به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

پروژه سیستم های کنترل خطی

بهمن ماه ۱۴۰۳

توحید بهشتی – اشکان بهروزی

۸۱۰۱۰۰۲۷۶ – ۸۱۰۱۰۰۱۰۰

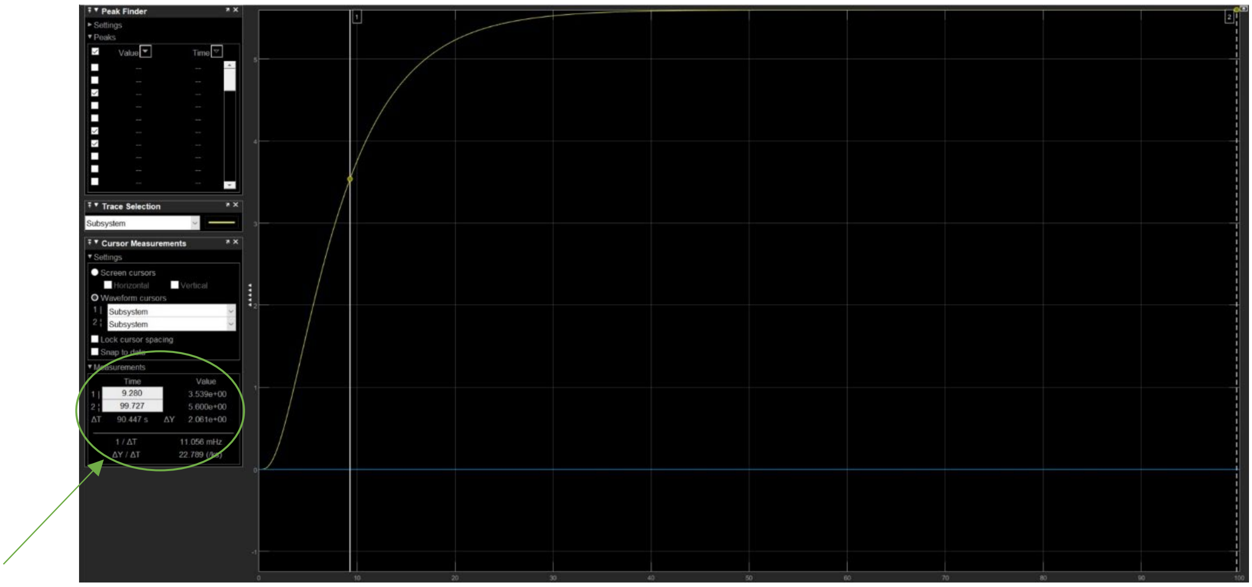
بهمن ماه ۱۴۰۳

سوال 1 :

بعد از اعمال ورودی های گفته شده در سوال و دادن مجموع یکان های شماره دانشجویی (6) :

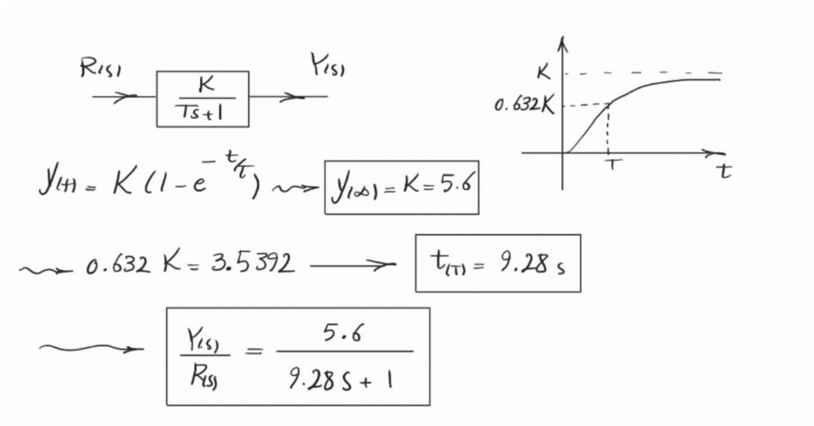
- تخمین مناسب مرتبه 1 :

شکل زیر پاسخ پله سیستم را نشان میده که با اطلاعات مشخص شده در شکل میتوان تابع مرتبه اول آن را تخمین زد.



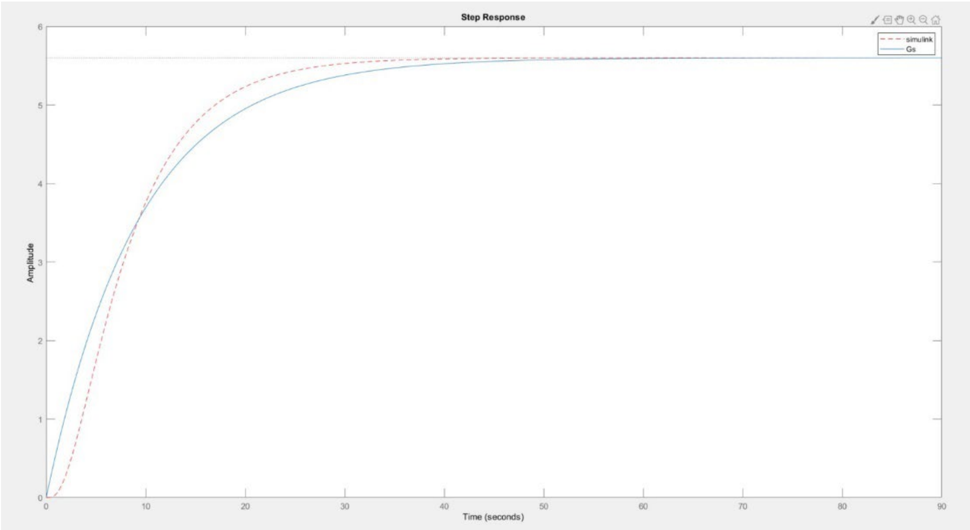
شکل ۱) پاسخ پله سیستم

با اطلاعات نشان داده شده در شکل بالا و توضیحات شکل پایین میتوان متابع مرتبه اول را بدست آورد.



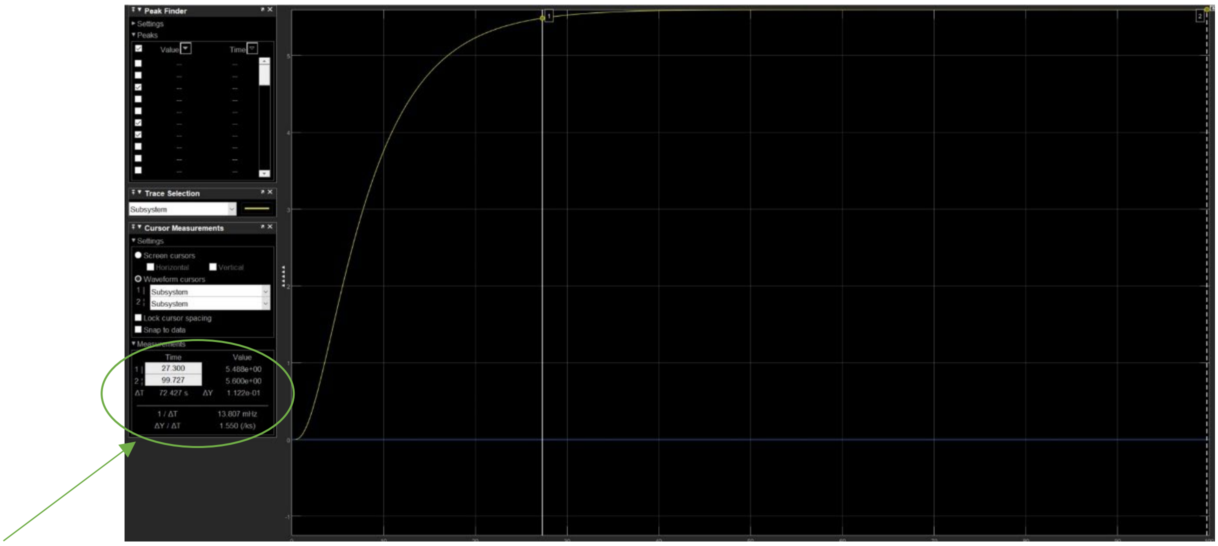
شکل ۲) توضیحات تخمین تابع مرتبه ۱

نمودار مرتبه اول را در کنار نمودار اصلی آمده از سیمولینک در شکل زیر مشاهده میکنید.



شکل ۳) مقایسه نمودار ها

- تخمین مرتبه 2 :



شکل ۴) پاسخ پله سیستم

در شکل بالا زمان ts (زمان نشست) آن در شکل مشخص شده است که با توجه به آن و توضیحات عکس زیر معادله درجه 2 را تخمین میزنیم.

$$R(s) \longrightarrow \boxed{\frac{K\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}} \longrightarrow Y(s)$$

میدانیم : $t_s = \frac{4}{\zeta\omega_n}\ (2\%)$

$0.98 \times K = 5.488 \longrightarrow$ در زمان $t_s = 27.3$ آن می‌آید

* برای $\zeta = 1$: $\omega_n = \frac{4}{1 \times 27.3} \simeq 0.1465$

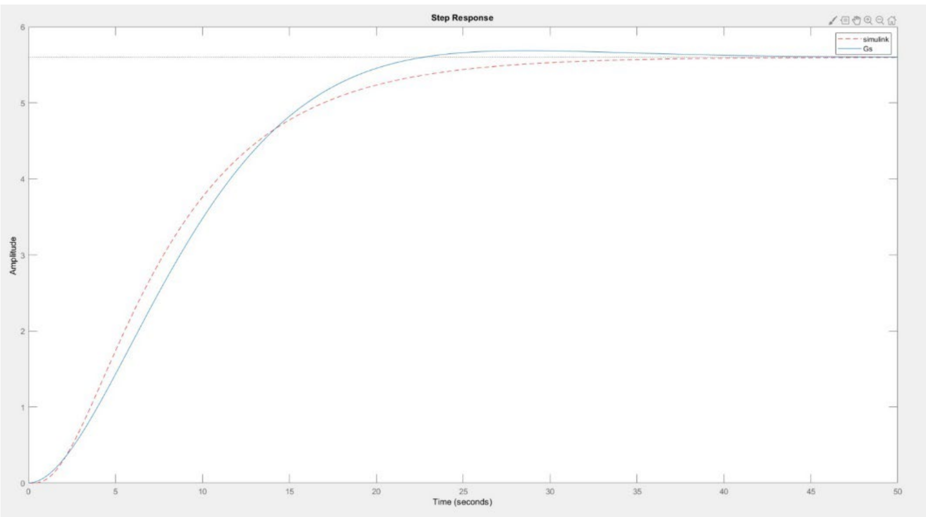
* برای $\zeta = 0.8$: $\omega_n = \frac{4}{0.8 \times 27.3} = 0.18315$

که با توجه به شبیه سازی مناسب تر است

$$\frac{Y(s)}{R(s)} = 5.6 \times \frac{(0.18315)^2}{s^2 + 0.293s + (0.18315)^{\boxed{2}}}$$

شکل ۵) توضیحات تخمین تابع مرتبه ۱

با توجه به معادلات بالا و شبیه سازی مقدار های گذاشته شده بهینه ترین حالت برای تخمین معادله درجه 2 میباشد که مقایسه دو نمودار را در شکل زیر مشاهده میکنید.



