Control Systems

Lab Assessment #5

Date of experiment – 29/09/2023

Group Members: Adarsh Santoria (B21176), Vanshaj Nathani (B20237)

Objective:

To design a speed controller for a hybrid electric vehicle that ensures

1. Stable performance
2. Maintenance of steady-state error within a predetermined threshold.

Experiment Design:

1. Observing the behaviour based on value of Kp:

Calculations:

1)

$$\frac{C(s)}{R(s)} = \frac{\frac{k_p s + 40}{s}(A)}{1 + \frac{k_p s + 40}{s}(A)}$$

$$\frac{C(s)}{R(s)} = \frac{0.11(s+0.6)(k_p s + 40)}{6s^3 + s^2(3.6127 + 0.11k_p) + s(4.45724 + 0.066k_p) + 2.64}$$

By Routh stability:

s^3	6	$4.45724 + 0.066k_p$
s^2	$3.6127 + 0.11k_p$	2.64
s	$\frac{7.26\times10^{-3}k_p^2 + 0.74094k_p + 0.25}{3.6127 + 0.11k_p}$	0
s^0	2.64	

$k_p 0.11 + 3.6127 > 0.$

$k_p > -32.84$

$k_p \in (-\infty, -101.7) \cup (-0.35, \infty)$

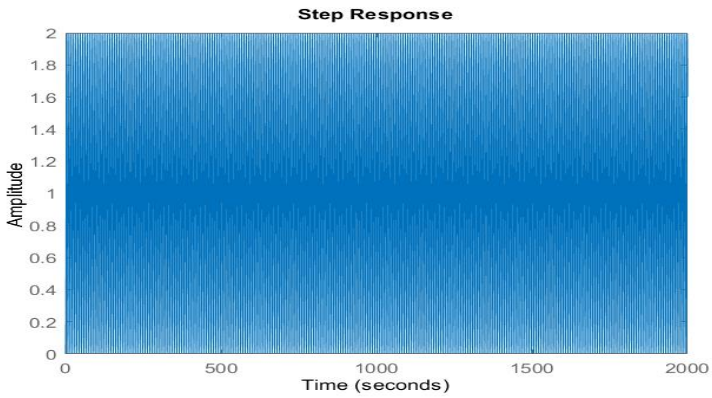
$\therefore \; k_p > 0.351.$

Code:

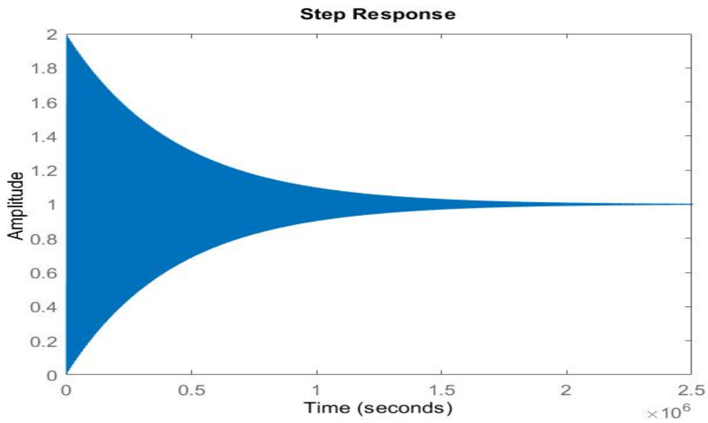
```
% looping around values of Kp
kp = [-.361752 -.3615 -.3618];
for i = 1:3
%     Declaring the transfer function
    tf_org = tf([0.11*kp(i) (0.066*kp(i)+4.4) 2.64],[6 (3.6127+0.11*kp(i)) (4.45724+0.066*kp(i)) 2.64]);
    figure
%     Performing step response
    step(tf_org)
end
```

Results:

