

第 15 組:控制系統第一次作業

組員:呂侑宸、陳柏全、郭典泰、李承洋

1.

問題 1: (系統模型鑑別)

考慮圖一系統 A 之單位步階響應

(i) 試鑑別系統 A 之模型。(20%)

(ii) 驗證所得模型之正確性。(10%)提醒:『大膽假設, 小心求證』!!

1.列出假設並說明原因

2.列出詳細推理過程及結論



圖一 系統 A 之步階響應

(i)由圖可知:

峰值(C_{max})約等於 1.25

峰值時間(T_p)約等於 0.6

假設 ζ 值會滿足峰值(C_{max}) 約等於 1.25

將 $\zeta=0.404$ 代入公式:峰值(C_{max})= $1 - e^{-\left(\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right)}$

會滿足峰值(C_{max})約等於 1.25

再將 $\zeta=0.404$ 代入公式:峰值時間(T_p)=

$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}$$

求得 ω_n 約等於 5.72

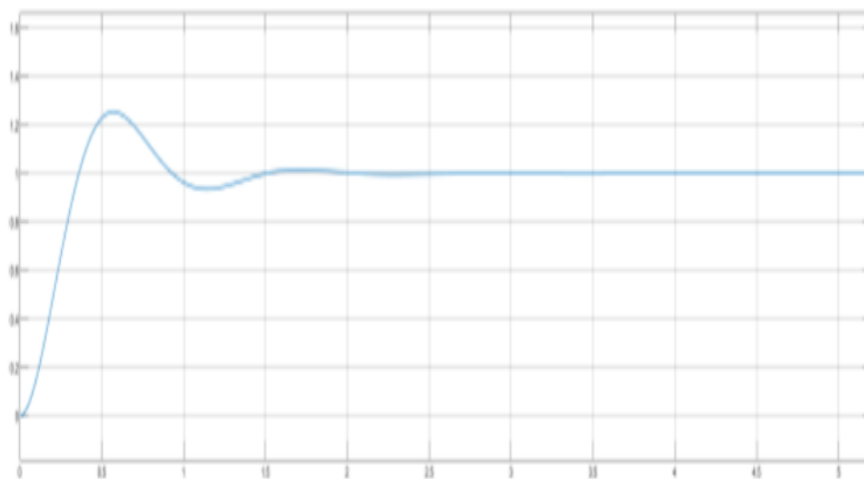
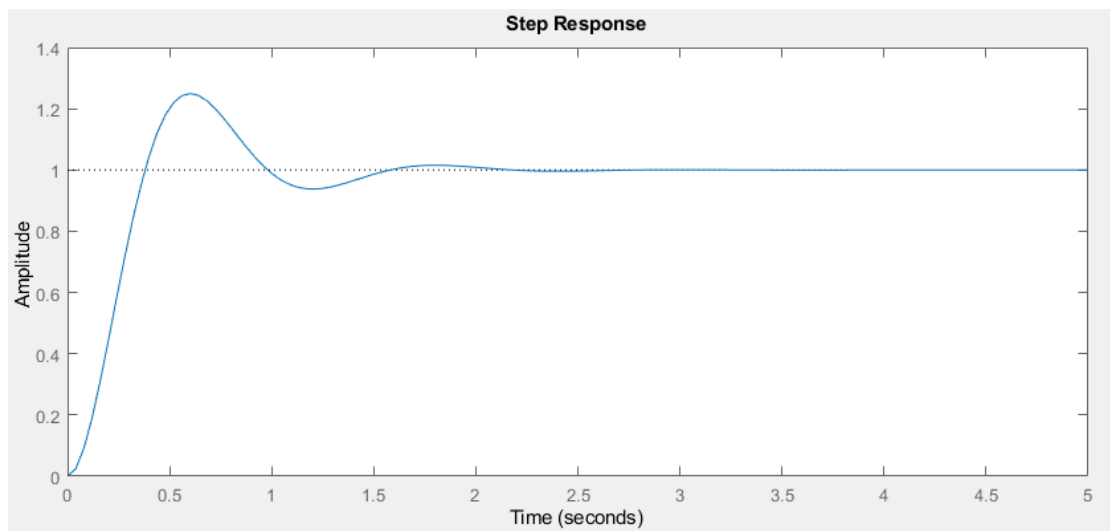
再將 ζ 與 ω_n 代入公式：
$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$2\zeta\omega_n$ 約等於 4.622