شکل ۶) مقایسه نمودار ها

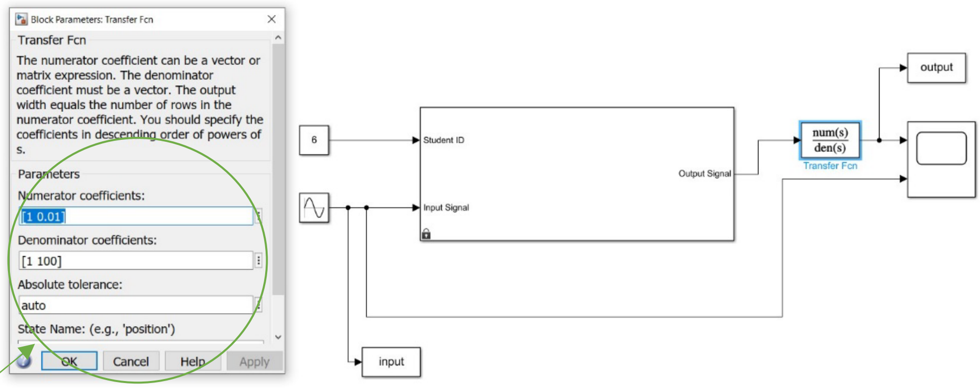
سوال 2 :

در این قسمت با اعمال ورودی های سینوسی با فرکانس های مختلف قصد داریم اندازه و فاز پاسخ سیستم را بررسی نماییم .

- در فرکانس های پایین میتوان با مقایسه تفاوت زمانی در پیک سیگنال سینوسی خروجی و ورودی و ضرب مقدار فرکانس در آن مقدار فاز را محاسبه کرد.

$$\measuredangle G_s = w * (input\ peak - output\ peak)$$

- به دلیل دیده نشدن همه پیک ها مجبور به گذاشتن یک فیلتر پایین گذر شدیم تا پیک ها را بیشتر نشان دهد. سپس با بردن داده های سیمولینک به متلب و با استفاده از دستور findpeaks در متلب همه پیک ها را پیدا میکنیم و سپس برای هر فرکانس پیک های جایی که تاثیر مقدار گذرا در آن کم است را در ورودی و خروجی از هم کم میکنیم و در نهایت اختلاف زاویه ای که فیلتر پایین گذر ایجاد کرده را هم از آن کم میکنیم تا به اخلاف فاز تابع برسیم.



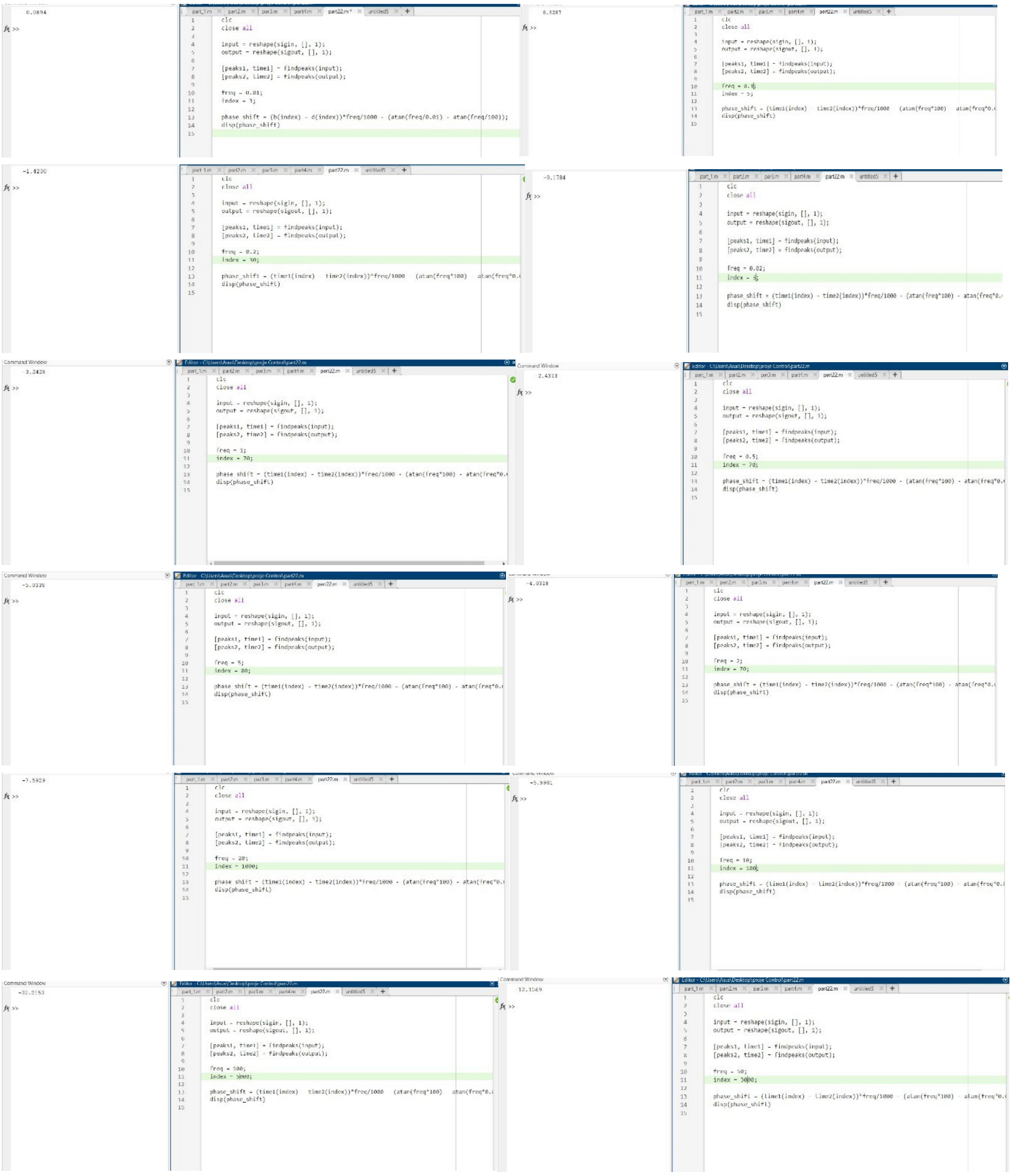
شکل ۷) فیلتر پایین گذر در سیمولینک

مشخصات فیلتر را در شکل مشاهده میکنید.

- تا امگا برابر 5 رادیان بر ثانیه، از روی شکل میشد مقدار فاز را بدون فیلتر محاسبه کرد اما برای

اما برای فرکانس های بالاتر از ان سیگنال های ورودی و خروجی را با استفاده از بلوک to workspace به محیط متلب برده و پیک های ان را استخراج کرده و از تفاضل زمانی محل پیک های حالت ماندگار و ضرب ان در فرکانس متناظر به مقدار فاز برسیم .

- باید توجه داشت که قاز تولید شده در فیلتر در هر فرکانس متفاوت است که باید آنرا لحاظ کنیم.



شکل 8) مقادیر فاز در هر فرکانس

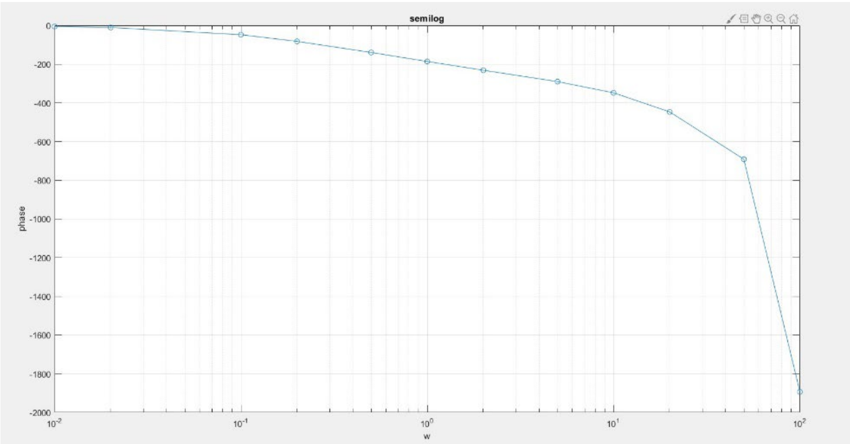
در شکل های بالا روش و مقدار اندازه گیری فاز ها را مشاهده میکنید.

- برای کشیدن نمودار اندازه و فاز کد زیر را در متلب میزنیم.

