LAB 6: Stability Analysis, Frequency Response & Complete Controller Design

Lab 6

Part 2

Q2-1 A PI controller can be used to avoid large disturbances and noise presents during operation process. PID controller can be used when dealing with higher order capacitive processes.

PI controller	PID Controller
★ Effective for many fast processes, such as flow, pressure, and tempreture loops	★ very effective for slower processes
★ equivalent of PID controller w/ its D (derivative) term set to zero	★ derivative term is important for integrating processes, such as level, position, & well-insulated tempreture
	★ using the D term allows for a larger proportional gain, and can significantly increase speed of response of non-integrating process.
★ response is slower, thus enabling a smooth + accurate PV change	★ Response is faster, thus enabling setpoint to be reached more quickly
★ no overshoot	★ some amount of overshoot may occur

Q2-2 I would chose a PI controller because there is zero steady state error, stability & max peak overshoot is better than Integral only controller, and because we are not concerned with slow moving process variables.

Q3-1

$$\text{output} = k_p e(t) + k_I \int_0^t e(t)\,dt + \cancel{k_D \frac{de(t)}{dt}} + \text{bias}$$

$$V_D = k_p (W_z^* - W_z) + k_I \int_0^t (W_z^* - W_z)\,dt + G_W W_z^*$$

Lab 6 cont.

Q3-2

$$M_z \dot{w}_z = G_T\left(k_p(w_z^* - w_z) + k_I \int_0^t (w_z^* - w_z)\,dt + \cancel{G_w w_z^*} - \cancel{G_w w_z^*}\right)$$

$$M_z \dot{w}_z = k_p G_T (w_z^* - w_z) + k_I G_T \int_0^t (w_z^* - w_z)\,dt$$

Q3-3

$$\mathcal{L}\left[M_z \dot{w}_z\right] = \left[k_p G_T (w_z^* - w_z) + k_I G_T \left[\int_0^t w_z^*\,dt - \int_0^t w_z\,dt\right]\right]\mathcal{L}$$

$$M_z\left[s W_z(s) - \cancel{W_z(0)}\right] = k_p G_T\left(W_z^*(s) - W_z(s)\right) + k_I G_T\left[\frac{W_z^*(s)}{s} - \frac{W_z(s)}{s}\right]$$

$$\cancel{s W_z(s) = \frac{k_p G_T}{M_z}\left(\frac{s W_z^*(s) - s W_z(s) + W_z^*(s) - W_z(s)}{s}\right)}$$

$$s^2 W_z(s) = \frac{G_T}{M_z}\left[k_p(s W_z^*(s) - s W_z(s)) + \left[W_z^*(s) - W_z(s)\right]k_I\right]$$

$$\frac{s^2 W_z(s) M_z}{G_T} = k_p s W_z^*(s) - k_p s W_z(s) + k_I W_z^*(s) - k_I W_z(s)$$

$$\frac{M_z}{G_T} s^2 W_z(s) + k_p s W_z(s) + k_I W_z(s) = k_p s W_z^*(s) + k_I W_z^*(s)$$

$$W_z(s)\left(\frac{M_z}{G_T} s^2 + k_p s + k_I\right) = W_z^*(s) = (k_p s + k_I)$$

$$\frac{W_z(s)}{W_z^*(s)} = \frac{k_p s + k_I}{\frac{M_z}{G_T} s^2 + k_p s + k_I} = \boxed{\frac{k_p s + k_I}{.010667 s^2 + k_p s + k_I}}$$

