RIS - LAB 2

Prof. Dr. Fanging Hu

Author

Harshit Mutha

LAB 2 – (TASK 1.25-1.27)

TASK 1.25

For this part, we started by basing our system on one of the equations. We have to show that when L_a is negligible, we have the following equations for $H_{er}(s)$:

$$H_{er}(s) = \frac{s^2 D(s)}{Q(s)},$$
$$Q(s) = R_a J\, s^3 + s^2(R_a b + k_e k_t + k_t K_D) + k_t K_P\, s + k_t K_I$$

We can prove this as follows:

We know that $H_{wv}(s) = \frac{k_t}{sD(s)}$ with $\rightarrow$ (1)

$D(s) = (R_a + L_a s)(Js + b) + k_e k_t$

$\boxed{\text{If } L_a \text{ negligible, } D(s) = R_a(Js+b) + k_e k_t}$

and $H_{er}(s) = \frac{1}{1 + H_{wv}(s).G(s)}$ with

$$G(s) = k_p + k_I \cdot 1/s + k_D \cdot s = \frac{s^2 k_D + s k_p + k_I}{s} \rightarrow (2)$$

Using (1) and (2)

$$H_{er}(s) = \frac{1}{1 + \frac{k_t}{sD(s)} \cdot \left(\frac{s k_p + k_I + s^2 k_D}{s}\right)}$$

$$= \frac{s^2 D(s)}{s^2 D(s) + k_t(s^2 k_D + s k_p + k_I)}$$

$$= \frac{s^2 D(s)}{s^2 D(s) + k_t k_p s + k_t k_I + s^2 k_D k_t}$$

[Plugging in $D(s)$ from above] in Denom

$$= \frac{s^2 D(s)}{s^2[R_a(Js+b) + k_e k_t] + k_t k_p s + k_t k_I + s^2 k_D k_t}$$

$$= \frac{s^2 D(s)}{s^3 R_a J + s^2(R_a b + k_D k_t + k_e k_t) + s k_p k_t + k_I k_t}$$

TASK 1.26

For this part, we use our Simulink model from before. By copying the model-file for the PID speed controller and changing the speed-controller to a servo-controller and get a model as follows:

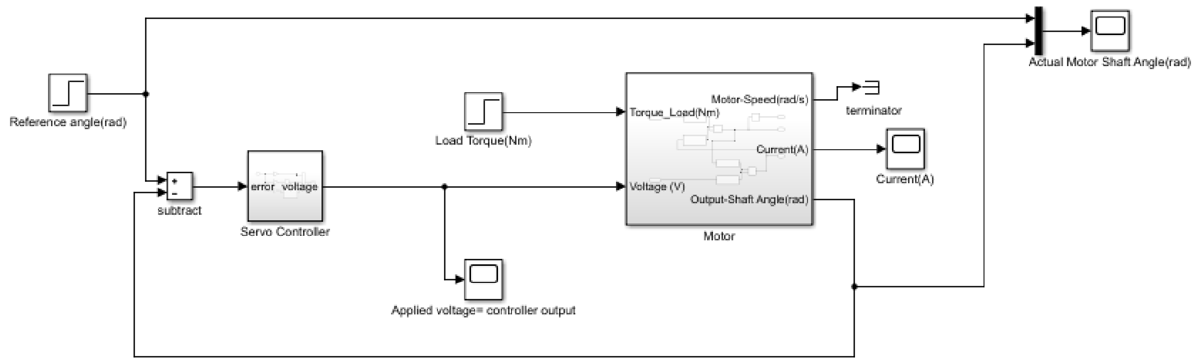

Figure 1.1 Simulink Model including Servo Controller

TASK 1.27

Part 1, 2 and 3

For this part, we first initialize our values. $K_P = 1$, $K_D = 0$, $K_I = 0$.

We set the refence input to $\frac{\pi}{3}\,rad$ and $T_L = 0$. After this, we simulate are model for 30 secs and get the scopes as follows:

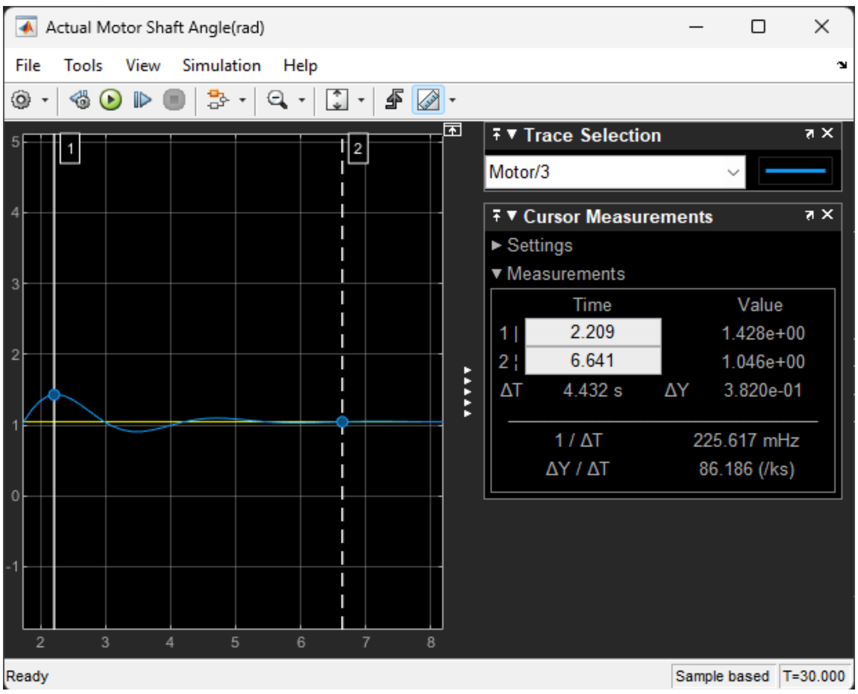

Figure 2.1: Actual Motor shaft scope when Kp=1. Here, we get angle and settle time

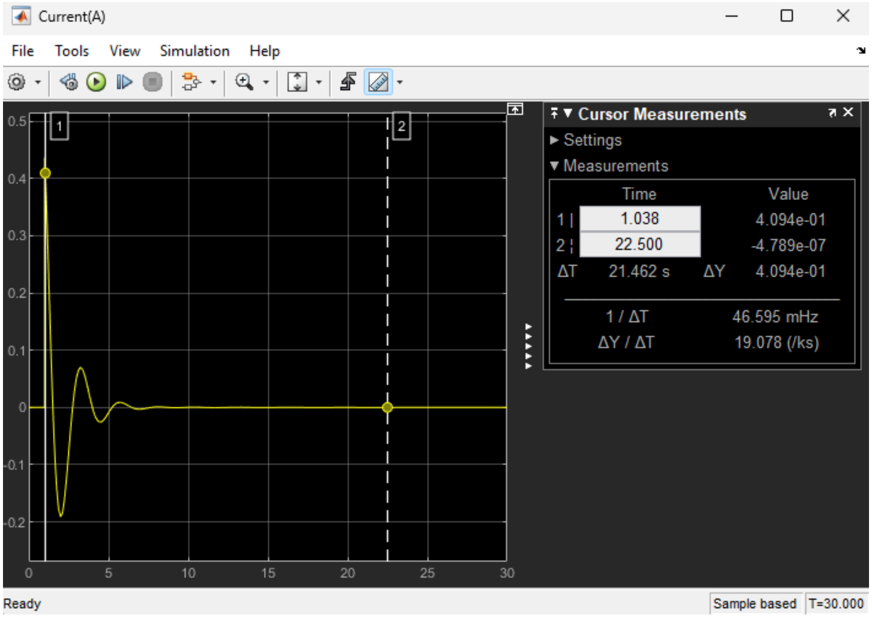

Figure 2.2: Current Scope. Here, we get the max current (A)