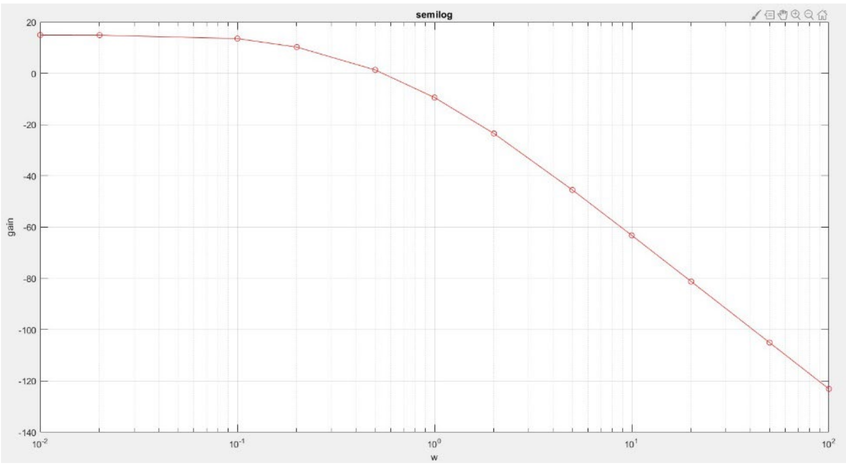
```
w = [0.01, 0.02, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100];
phase_rad = [0.08942, 0.17828, 0.8287, 1.428, 2.431, 3.2438, 4.0348, 5.0608, 6.0708, 7.7908, 12.06008, 33.0508];
phase_deg = -1 * rad2deg(phase_rad);
gain = [5.588, 5.553, 4.756, 3.253, 1.165, 0.337, 0.06708, 5.323*10^(-3), 6.909*10^(-4), 8.721*10^(-5), 5.597*10^(-6), 6.996*10^(-7)];
figure;
semilogx(w, phase_deg, '-o');
xlabel('w');
ylabel('phase');
title('semilog');
grid on;
figure
semilogx(w, 20*log10(gain), '-or');
xlabel('w');
ylabel('gain');
title('semilog');
grid on
```

شکل 9) کد متلب

- کد بالا نتایج زیر را میدهد.



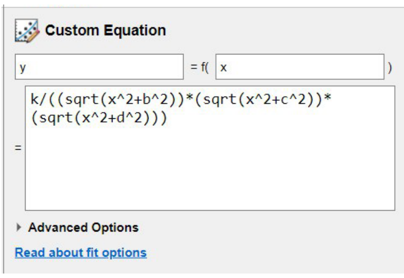
شکل 10) نمودار فاز



شکل 11) نمودار بهره

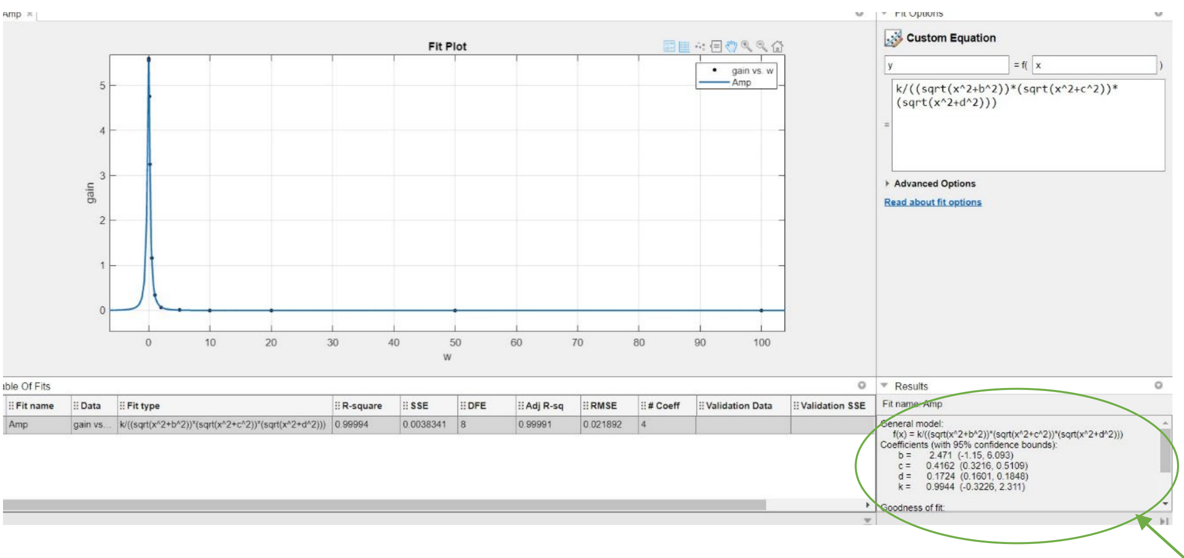
- سپس با قسمت Cftool در Custom Equation، که با استفاده از آن میخواهیم بهترین منحنی را به نمودار بدست آمده فیت کنیم.
- با دیدن مقدار گین متوجه میشویم که اختلاف درجه صورت و مخرج برابر 3 است و 3 قطب داریم.

- با دیدن نمودار فاز متوجه میشویم که در صورت جمله تاخیر(نمایی) وجود دارد. و با گذاشتن اندازه تابع تبدیل توضیح داده شده در بالا در Cftool بهترین نمودار را به آن فیت میکند که در شکل زیر مشاهده میکنید.



شکل 12) تابع وارد شده در Cftool

- سپس فیت میکنیم.



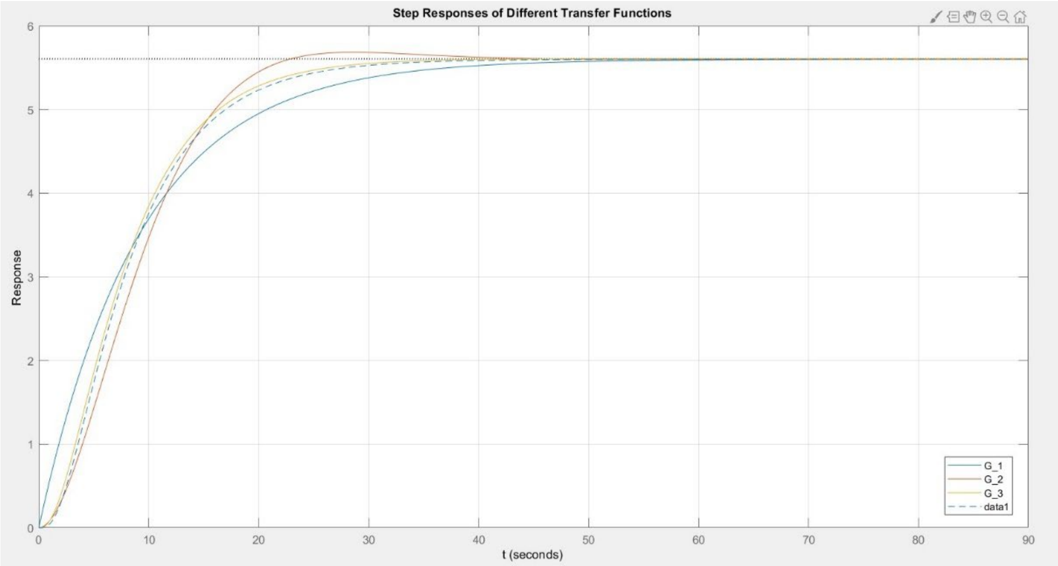
شکل 13) خروجی Cftool

مقادیر نهایی تابع تبدیل در شکل بالا مشخص شده اند که به ما تابع تبدیل G3 را میدهد.

سوال 3 :

تابع تبدیل سیستم های G1 و G2 در قسمت بخش ۱ طراحی شد و همچنین سیستم G3 در بخش قبل به وسیله curve fitting در Cftool متلب طراحی شد و تابع تبدیل ان مشخص گردید. حالا هر چهار نمودار خواسته شده را در یک figure میکشیم و سپس مشخصات زمانی حالت گذرا و ماندگار ان را را با هم مقایسه میکنیم.

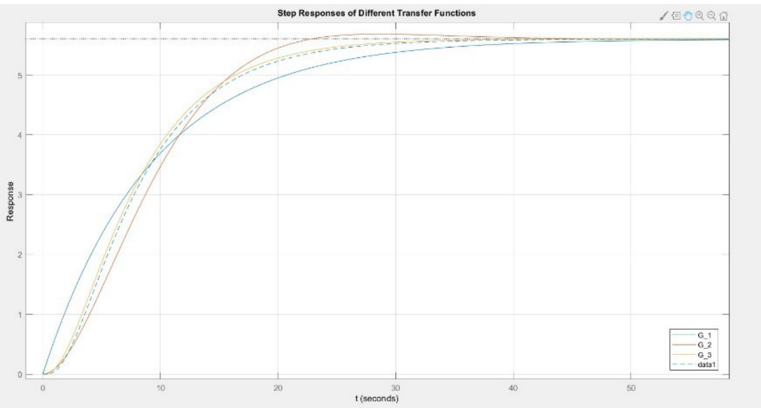
- نتایج همه تابع تبدیل های بدست آمده را در یک نمودار میکشیم که به شکل زیر میرسیم.



شکل 14) هر 4 نمودار کنار هم

نمودار خط چین از محیط سیمولینک آورده شده و رسم شده است. همانطور که مشاهده می شود، نمودار زرد رنگ که با Cftool به دست آمده از همه نمودار های دیگر به نمودار اصلی نزدیک تر است.

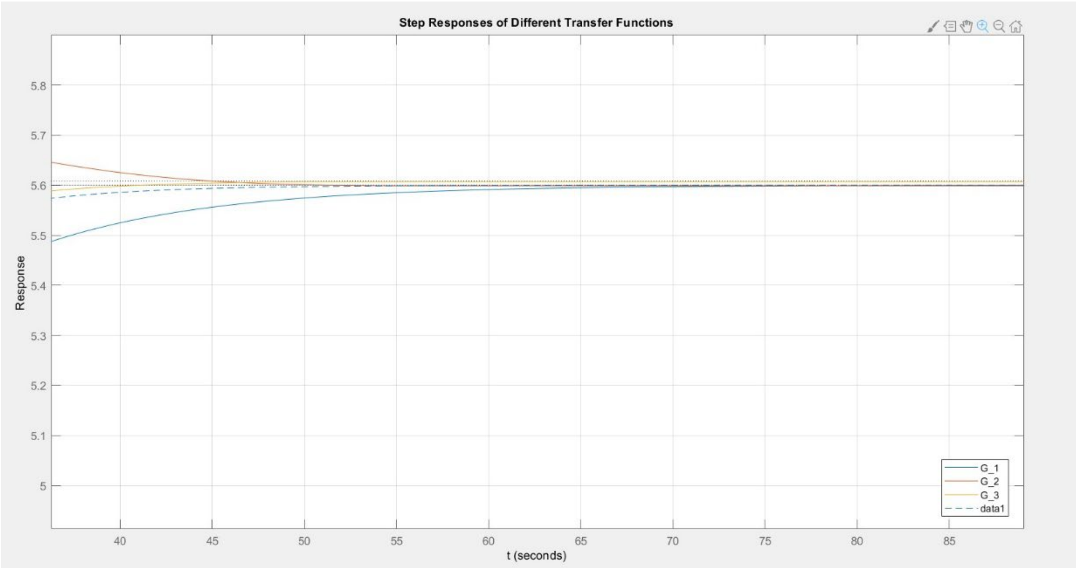
- اگر بخواهیم حالت گذرا را در نظر بگیریم:



با توجه به شکل، زمان فراجهش G2 از بقیه بیشت است و مقدار فراجهش بیشتری دارد.

