kp=10)
Desired Theta
Current Theta
Theta2
Time
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25
30
35
40
45
50


Theta2 vs time (rigid Kp=110)
Desired Theta
Current Theta
Theta2
Time
5
4
3
2
1
0
-1
-2
-3
-4
0
5
10
15
20
25
30
35
40
45
50

3.4] Introducing compliance makes the system harder to control. The compliant system has more complex dynamics with an additional pole (4^{th} Order), so it is more difficult to tune the controller to achieve the desired response.

It is also limited in range of proportional gain that result in stability.

[4] Dynamics for the robot arm:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta)$$

& Controller :

$$\tau = M(\theta)y + C(\theta, \dot{\theta})x + g(\theta) - K_r$$

Where,

$$y = \dot{\theta}_d - \Lambda(\theta - \theta_d)$$

$$x = \dot{\theta}_d - \Lambda(\theta - \theta_d)$$

$$r = (\dot{\theta} - \dot{\theta}_d) + \Lambda(\theta - \theta_d)$$

Here, we compute $\dot{r}$ →

$$\dot{r} = (\ddot{\theta} - \ddot{\theta}_d) + \Lambda(\dot{\theta} - \dot{\theta}_d)$$

And we equate the dynamics and the controller, we get:

$$M\ddot{\theta} + C\dot{\theta} + g(\theta) = My + Cx + g(\theta) - Kr$$

Therefore, we extract $M(\theta)$, we get:-

$$M(\theta) = -\frac{(C+K)(\dot{\theta} - \dot{\theta}_d + \Lambda(\theta - \theta_d))}{\ddot{\theta} - \ddot{\theta}_d + \Lambda(\dot{\theta} - \dot{\theta}_d)}$$

We can see, these are values of r and $\dot{r}$:-

$$\therefore M(\theta) = -\frac{[C+K]\, r}{\dot{r}} \qquad \text{---------- (1)}$$

We have the Lyapunov generalized energy function:

$$v = \frac{1}{2} r^T M(\theta) r + (\theta - \theta_d)^T \Lambda K (\theta - \theta_d)$$

Taking the time derivative to see whether energy is decreasing:

$$\rightarrow \dot{v} = \frac{1}{2}(\dot{r}^T M r + r^T M \dot{r}) + \frac{1}{2} r^T \dot{M} r + 2(\dot{\theta}^T) \Lambda K (\theta - \theta_d)$$

Using passivity and value of r;

$$\rightarrow \dot{v} = r^T M \dot{r} + \frac{1}{2} r^T \dot{M}(\theta) r + 2(\dot{\theta}^T) \Lambda K (\theta - \theta_d)$$

$\rightarrow$ Plugging-in closed-loop dynamics, value of $M(\theta)$ we got from eqn ① here :—

$$\Rightarrow \dot{V} = r^T \dot{r} - (C(\theta,\dot{\theta}) + K) r \dot{r} + \frac{1}{2} r^T \dot{M} r$$

$$+ 2(\dot{\theta}^T) \Lambda K (\theta - \theta_d)$$

Rearranging ;

$$\Rightarrow \dot{V} = -r^T K r + \frac{1}{2} r^T (\dot{M} - 2C) r + 2\dot{\theta}^T \Lambda K (\theta - \theta_d)$$

Since, $(\dot{M} - 2C)$ is skew symmetric, for a vector r, $r^T (\dot{M} - 2C) r = 0$.

$$\Rightarrow \dot{V} = -r^T K r + 2\dot{\theta}^T \Lambda K (\theta - \theta_d)$$

Upon further solving ;

$$\Rightarrow \dot{V} = -\left[(\dot{\theta} - \dot{\theta}_d) + \Lambda(\theta - \theta_d)\right]^T K \left[(\dot{\theta} - \dot{\theta}_d) + \Lambda(\theta - \theta_d)\right]$$

$$- 2\dot{\theta}^T \Lambda K (\theta - \theta_d)$$

$$= -\left[\dot{\theta}^T - \dot{\theta}_d^T + (\theta^T - \theta_d^T)\Lambda^T\right]\left[K(\dot{\theta} - \dot{\theta}_d) + K\Lambda(\theta - \theta_d)\right]$$

$$+ 2\dot{\theta}^T \Lambda K (\theta - \theta_d)$$

$$= \dot{\theta}^T K(\dot{\theta} - \dot{\theta}_d) - K\Lambda\dot{\theta}_d^T(\theta - \theta_d) - K\dot{\theta}_d^T(\dot{\theta} - \dot{\theta}_d)$$

$$+\Lambda^T K(\dot{\theta} - \dot{\theta}_d)(\theta^T - \theta_d^T) + \dots$$

$\Rightarrow$ Upon solving, we find :

$$\dot{v} = -(\theta - \theta_d)^T \Lambda^T K \Lambda (\theta - \theta_d) - (\dot{\theta} - \dot{\theta}_d)^T K (\dot{\theta} - \dot{\theta}_d)$$

Both terms negative, can assume to be of form

$$= -e^T \theta e \leq 0$$

$\because \Lambda$ & K are positive definite matrices,

$\dot{v} < 0$, will be negative definite or semi-definite until $\dot{\theta} = 0$

Energy decreasing.