所以我們得出的轉移函數答案為

$$G(s) = 32.72 / (s^2 + 4.622s + 32.72)$$

(ii)驗證:和原先的步階響應圖對比:



圖一 系統 A 之步階響應

從圖形的相似程度可驗證假設的正確性。

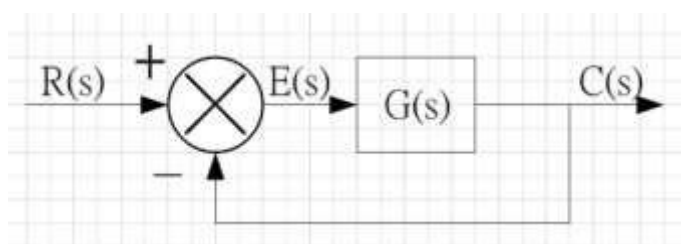
2.

問題 2: (凹陷濾波器(Notch filter) 設計)

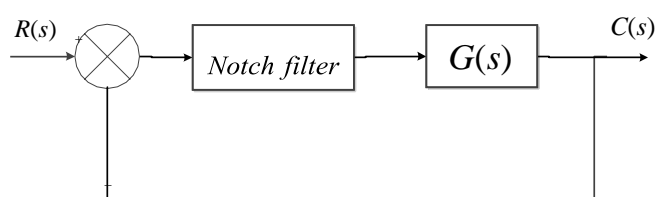
(i) 如圖二， $G(s)$ 有 4 個極點、無零點。試舉出一例 $G(s)$ 為多少，可使圖二閉迴路系統之單位步階響應有高頻震盪。畫出閉迴路系統之根軌跡及單位步階響應。

(15%)

(ii) 如圖三， $G(s)$ 如(i)所得。設計一凹陷濾波器，改善高頻震盪問題。畫出閉迴路系統之根軌跡及單位步階響應。(15%)