و برای G1 از همه کمتر است.

- اگر حالت ماندگار را در نظر بگیریم:



همانطور که مشاهده میشود، زمان نشست G1 بیشتر از حالت های دیگر است. و G3 و نمودار اصلی زمان نشست کمتری دارند. و زمان نشست نمودار G2 از G1 کمتر است.

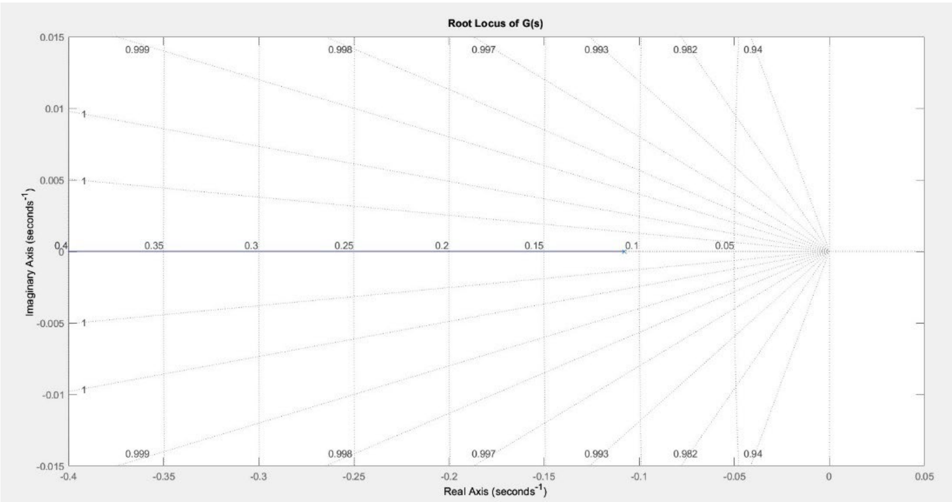
سوال 4 :

در این قسمت توابع تبدیل محاسبه شده را در مسیر مستقیم یک حلقه فیدبک واحد منفی قرار داده و نمودار مکان ریشه انها را رسم کرده تا بازه ای برحسب k (بهره سیستم) برای پایداری سیستم ها به دست بیاوریم .

- سیستم G1 : همانطور که از نمودار مکان ریشه ان مشخص است این سیستم به ازای همه مقادیر K مثبت پایداراست .
- سیستم G2 : این سیستم نیز با توجه به محل قرارگیری قطب ها و محل مکان ریشه ان به ازای تمامی مقادیر k پایدار است .
- سیستم G3 : همانطور که از نمودار مکان ریشه ان مشخص است با افزایش k از مقدار مشخصی میتوان دید که قطب های سیستم سمت راست میرود و سیستم ناپایدار میشود .

مکان ریشه هر کدام را در شکل های پایین مشاهده میکنید.

- مکان ریشه G1 :

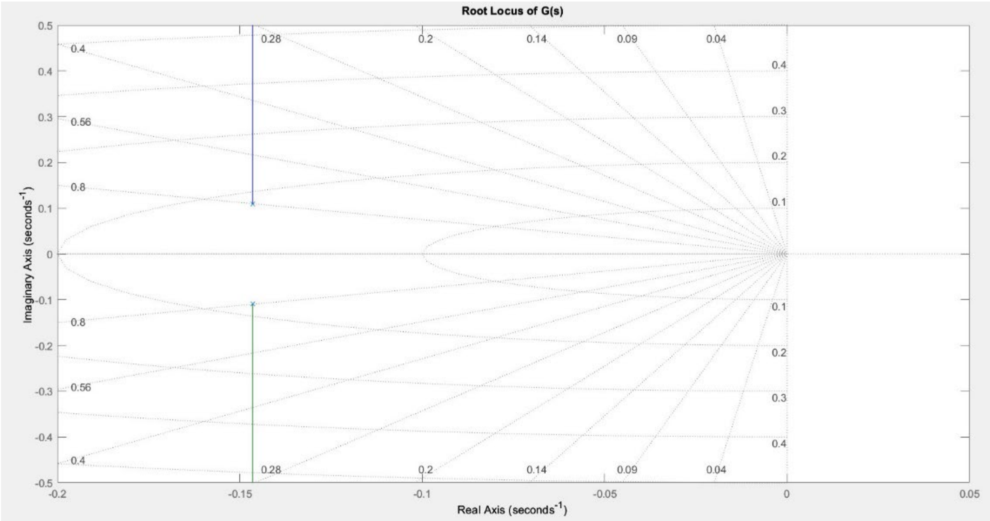


شکل 15) مکان ریشه G1

همانطور که در شکل مشخص است به ازای $k > 0$ پایدار خواهد بود.

برای K های منفی هم زده شد بازه پایین آن $K < -0.18$ میباشد.

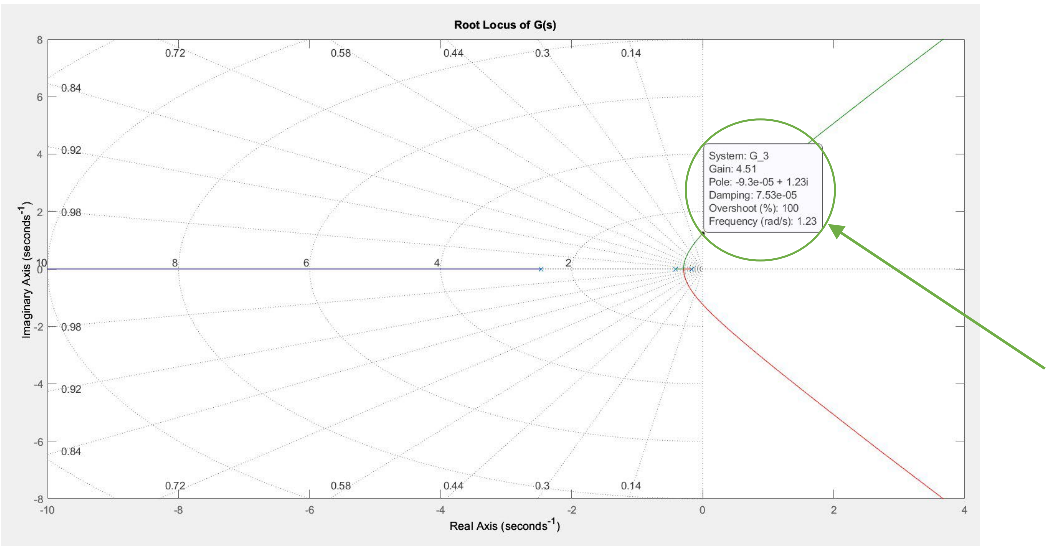
- مکان ریشه G2 :



شکل 16) مکان ریشه G2

همانطور که در شکل مشخص است به ازای تمامی مقادیر k پایدار است.

- مکان ریشه G3 :

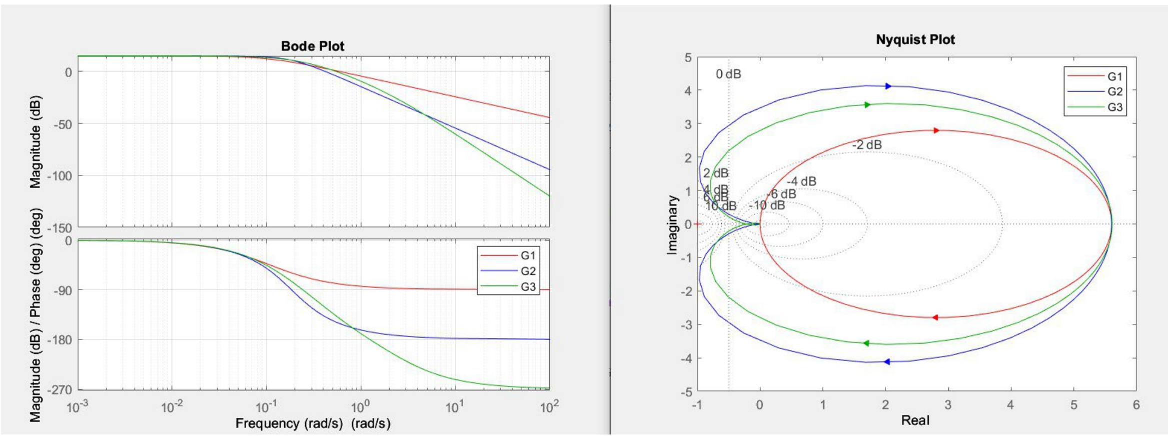


شکل 17) مکان ریشه G3

همانطور که در شکل مشخص است به ازای k > 4.51 پایدار خواهد بود.

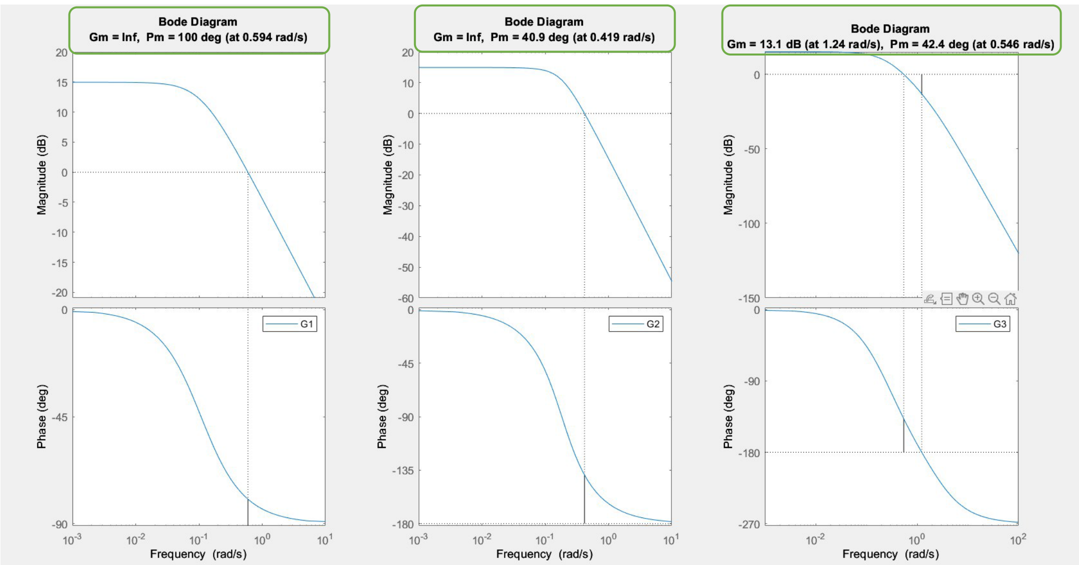
سوال 5 :

با استفاده از متلب نمودار بود و نایکوییست هر کدام را در یک نمودار رسم میکنیم که میتوانید آن ها را در شکل زیر مشاهده کنید.