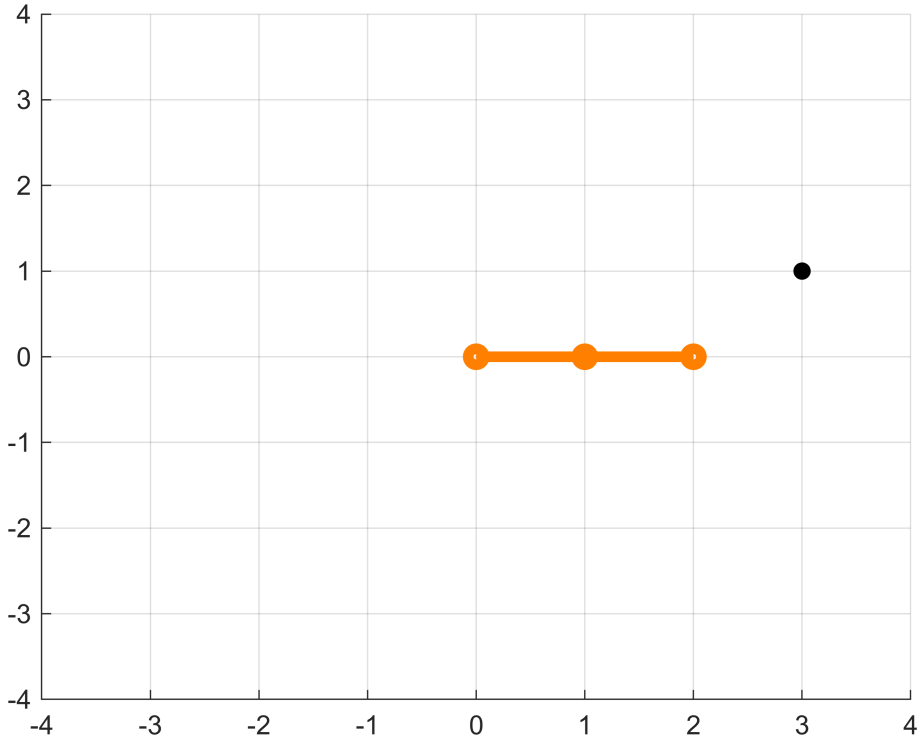
$\star$, $v(t) > 0$ for all value of $t \neq 0$ and $v(0) = 0$

Also, partial derivative $\frac{\partial v}{\partial t} = t$ is continuous.

```
close all
clear
clc

% create figure
figure
axis([-4, 4, -4, 4])
grid on
hold on

% save as a video file
v = VideoWriter(' test.mp4', 'MPEG-4');
v.FrameRate = 100;
open(v);

% pick your system parameters
m1 = 1;
m2 = 1;
m3 = 1;
I3 = 0.1;
L = 1;
g = 9.81;
deltaT = 0.01;

% initial conditions
theta = [0; 0; 0];
thetadot = [0; 0; 0];
thetadotdot = [0; 0; 0];
time = 0;

% forward kinematics to end-effector
S1 = [0;0;0;1;0;0];
S2 = [0;0;0;0;1;0];
S3 = [0;0;1;0;-L;0];
S = [S1, S2, S3];
M3 = [eye(3), [2*L;0;0]; 0 0 0 1];

% For part (a)
% Kp = eye(3);
% Kd = eye(3);

% For part (b), taking random Kp and Kd value to test as mentioned
% Kp = rand(3);
% Kd = rand(3);

% For part (c)
Kp = eye(3)*190;
Kd = eye(3)*20;

M0 = [eye(3), [L;0;0]; 0 0 0 1];
```

```
M1 = [eye(3), [L;0;0]; 0 0 0 1];
M2 = [eye(3), [L;0;0]; 0 0 0 1];

for idx = 1:1000

    % get desired position
    % theta_d = [-2; 2; pi/4];
    % thetadot_d = [0; 0; 0];
    theta_d = [2*cos(pi*time/2); 2*sin(pi*time/2);pi/2];
    thetadot_d = [-pi*sin(pi*time/2); pi*cos(pi*time/2);0 ];
    T_d = fk(M3, [S1 S2 S3], theta_d);

    % plot the robot
    p0 = [0; 0];
    T1 = fk(M1, S1,theta(1:1,:));
    p1 = T1(1:2,4);                          % position of end of link 1
    T2 = fk(M2,[S1 S2],theta(1:2,:));
    p2 = T2(1:2,4);                          % position of end of link 2
    T3 = fk(M3,[S1 S2 S3],theta(1:3,:));
    p3 = T3(1:2,4);                          % position of end of link 3
    P = [p0, p1, p2, p3];
    cla;
    plot(P(1,:), P(2,:), 'o-', 'color',[1, 0.5, 0],'linewidth',4)
    % plot the desired position
    plot(T_d(1,4), T_d(2,4), 'ok', 'MarkerFaceColor','k')
    drawnow
    frame = getframe(gcf);
    writeVideo(v,frame);

    % mass matrix
    M = [m1 + m2 + m3, 0, -L*m3*sin(theta(3));
            0, m2 + m3, L*m3*cos(theta(3));
            -L*m3*sin(theta(3)), L*m3*cos(theta(3)), m3*L^2 + I3];
    % Coriolis matrix
    C = [0, 0, -L*thetadot(3)*m3*cos(theta(3));
        0, 0, -L*thetadot(3)*m3*sin(theta(3));
        0, 0,                 0];
    % gravity vector
    G = [0; g*m2 + g*m3; L*g*m3*cos(theta(3))];

    % choose your controller tau
    e1 =    theta_d - theta;
    e1dot = thetadot_d - thetadot;
    % Reference from a journal and a book (Modern Robotics)
    tau = Kp*(theta_d-theta) + Kd*(thetadot_d - thetadot) + G;

    % integrate to update velocity and position
    thetadotdot = M \ (tau - C*thetadot - G);
    thetadot = thetadot + deltaT * thetadotdot;
    theta = theta + deltaT * thetadot;
```

```
    time = time + deltaT;

end
```



Warning: The video's width and height has been padded to be a multiple of two as required by the H.264 codec.

```
close(v);
close all
```