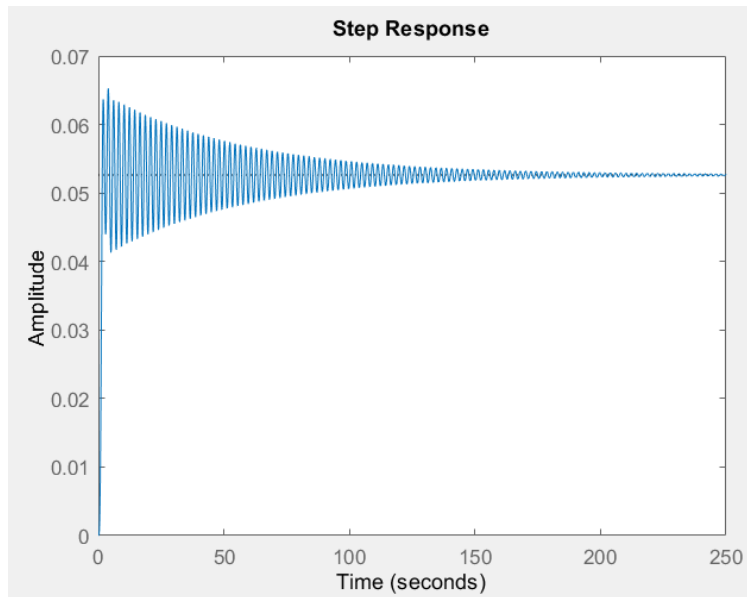


圖二 系統 B

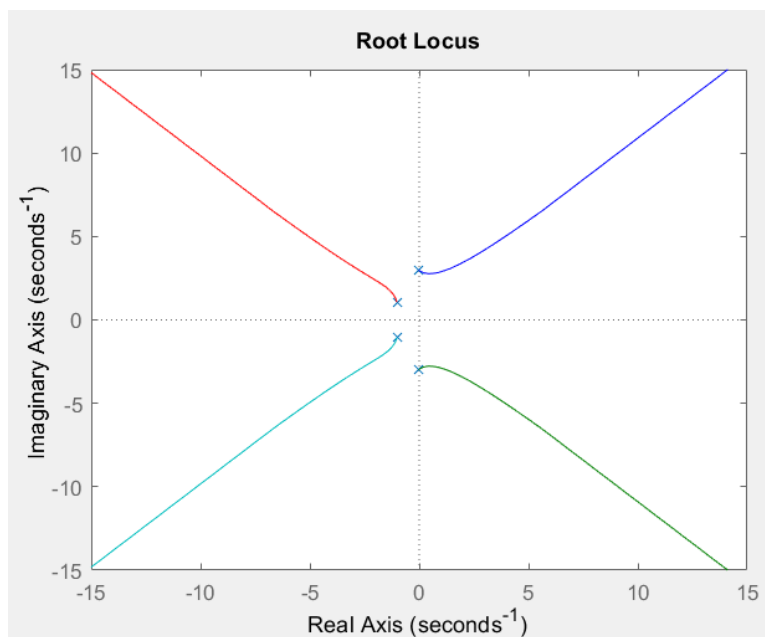


圖三 系統 C

(i) 要實現高頻振動形式，則裝置本身需要有近虛軸的複數極點作為轉移函數的一部分，因此假設 4 極點的系統中，其中兩接近虛軸的共軛極點為 $-0.03+3j$, $-0.03-3j$ ，另外兩個共軛極點設為 $-1+j$, $-1-j$ 。此時高頻振動裝置的單位步階響應圖如圖一