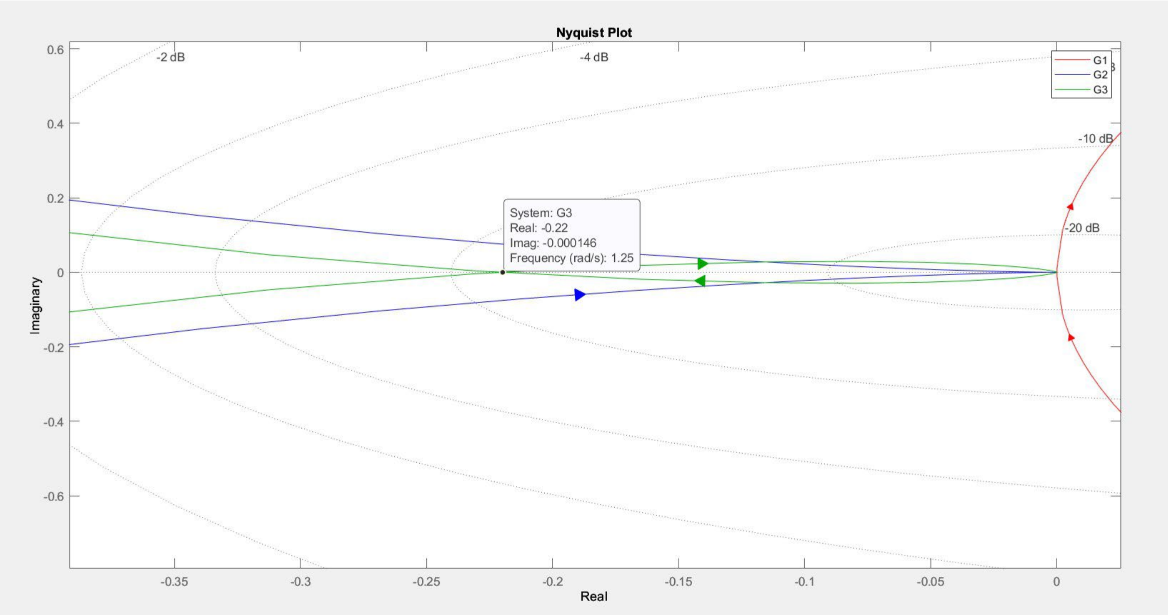
شکل 18) نمودرا های بود و نایکوییست

- حال با استفاده از متلب و نمودار های بود مقادیر PM و GM را بدست می آوریم.



شکل 19) حاشیه بهره و فاز نمودار بود

- مقادیر خواسته شده در بالای هر نمودار نوشته شده است.



شکل 20) مقادیر روی نایکوییست

از نمودار نایکوییست G1 و G2 میتوان فهمید که GM بینهایت است و مقادیر حاشیه فاز هم مانند قسمت های قبل میشود.

برای G3، مکانی را که در زاویه 180- قطع میکند را اندازه میگیریم و همانطور که در درس خواندیم، مقدار GM برابر 1 تقسیم بر اندازه آن در زاویه 180- است.

$$GM = \frac{1}{0.22} = 4.545$$

که تقریبا با مقادیر بدست آمده با روش های دیگر برابر است.

سوال 6 :

با استفاده از تکنیک هایی که در کلاس تدریس شده برای G1-3 کنترل کننده طراحی میکنیم.

- کنترل کنندG1 :

با توجه به اینکه گفته شده خطای حالت ماندگار باید صفر باشد، میتوان فهمید که یک قطب باید در مبدا داشته باشیم.

با گرفتن O.S = 3% و ts = 9s مقدار $\xi = 0.744804$ و $cos^{-1}(\xi) = 41.8578\ deg$ و $\sigma > \frac{4}{9}$

با توجه محلی که میخواهیم مکان ریشه ما از آن بگذرد

کنترل کنندد را به صورت زیر تبدیل میکنیم :

$$Gc1 = Kc\,.\frac{s+z}{s}$$

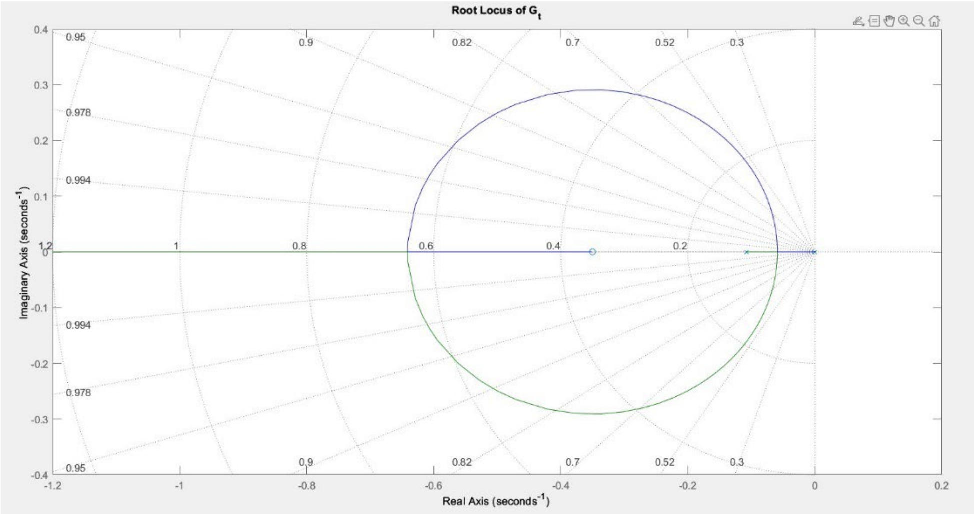
سپس با محاسبه نقص زاویه (120.923) محل مناسب صفر را پیدا میکنیم که برابر z = 0.350242 و با استفاده از شرط اندازه مقدار K را محاسبه میکنیم که برابر K = 11.4674 میباشد.

- با زدن کد زیر در متلب مکان ریشه و پاسخ پله سیستم را با کنترل کننده طراحی شده میکشیم.

```
G1 = tf(5.6, [9.28, 1]);  % G1 = 5.6 / (9.28s + 1)

G_c = tf([11.46, 11.46 * 0.35], [1, 0]);  % G_c = (11.46s + 11.46*0.35) / s

G_t = G1 * G_c;

G_f = feedback(G_t, 1);

% Step Response
figure;
step(G_f);
grid on;
title('Step Response of the System');
xlabel('Time (seconds)');
ylabel('Response');

% Root Locus
figure;
rlocus(G_t);
grid on;
title('Root Locus of G_t');
xlabel('Real Axis');
ylabel('Imaginary Axis');
```

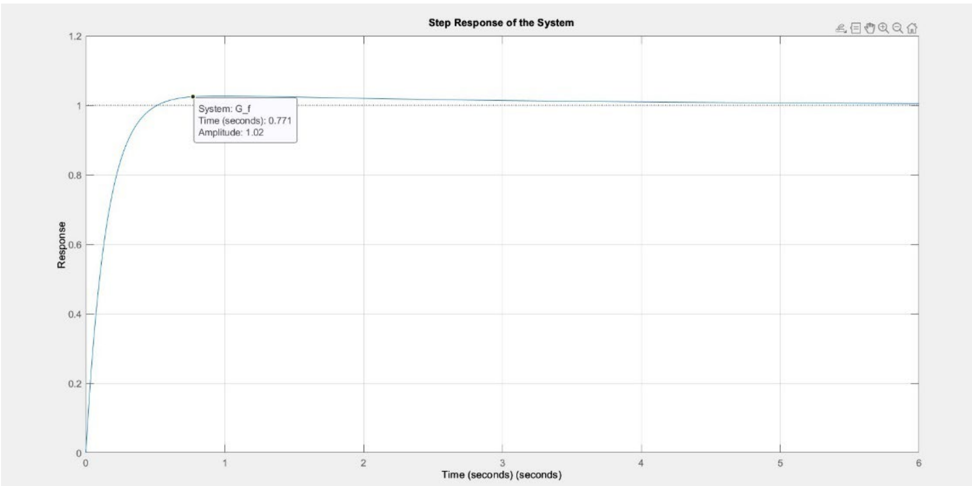
شکل 21) کد برای G1

- مکان ریشه سیستم طراحی شده:



شکل 22) مکان ریشه

- پاسخ پله سیستم :



شکل 23) پاسخ پله سیستم طراحی شده

شرط فراجهش هم براورده شده است.

- کنترل کننده G2 :

در این حالت با گرفتن کنترل کننده به صورت شکل زیر :

$$Gc2 = K\,.s + z_1\,.\frac{s + z_2}{s}$$

که جمله $s + z$ برای جبران نقص زاویه است و جمله $\frac{s+z}{s}$ برای درست کردن شرط صفر شدن حالت ماندگار است.