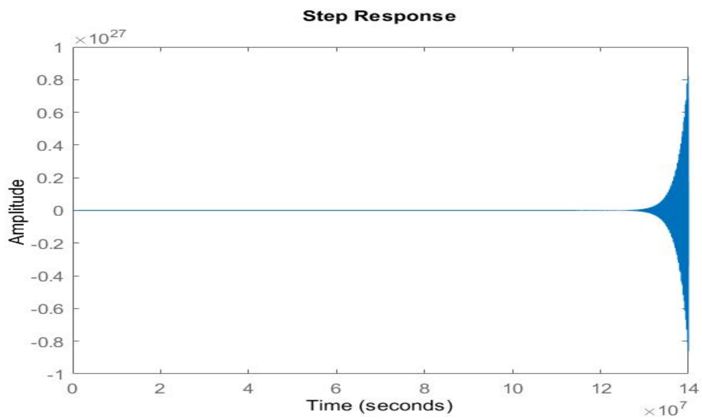
1. Kp = -.361752



2. Kp = -.3615



3. Kp = -.3618



Inferences:

Critical Stability Value: A value of K p >−0.361 (or − 0.351 −0.351 in another method) is required for stability. This value approximates the practical value of 0.361752 0.361752 obtained in simulation.

System Behaviour at K p =0.361752: At this value, the system is critical, indicating a delicate balance between stability and instability.

Stability Analysis:

- Stable System (K p =−0.3618): When K p is greater than the critical value (− 0.3618 −0.3618), the system becomes stable, suggesting that it behaves in a predictable and controlled manner.
- Unstable System (K p =−0.3615): When K p is less than the critical value (− 0.3615 −0.3615), the system becomes unstable, implying that it behaves erratically and uncontrollably.

2. Kp_{sc} verification:

Calculations:

$C(s) = k_{psc} A \cdot E_s \cdot 1$

$R(s) - E(s) = k_{psc} A E(s)$

$$\frac{E(s)}{R(s)} = \frac{s(s+0.5123)+5(s+0.6)(s+0.01908)}{s(s+0.5123)+5(s+0.6)(s+0.01908)+k_{psc}(0.11)(s+0.6)}$$

$R(s) = \frac{1}{s}$

$$E_{ss} = \lim_{s \to 0} s \cdot E(s) = \frac{5 \times 0.6 \times 0.01908}{5 \times 0.6 \times 0.01908 + k_{psc}(0.11)(0.6)} = 0.01$$

$$k_{psc} = 85.86$$

Code:

```
% Value of Kp_sc
kp_sc = 85.86;
% Reduced transfer function
tf_reduced = tf([0.11*kp_sc 0.066*kp_sc],[6 (3.6127) 0.05724]);
err = 1/(1+tf_reduced);
% Calculating the error
[Num,Den] = tfdata(err,'v');
syms s
sys_syms=poly2sym(Num,s)/poly2sym(Den,s);
lim = limit(sys_syms,s,0);
error = double(lim)*100
```

Results:

```
>> lab5_cs_b
error =
      1
```

Inferences:

- When the Kpsc value is decreased, the error value increases, and conversely, increasing Kpsc leads to a decrease in error.
- Our findings demonstrate the validity of this relationship.
- Specifically, at a Kpsc value of 85.86, we achieve a 1% error rate.

3. Integrator added speed controller:

Calculations:

3) $G_{sc}(s) = K_{ps} + \frac{K_{Is}}{s} = 100 + \frac{K_{Isc}}{s}$.

$$\frac{E(s)}{R(s)} = \frac{1}{1 + G_{sc}A} = \frac{s}{s + (100s + K_{Isc})A} = \frac{s\left(s(s+0.5173) + 5(s+0.6)(s+0.01908)\right)}{s\left(s(s+0.5173) + 5(s+0.6)(s+0.01908)\right) + (100s + K_{I})(0.11)(s+0.6)}$$

Ramp function $R(s) = 1/s^2$

$$E(s) = \frac{1}{s^2}\left[\frac{E(s)}{R(s)}\right]$$

$$E_{ss} = \lim_{s \to 0} sE(s) = \lim_{s \to 0} E(s) = 0.025.$$

$$0.025 = \frac{5 \times 0.6 \times 0.01908}{K_{Isc} \times 0.11 \times 0.6}$$

$K_{Isc} = 34.69$.

Code:

```
syms s;
% declaring Ki
ki = 34.65;

% Transfer function initialization
tf_w = (0.11*s+0.066)/(6*s^2+3.6172*s+0.05724);


err = s/(s^2*(1+((100+ki/s)*(tf_w))));

lim = limit(err,s,0);
ep = double(lim)*100;
```

Results:

```
>> lab5_cs_c

ep =

    2.5030
```

Inferences:

- Increasing the Ki value results in a decrease in error, whereas decreasing Ki leads to an increase in error.
- Specifically, we obtained a 2.503% error for a Ki value of 34.68.
- This calculated value closely approximates the desired 2.5% error.