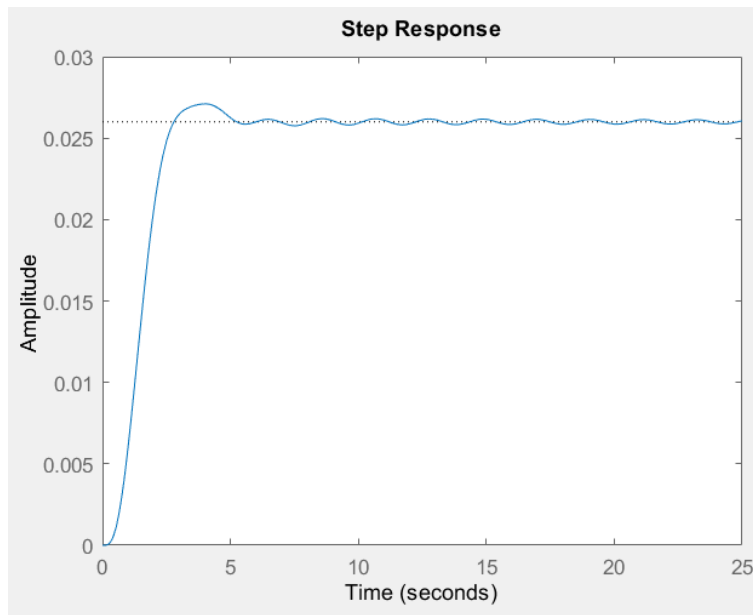


根軌跡如圖二

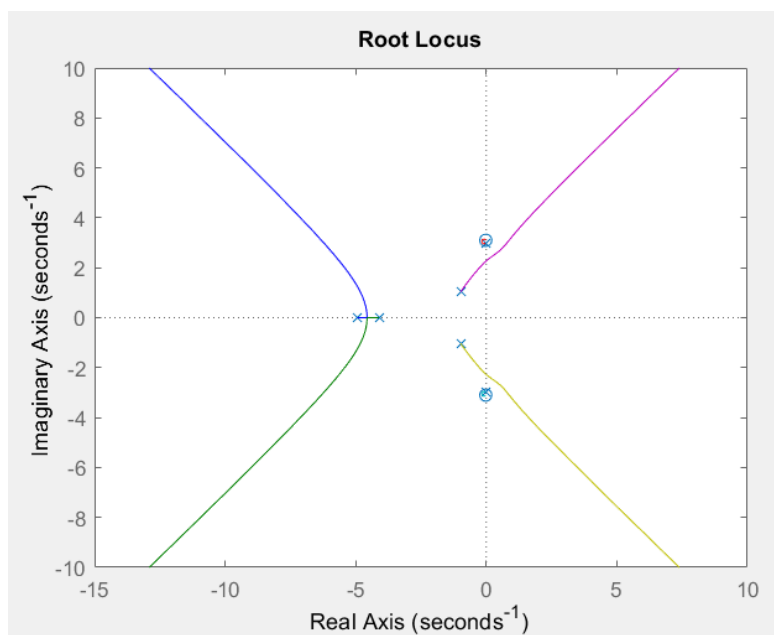


(ii)所設計的凹陷濾波器有兩實極點兩零點，其有兩個零點要靠近原裝置極點，能視為和原裝置極點互消，使裝置高頻響應可忽略，假設為 $-0.02+3.1j$ 、 $-0.02-3.1j$ ；另外兩個實極點則為 $-4, -6$ 。

互消後的單位步階響應圖如圖三



根軌跡如圖四



3.

問題3: (雷達天線定位控制問題)

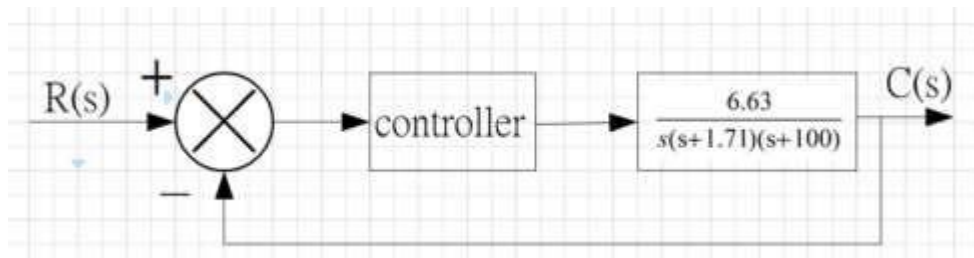
如圖四，設計一 PID 控制器，達成下列目標。(25%)

(i) 25% O.S.

(ii) $T_s = 2 \text{ sec.}$

(iii) $e(\infty) = 0$

並且以主動電路實現。(15%)



圖四 系統 D

$\%OS=25, T_s=2, e(\infty)=0$, 依據題目可算出 $\zeta=0.404$,

$\arccos(0.404)=66.2^\circ$ 。

補償主極點:

$$\sigma = \frac{4}{2} = 2$$

$$\zeta\omega_n = 4.5346$$

然後求出補償零點

$$180^\circ - \tan^{-1}\left(\frac{4.535}{2}\right) + 180^\circ - \tan^{-1}\left(\frac{4.535}{2-1.71}\right) + \tan^{-1}\left(\frac{4.535}{100-2}\right)$$

$$= 113.8^\circ + 93.65^\circ + 2.64^\circ = 210.09^\circ$$

$$= 210.09^\circ - 180^\circ = 30.09^\circ$$

$$\tan(30.09) = \frac{4.535}{Z_c - 2} \quad , \quad Z_c = -9.8$$

$$PD_{controller} = G_{PD}(s) = (s + 9.8)$$

$$PI_{controller} = G_{PI} = \frac{s + 0.5}{s}$$

$$G_{PID} = \frac{K(s + 0.5)(s + 9.8)}{s} = s + 10.3 + \frac{4.9}{s}$$

$$G_c(s) = -\left[\left(\frac{R_2}{R_1} + \frac{C_1}{C_2}\right) + R_2 C_1 s + \frac{R_1 C_2}{s}\right]$$