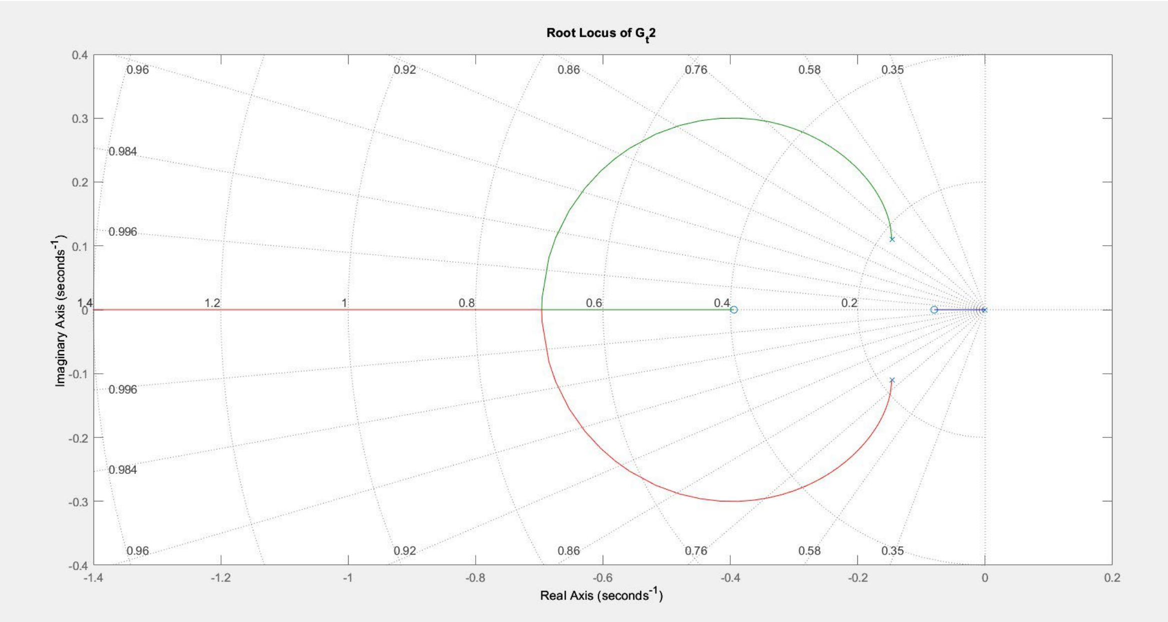
با نوشتن محاسبات نقص زاویه، مقدار $z_1 = 0.3943$ و $z_2 = 0.09$ بدست می‌آید و با نوشتن شرط اندازه، مقدار K بدست می آید که برابر K = 2.89 میشود.

سپس با زدن کد متلب زیر میتوان مکتن ریشه و پاسخ پله آن را محاسبه کرد.

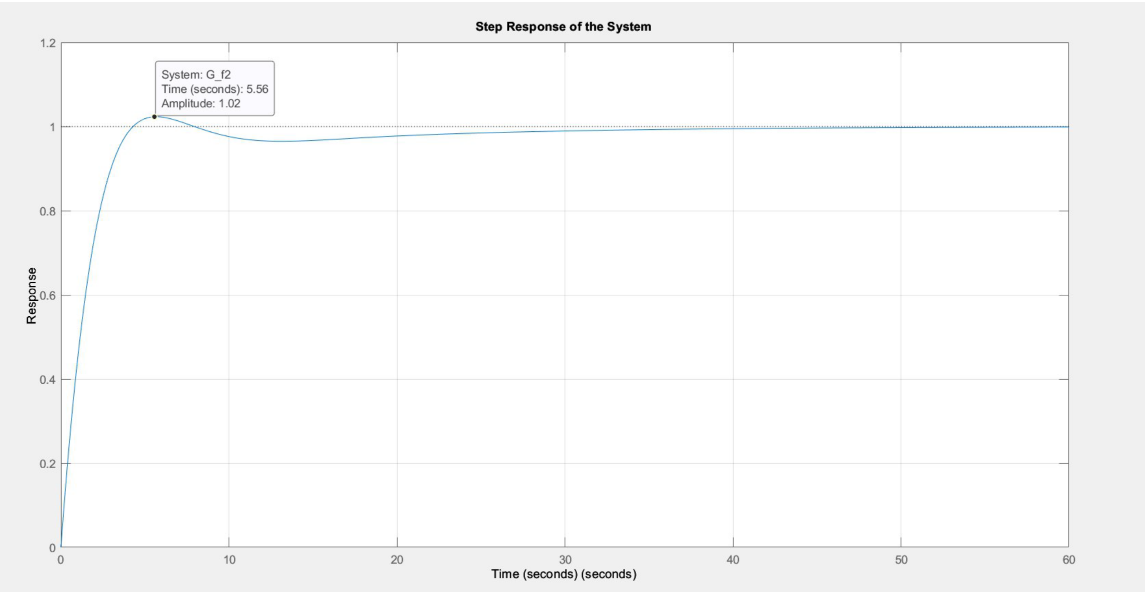
```
G_2 = 0.1876 * tf(1, [1, 0.293, 0.0335]);

G_c2 = 2.89 * tf(conv([1 0.3943], [1 0.08]), [1, 0]);
%G_c2 = 1.4326 * tf([1 0.4842], [1, 0]);
G_t2 = G_2 * G_c2;
G_f2 = feedback(G_t2, 1);

% Step Response
figure;
step(G_f2);
grid on;
title('Step Response of the System');
xlabel('Time (seconds)');
ylabel('Response');

% Root Locus
figure;
rlocus(G_t2);
grid on;
title('Root Locus of G_t2');
xlabel('Real Axis');
ylabel('Imaginary Axis');
```

شکل 24) کد برای G2

• مکان ریشه :



شکل 25) مکان ریشه



شکل 26) پاسخ پله سیستم طراحی شده

همانطور که در شکل مشاهده میشود شرط فراجهش هم براورده شده است.

- کنترل کننده G3 :

در این حالت هم مانند قسمت قبل از کنترل کننده زیر استفاده میکنیم.

$$Gc3 = K\,.\,s + z_1\,.\frac{s + z_2}{s}$$

با محاسبه نقص زاویه و در نظر گرفتن شرط اینکه خطای حالت ماندگار برابر صفر باشد، $z_1 = 0.5139$

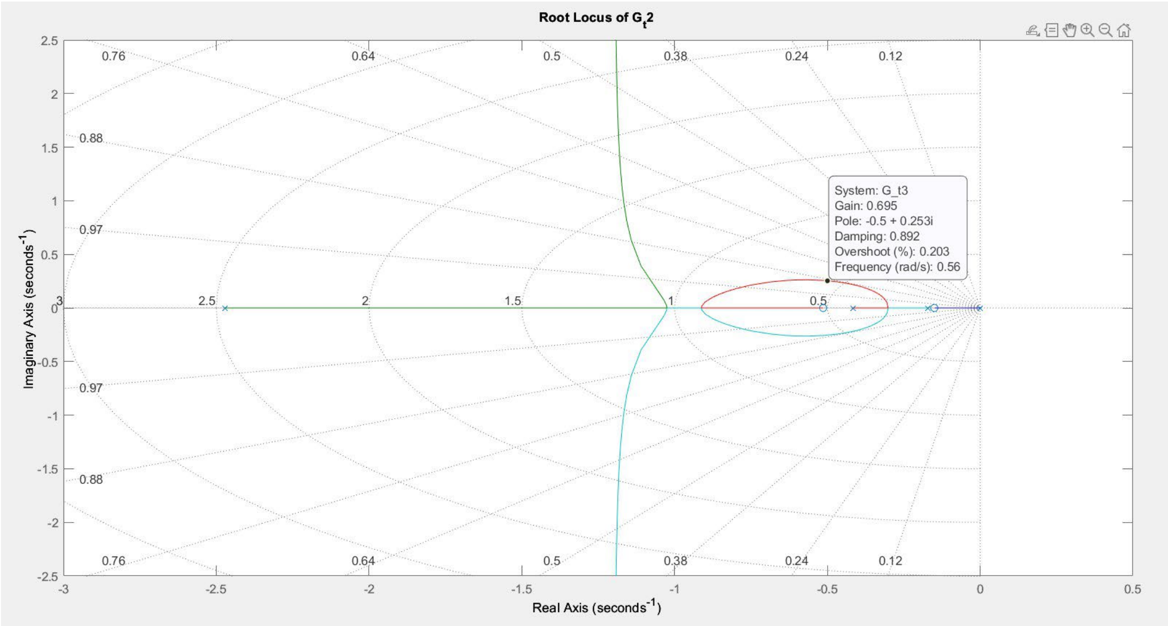
$z_2 = 0.15$, و $K = 1.622$ میشود.

$$Gc3 = 1.622\,.\,s + 0.5139\,.\frac{s + 0.15}{s}$$

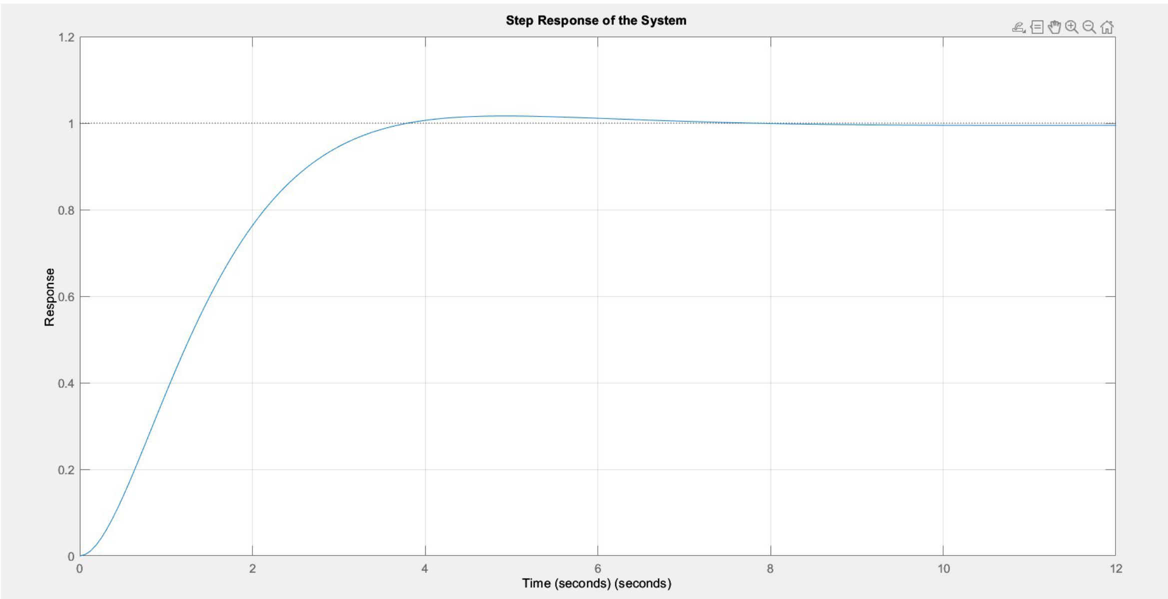
سپس کد متلب زیر را برای رسم مکان ریشه و پاسخ پله میزنیم.

```
G_3 = 0.9944 * tf(1, conv([1 2.471], conv([1 0.4162], [1 0.1724])));

G_c3 = 1.622 * tf(conv([1 0.5139], [1 0.15]), [1, 0]);

G_t3 = G_3 * G_c3;
G_f3 = feedback(G_t3, 1);

% Step Response
figure;
step(G_f3);
grid on;
title('Step Response of the System');
xlabel('Time (seconds)');
ylabel('Response');

% Root Locus
figure;
rlocus(G_t3);
grid on;
title('Root Locus of G_t2');
xlabel('Real Axis');
ylabel('Imaginary Axis');
```