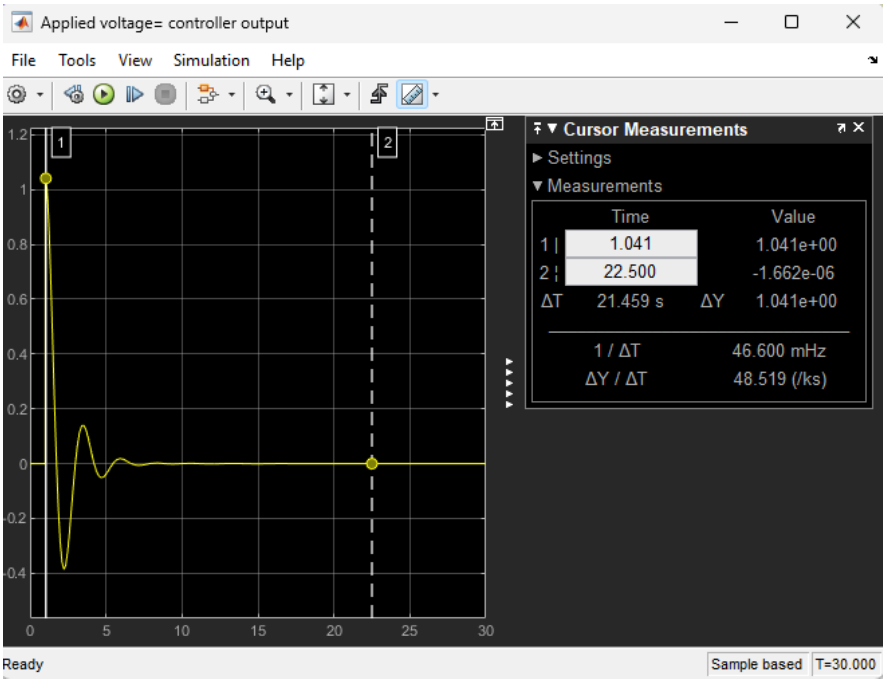

Figure 2.3: Applied Voltage scope. Here, we get the max voltage (V)

Considering the values from the above scopes and repeating for other values of K_I, K_D, we can create a table containing K_P, K_I, K_D, max abs. value of Current, Voltage, angle above reference and settle time which is the time needed for angle to settle near its steady state value. The table is below:

N	K_P	K_I	K_D	Max Current(A)	Max Voltage(V)	Max overshoot of angle(rad)	Settle-time(s)
1.	1	0	0	0.4094	1.041	1.428	6.641
2.	1	0	0.1	3.754	9.805	1.284	3.704
3.	1	0.1	0.1	4.890	11.364	11.124	4.950

4)

For this part, we want to see what the controller can do if we add a step torque. We first reset the value of $K_I = 0$ and change to $K_P = 0.5$ and $K_D = 0.001$. We also add a step load of $T_L = 0.01$Nm at t = 10 secs.

We get the results as follows; however, we can observe that the system doesn't reach its reference value due to the addition of torque at t = 10 secs.

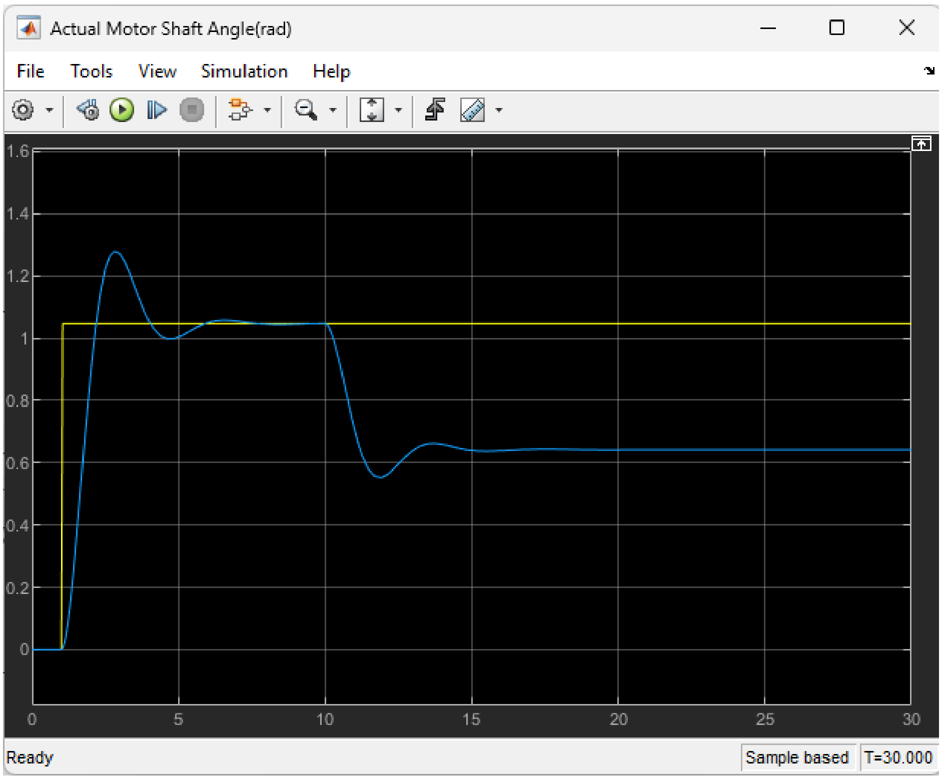

Figure 3.1: Motor Shaft Scope when Ki = 0, Kp = 0.5, Kd = 0.001 and TL = 0.01Nm at t-10s

5)

For the next step, we increase K_I to $K_I = 0.1$ and simulate the model. We can see from the scope that when K_I is increased, the system reaches its reference value, while kind of minimizing steady state error to a noticeable level.