$$M_z = \frac{I_z dr}{dw} = \frac{.0010(.034)}{.085} = .0004$$

$$G_T = \frac{k_T}{R} = \frac{.2025}{5.4} = .0375$$

$$\frac{M_z}{G_T} = \frac{.0004}{.0375} = .010667$$

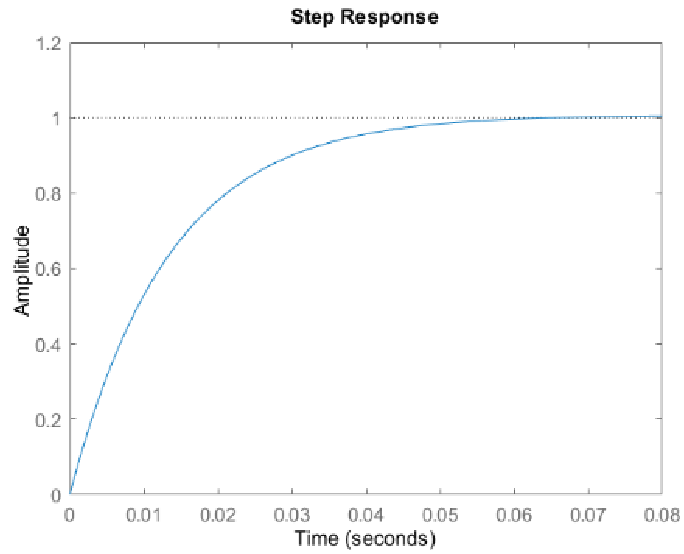
$k_p = .8$

$k_I = .5$

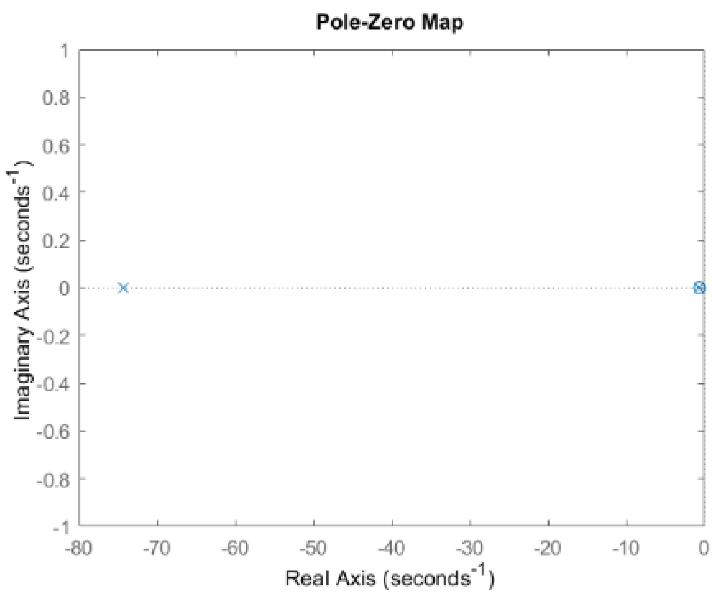
$$\boxed{\text{Transfer function} = \frac{W_z(s)}{W_z^*(s)} = \frac{.8s + .5}{.010667s^2 + .8s + .5}}$$

Q4-1

```
num = [.8,.5];
dem = [.010667, .8, .5];

sys= tf(num, dem)

pole(sys)

step(sys)

pzmap(sys)
```




```
sys =

        0.8 s + 0.5
  -------------------------
  0.01067 s^2 + 0.8 s + 0.5

Continuous-time transfer function.
ans = 2x1
   -74.3674
    -0.6303
```



Pole values: -74.3674 and -0.6303
Gain values:

- Kp = 0.8
- Ki = 0.5

Q4-2:

- Kp = 0.8
- Ki = 0.5

Lab 6

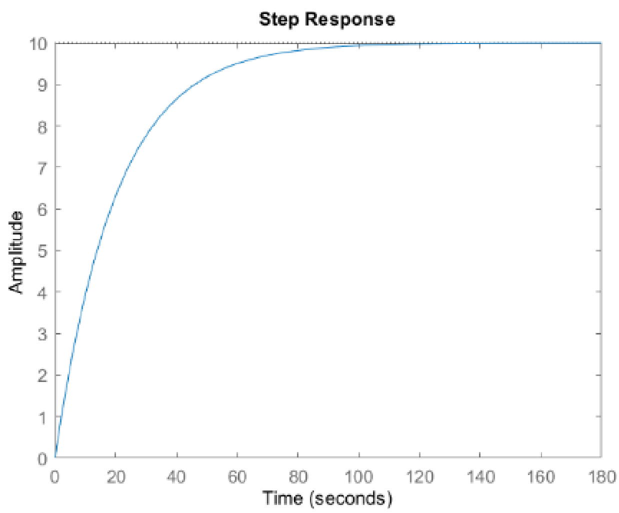
$\boxed{\text{Cruise - Control Car}}$

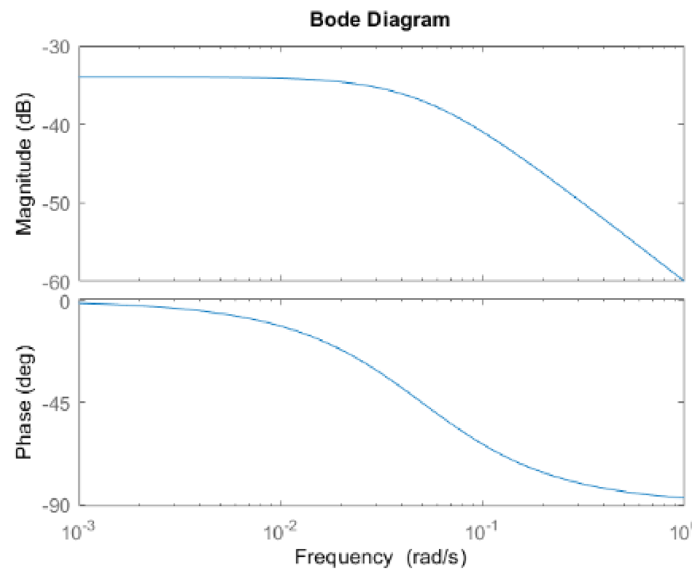
1) $OLTF = C(S)\,G(S)$

$C(S) = K_p = 1$

$P(S) = \dfrac{V(S)}{U(S)} = \dfrac{1}{ms+b} = \dfrac{1}{1000s+50}$

```
m = 1000;
b = 50;
u = 500;

Kp = 1;
s = tf('s');
P_cruise = 1/(m*s+b);
C = Kp;
step(u*C*P_cruise)
bode(C*P_cruise)
```



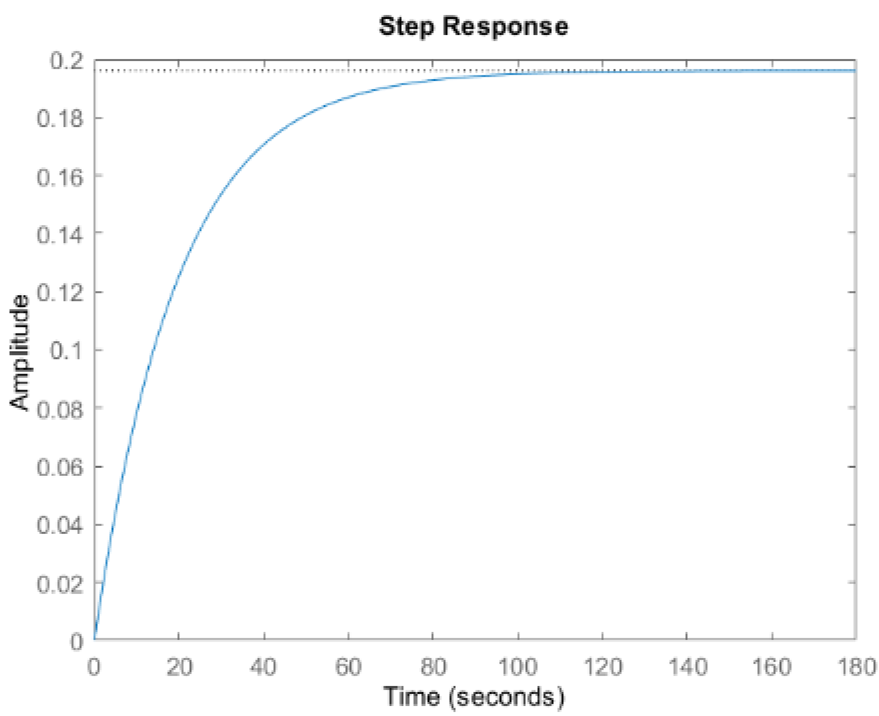


2) $\frac{Y(s)}{\mathcal{E}(s)} = \frac{k_p}{ms+b}$

$$e_{ss} = \lim_{s\to 0} s\,\mathcal{E}(s) = \frac{1}{1+k_{p_1}} = \frac{1}{1+\frac{1}{50}} = .980$$

$$k_{p_1} = \lim_{s\to 0} p(s) = \lim_{s\to 0} \frac{1}{ms+b} = \frac{1}{b} = \frac{1}{50}$$

```
r = 10;
sys_cl = feedback(C*P_cruise,1);
step(r*sys_cl);
```