شکل 27) کد برای G3



شکل 28) مکان ریشه

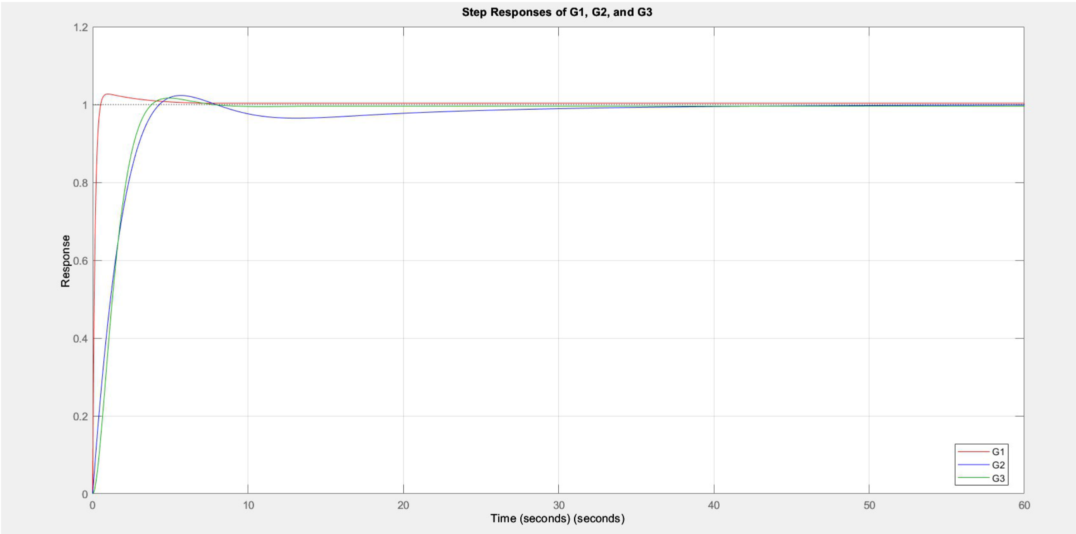


شکل 29) پاسخ پله سیستم طراحی شده

مقدار فراجهش هم برابر 2 درصد درآمد که شرط فراجهش را برآورده میکند.

سوال 7 :

سپس همه پاسخ پله های بدست آمده در قسمت قبل را روی یک نمودار می اندازیم تا حالت گذرا و ماندگار آنرا بررسی کنیم.



شکل 30) پاسخ پله هر 3 سیستم با هم

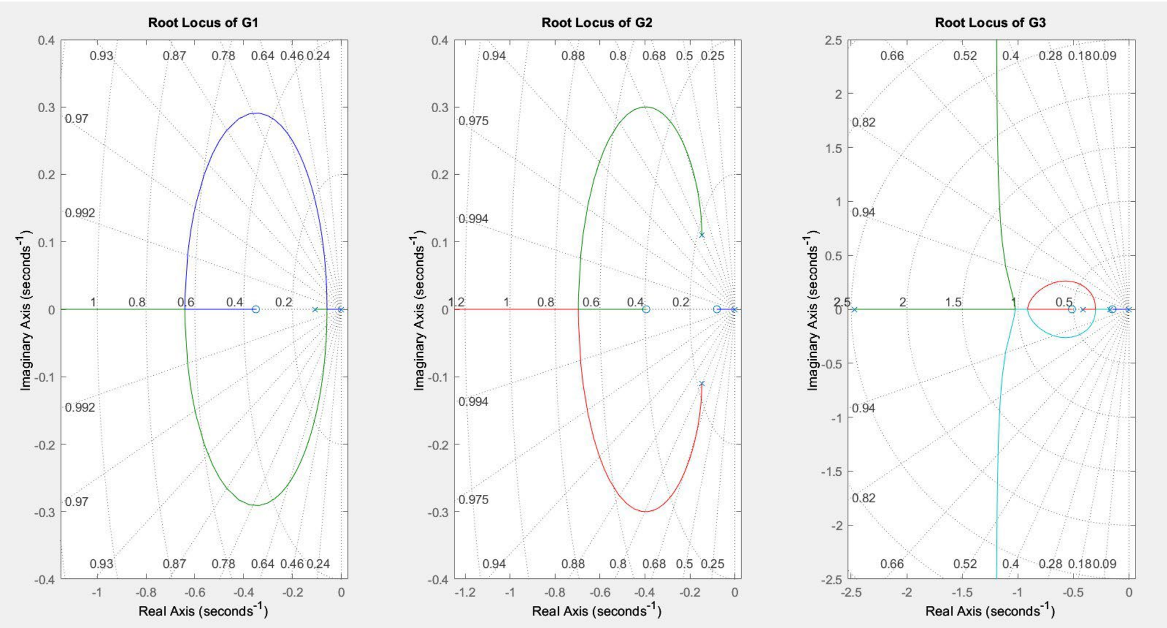
با توجه به شکل بالا، زمان فراجهش در G1 از همه کمتر است و زمان فراجهش G2 از همه حالات دیگر بیشتر است.

در مورد زمان نشست، نمودار G1 زمان نشست کمتری دارد و بعد آن G3 زمان نشست کمتری نسبت به G2 دارد. پس G2 از همه زمان نشست بیشتری دارد.

سوال 8 :

همان مراحل قسمت 4 و 5 را برای این بخش با کنترلر های طراحی شده تکرار میکنیم.

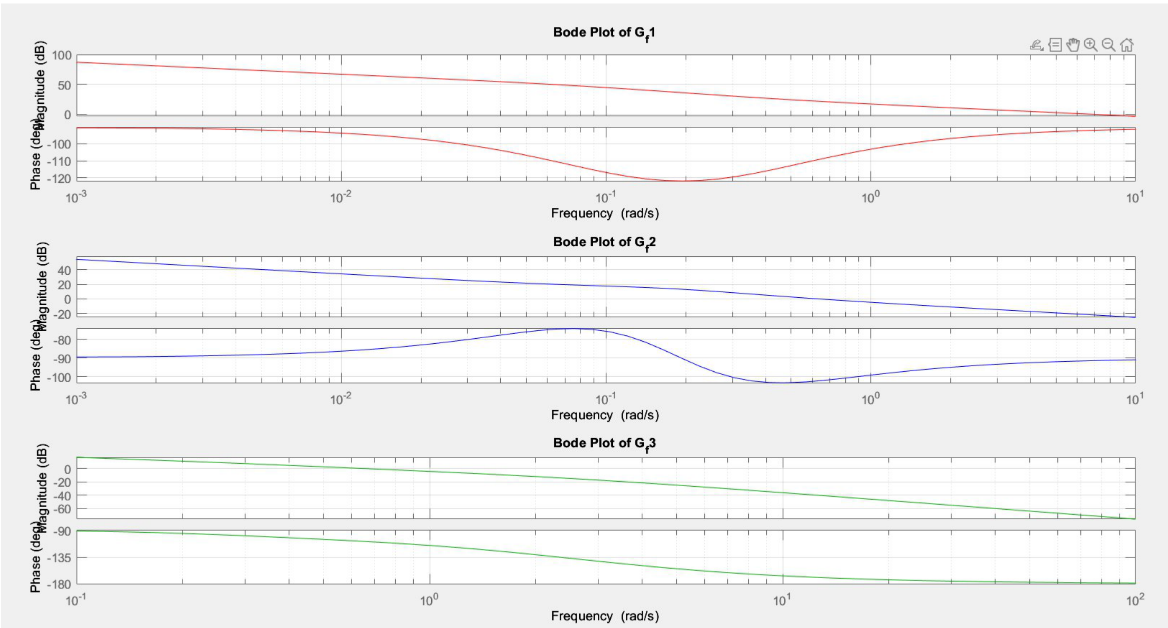
- مکان ریشه هر کدام را در شکل زیر مشاهده میکنید.



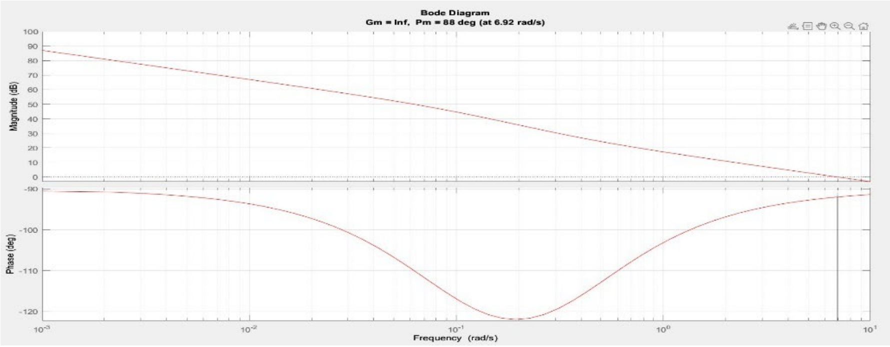
شکل 31) مکان ریشه های هر سه تابع تبدیل

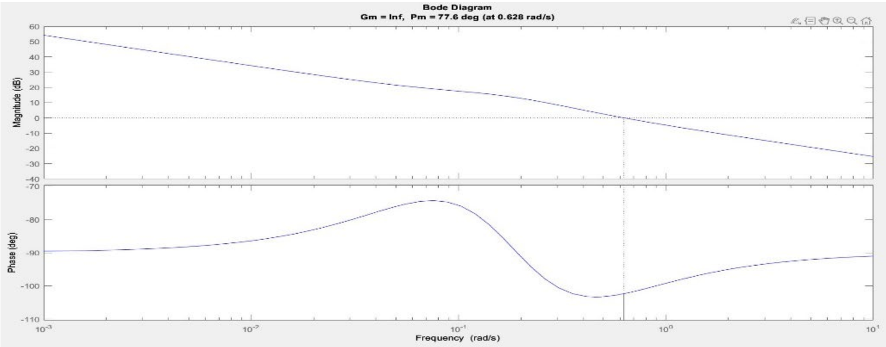
از روی مکان ریشه میتوان فهمید که به ازای همه K ها پایدار هستند.