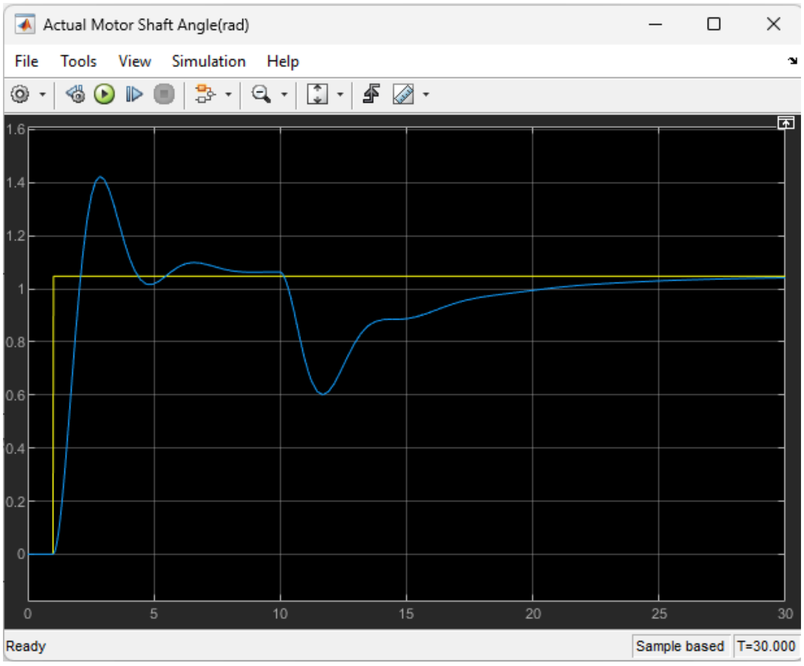

Figure 4.1: Motor Shaft scope at Ki = 0.1

6)

We know from the lab manual that it's tempting to increase K_I, it is better to focus on how to increase the value of K_I. If K_I value is too big, it can easily make then closed-system unstable which is obviously something we should avoid. For checking how this works and developing a concise and better concept, we change K_I value to $K_I = 1$ and get the results as follows:

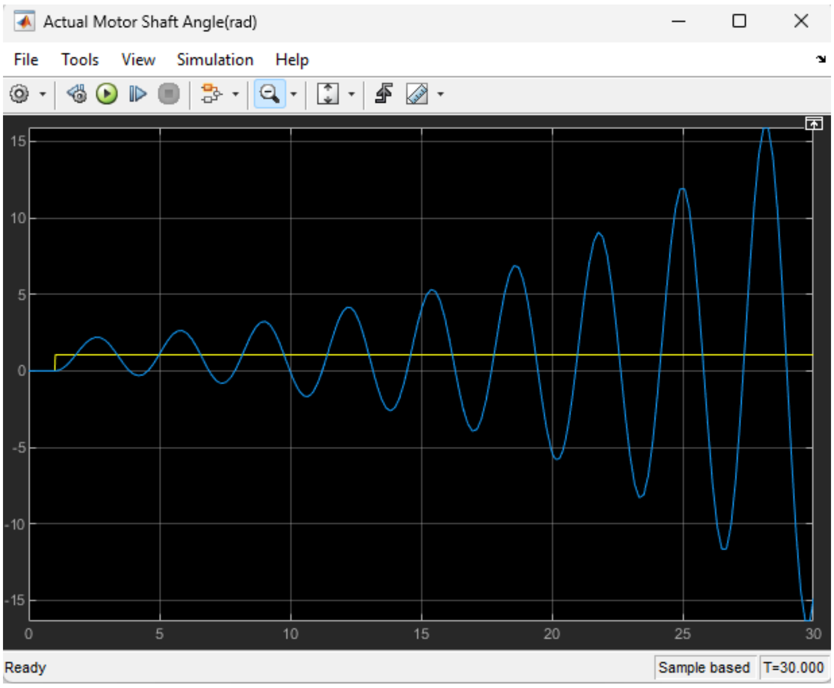

Figure 5.1: Motor shaft scope at Ki = 1

7)

For this part, we will check if our model also predicts the instability of the system by finding the poles. In order to find the roots of Q(s), we use MATLAB's commands. To begin with, we set new values of controller parameters and find roots as follows:

```
>> Inputref = pi/3

Inputref =

    1.0472

>> K_P = 0.5

K_P =

    0.5000

>> K_D = 0.001

K_D =

   1.0000e-03

>> K_I = 1

K_I =

     1

>> Q = [R*J, (R*b + ke*kt + kt*K_D), kt*K_P, kt*K_I];
>> roots(Q)

ans =

   0.0969 + 1.9679i
   0.0969 - 1.9679i
  -1.8185 + 0.0000i
```

We can know by assessing the results that not all the roots are negative and situate at the Left side of plane. Hence, we had already predicted the system's instability, given the controller parameters and can conclude that this system is not stable.