4. Integrator added speed controller:

Calculations:

$$E(s) = X(s)\left[\frac{6.13\times10^{-7}}{s+0.7908}\right]$$

$$X(s) = T_H(s) - X(s)\left[\frac{13.535}{s+0.5}\cdot\left(100+\frac{70}{s}\right)\cdot\frac{1(s+0.1)}{(s+0.5)}\right]$$

$$= T_H(s) - X(s)\left[\frac{313.535 s + 7004.72}{s(s+0.5)}\right]$$

$$E(s) = \frac{T_H \times 6.13\times10^{-7}}{s+0.01903}\cdot\frac{s(s+0.5)(s+0.01903)}{s(s+0.5)(s+0.01903) + (323.535 s + 7004.728\times10^{-7})}$$

~~$E(s) = T_H(s) - E(s)$~~

$$X(s) = \lim_{s\to0} s\,E(s)$$

$$X(s) = \lim_{s\to0}\left(T_H - E(s)\right)$$

$$= \lim_{s\to0} T_H(s)\times s\left[1 - \frac{6.13\times10^{-7}\times(s+0.5)(s+0.01903)}{s(s+0.5)(s+0.01903) + (343.5 s + 7004.72)(6.13\times10^{-7})}\right]$$

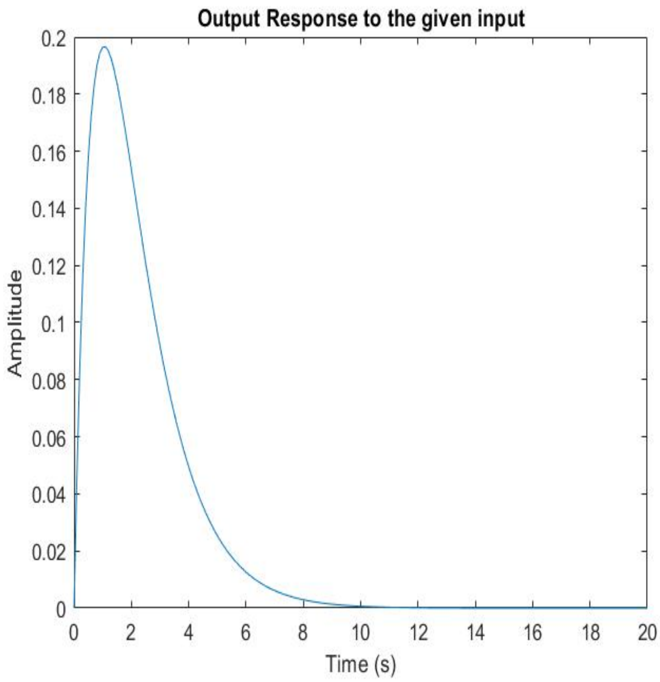
$$X(s) = 83.2$$

$$E(s) = 83.2 - 83.2 = 0.$$

Code:

```
syms s;

% Initializing the given transfer functions
tf_g = 6.13e-3/(s+0.01908);

tf_h1 = 13.53*s/(s+.5);
tf_h2 = ((100*s+40)/s)*(3*(s+.6)/(s+.5));
tf_ac = tf_g/(1+tf_g*(tf_h1+tf_h2));

% Error output wrt given input torque
err = (tf_ac)*(83.7/s);
ep = double(limit(err*s,s,0))*100

% Step response
t = 0:0.1:20;
e = subs(ilaplace(err));
plot(t,e);
xlabel("Time (s)")
ylabel("Amplitude")
title("Output Response to the given input")
```

Results:




```
>> lab5_cs_d

ep =

     0
```

Inferences:

- Initially, the error (system output in response to the given input) shows a rise, reaching a peak of 0.2.
- After around 1.5 seconds, the output begins to decline and eventually stabilises at zero.
- This pattern is a result of the initial oscillations settling down. In the steady state, as per our calculations, the output remains stable at zero.

Improvements and Learnings:

1. In the simplified scenario, it's evident that the system significantly improved upon incorporating the Ki component.
2. Furthermore, introducing a derivative controller could further enhance the system's performance.
3. We can now practically assess the indirect analogy of a PID controller in this context as well.