3) $\frac{1}{1+k_{p_1}} < .02$

$\frac{1}{.02} - 1 < k_{p_1}$

$k_{p_1} > 49$

$20 \log(49) = 33.8 \text{ dB}$

$20 \log(.02) = -33.98 \text{ dB} = -34 \text{ dB}$

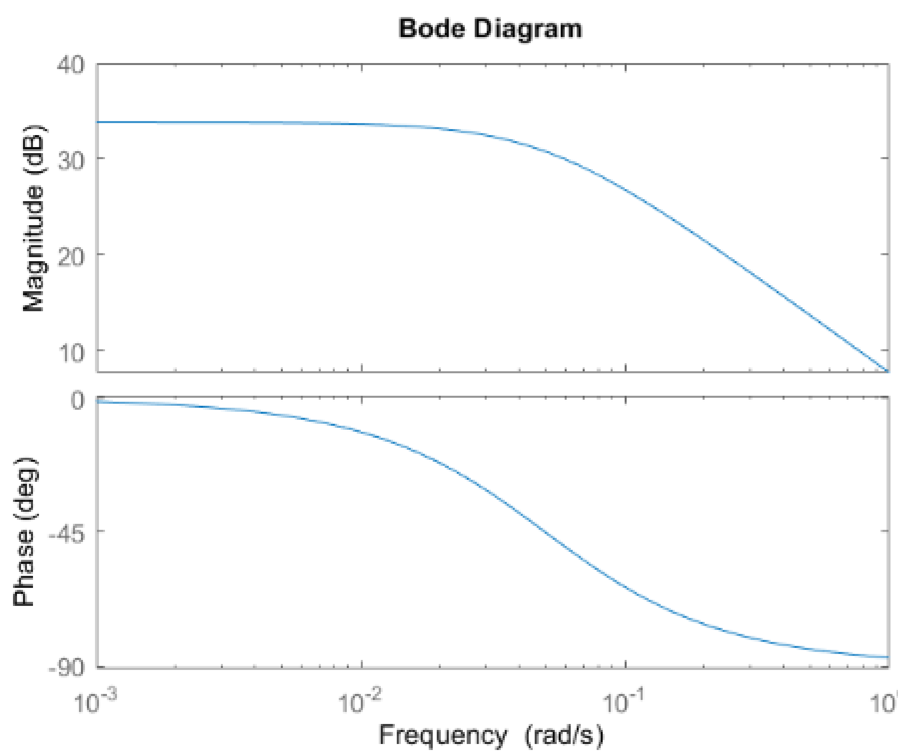
$k_{p_1} > (33.8 + 34) \text{ dB}$

$\boxed{k_{p_1} > 67.8 \text{ dB}}$

$20 \log(x) = 67.8 \text{ dB}$

$x = 2454.71$

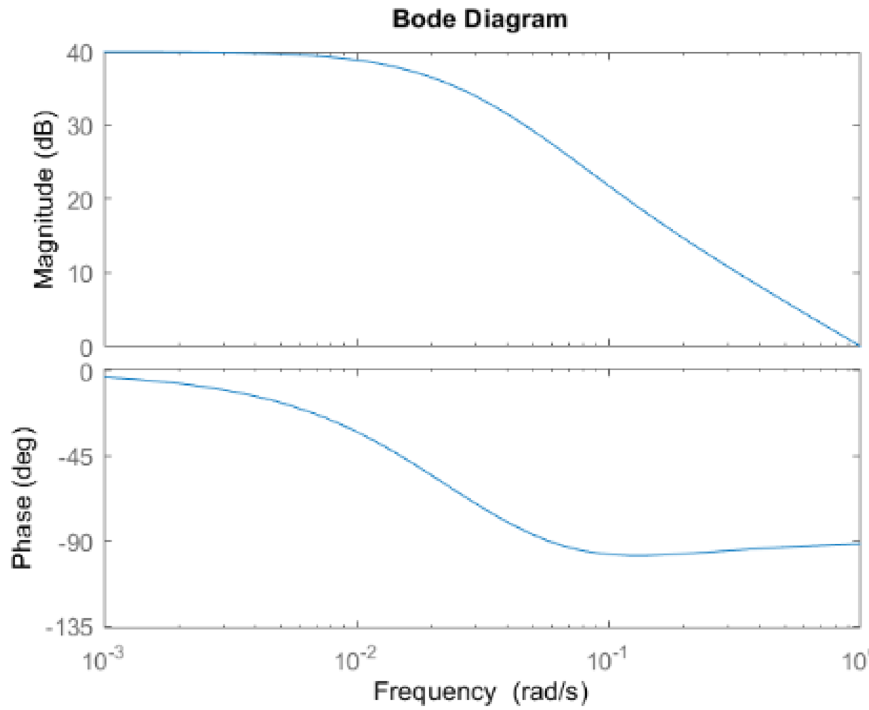
```
Kp = 2455;
C = Kp;

bode(C*P_cruise);
```



4. The steady-state error meets the requirements; however, the rise time is much shorter than is needed and is unreasonable in this case since the car can not accelerate to 10 m/s in 2 sec. Therefore, we will try using a smaller proportional gain to reduce the control action required along with a lag compensator to reduce the steady-state error.

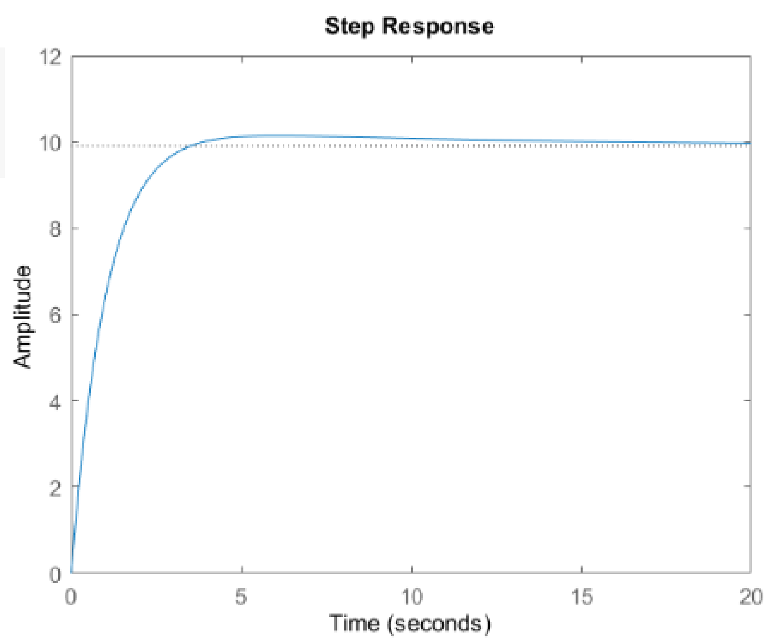
5.

```
Kp = 1000;
zo = 0.1;
po = 0.02;

C_lag = (s+zo)/(s+po);
bode(Kp*C_lag*P_cruise);
```



Confirm the performance by generating a closed-loop step response:

```
sys_cl = feedback(Kp*C_lag*P_cruise,1);
t = 0:0.1:20;
step(r*sys_cl,t);
```



There is a very slight overshoot, the steady-state error is close to zero, and the rise time is under 5 seconds. The system has now met all of the design requirements. No more iteration is needed.

Introduction:

In this lab, we learned how to analyze stability and frequency responses with bode plots. Additionally, we learned how to reduce steady-state error and learn how to create a lag compensator, to ultimately complete controller design for the BalBot (self-balancing robot).

Conclusion:

From this lab, we were able to understand how to analyze stability and frequency response. In addition to designing complete controllers for real robotic systems (the BalBot) using state feedback and PID controller. We used our derived equations to find the P-D control variables, then analyzed our function/system in Matlab.