$$\left(\frac{R_2}{R_1} + \frac{C_1}{C_2}\right) = 10.3$$

$$R_2 C_1 = 1$$

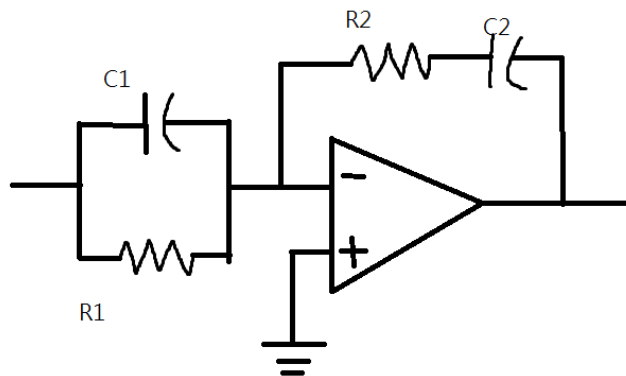
$$\frac{1}{R_1 C_2} = 4.9$$

Assume: $C_2 = 0.1 \mu F$

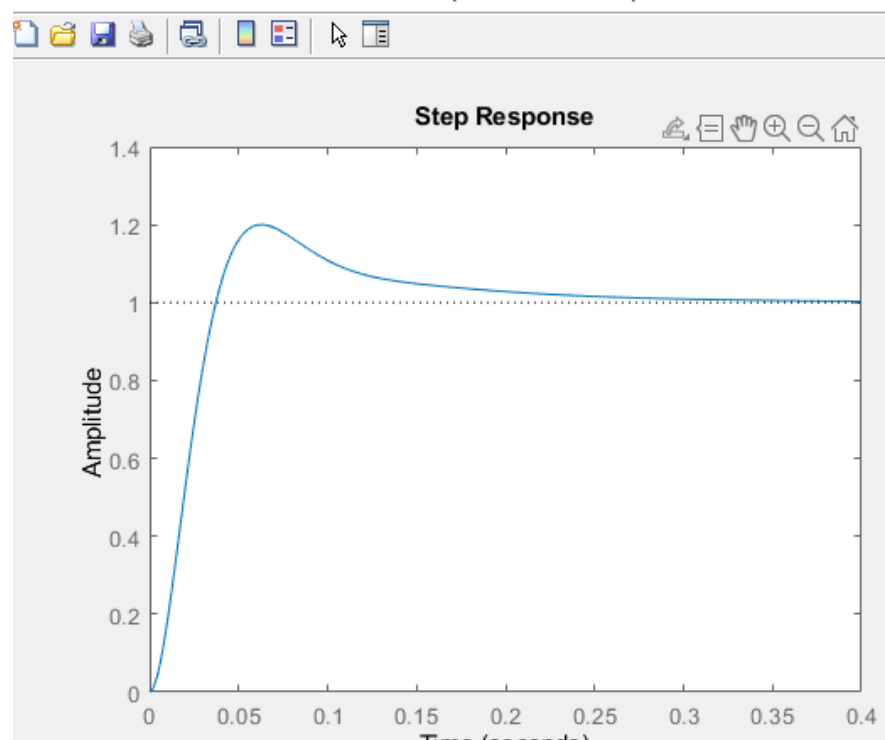
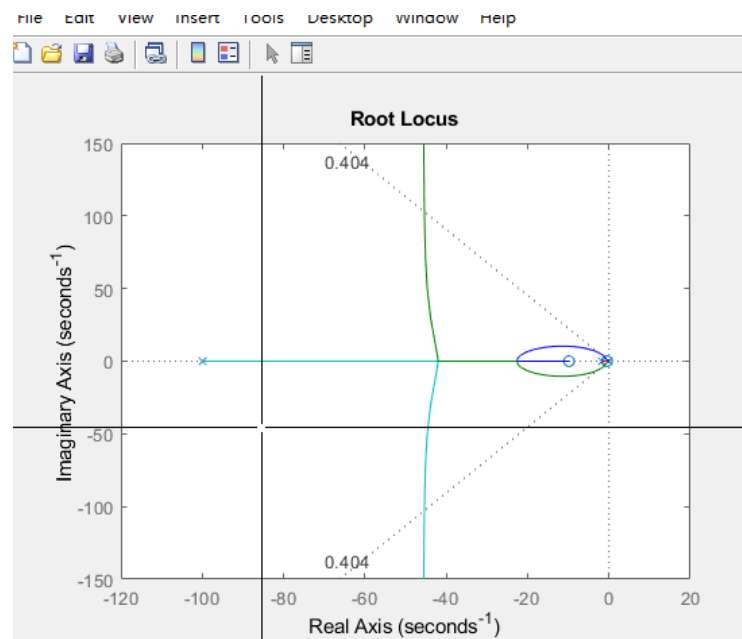
$$R_1 = 2040.8 k\Omega \quad R_1 = 2040.8 k\Omega$$

$$C_1 = 0.49 \mu F \quad C_1 = 0.49 \mu F$$

$$R_2 = 202.14 k\Omega \quad R_2 = 202.14 k\Omega$$



根軌跡



4.

作業程式碼：

%% 作業一

```
num=[32.72];           %轉移函數分子
den=[1 4.622 32.72];   %轉移函數分母
G=tf(num,den);         %轉移函數G
figure(1)              %選定圖片編號1的視窗
step(G,5)              %畫出G的步階響應圖
```

%% 作業二

```
pole=poly([-0.03+3j,-0.03-3j,-1+j,-1-j]); %%裝置極點
G1=tf(1,den);
T1=feedback(G1,1);
figure(1);
step(T1);
figure(2);
rlocus(T1);

Nzero=poly([-0.02+3.1j,-0.02-3.1j]); %%凹陷濾波器零點
Npole=poly([-4,-5]); %%凹陷濾波器極點
G2=tf(Nzero,Npole);
T2=feedback(G1*G2,1);
figure(3);
step(T2);
figure(4);
```

%% 作業 3

```
clc  
clear
```

%% 根軌跡

```
s=tf('s');  
num = [6.63]  
den = poly([0 -100 -1.71])  
Ps=tf(num,den);
```

%% Run SISO tool to analyze Plant transfer function

%% PID method

```
Ps1=Ps
```

%% Run SISO tool to analyze Plant transfer function

%% Export the PD compensator for analysis

```
PD=(s+9.82)
```

```

16
17 %% Export the PD compensator for analysis
18 PD=(s+9.82)
19 Ps2=Ps1*PD
20
21 %% Apply PD compensation
22 %% Export the PID compensator value for analysis
23 PID=(s+0.5)/s
24 Ps3=Ps2*PID
25 rlocus(Ps3)
26 sgrid(0.404,0)
27 [K,pole]=rlocfind(Ps3)
28
29 Ts3 = feedback(K*Ps3,1)
30
31 %%
32 step(Ts3)
33 %% Finally we will calculate our steady-state error (refer to equations)
34
19 Ps2=Ps1*PD
20
21 %% Apply PD compensation
22 %% Export the PID compensator value for analysis
23 PID=(s+0.5)/s
24 Ps3=Ps2*PID
25 rlocus(Ps3)
26 sgrid(0.404,0)
27 [K,pole]=rlocfind(Ps3)
28
29 Ts3 = feedback(K*Ps3,1)
30
31 %%
32 step(Ts3)
33 %% Finally we will calculate our steady-state error (refer to equations)
34
35 kp=dcgain(Ps3); % dcgain is a function that will carry out our limit s
36 ess=1/(1+kp) % this is the equation for the steady state error for a s

```