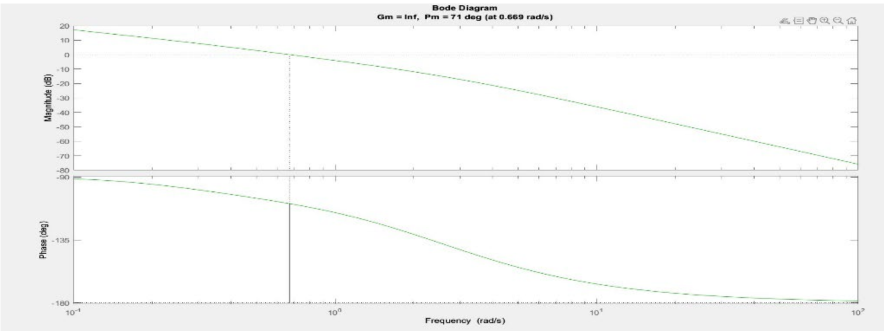
- نمودار بود برای هر 3 حالت :



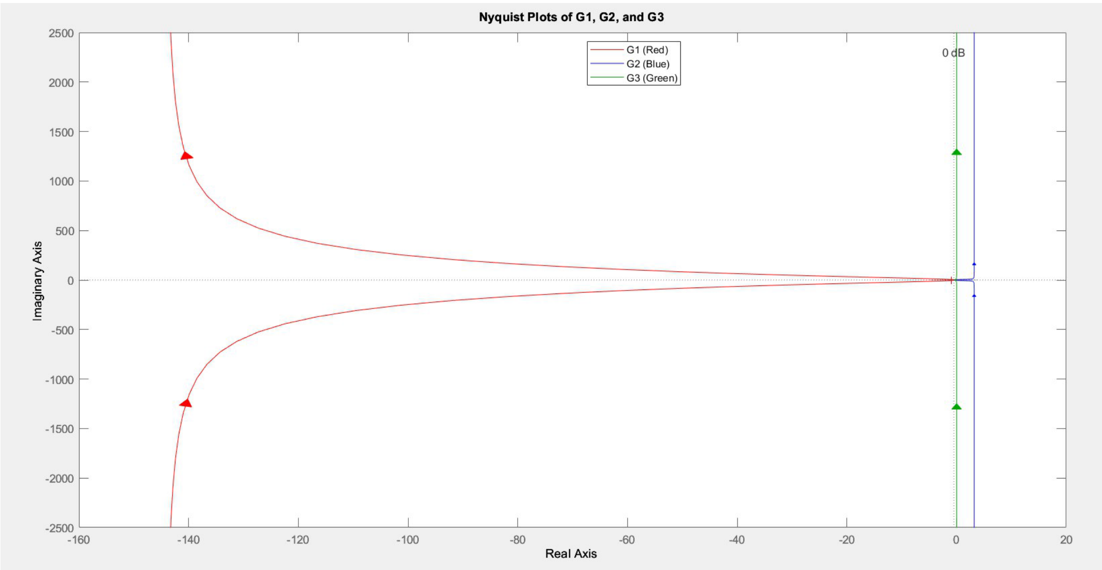
شکل 32) نمودار بود های هر 3 سیستم







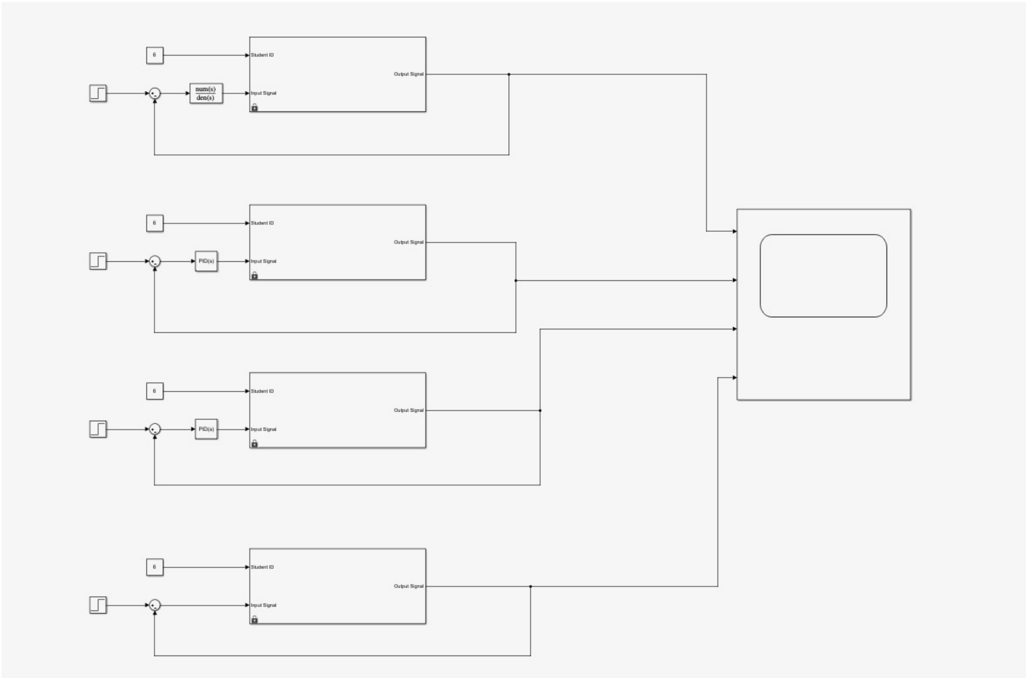
- نایکوییست برای هر 3 سیستم :



شکل 33) نایکوییست برای هر 3 سیستم

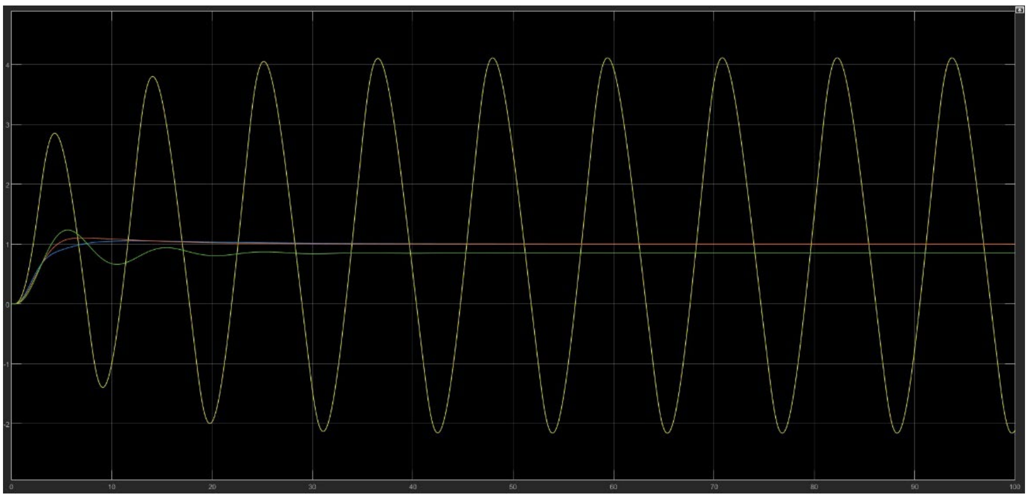
سوال 9 :

در محیط سیمولینک هر 3 سیستم که بدست آوردیم وارد میکنیم و با پاسخ پله اصلی (در فیدبک منفی). در سیمولینک شکل زیر را میبندیم.



شکل 34) شکل سیمولینک

سپس ران میکنیم و نتیجه را بررسی میکنیم.

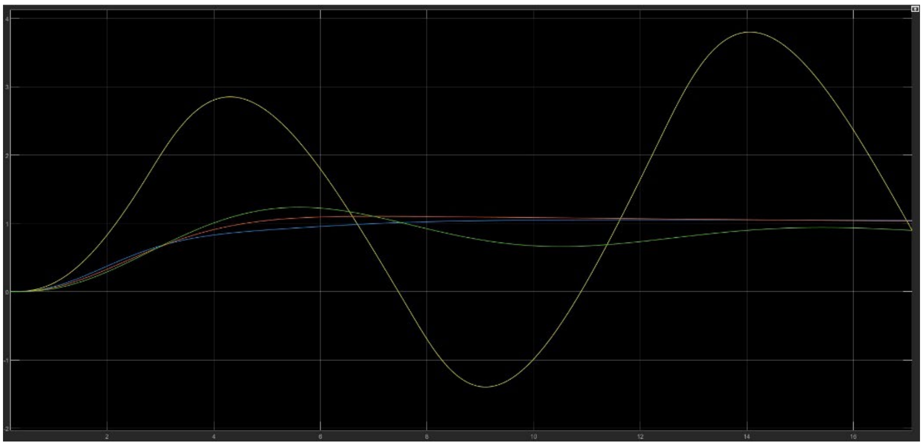


شکل 35) نمودار خروجی

همانطور که در شکل بالا دیده میشود، نمودار زرد برای G1 میباشد که به این شکل درامده است.

نمودار سبز برای سیستم بدون کنترل کننده است، که همانطور که دیده میشود جفت نمودار های قرمز و آبی به نمودار اصلی شبیه هستند، اما نمودار قرمز به آن نزدیک تر است و پاسخش مناسب تر است.

که پاسخ قرمز همان سیستم طراحی شده G3 میباشد.



شکل 36) شکل خروجی زوم شده

سوال 10:

در این بخش با استفاده از G3، سعی میکنیم در حوزه فرکانس برای بهترین تخمین یک کنترل کننده طراحی کینم.

با توجه به اینکه میدانیم، $\xi = 0.744804$ و اینکه $PM = 100 . \xi$، PM برابر مقدار :

$$PM = 100 * 0.744804 = 74.48$$

- با تغییر تابع تبدیل، کاری میکنیم تا نتایج به دلخواه ما تبدیل شود و کنترل کننده مناسب را طراحی میکنیم.

سپس با نوشتن تئوری زیر کنترل کننده را طراحی میکنیم.

$$K_v \simeq 10 \quad , \quad PM_{desired} = 74.48^\circ$$

$$G_{c(s)} = K \cdot \frac{Ts+1}{\alpha Ts+1} \rightarrow K \times \frac{0.9944}{0.1773} = 10$$

$$G_{3(s)} = 0.9944 \times \frac{e^{-t/\tau}}{(s+2.471)(s+0.4162)(s+0.1724)}$$

$$\rightarrow \boxed{K \simeq 1.783} \quad , \quad PM_u = 42.4^\circ$$

8.915

$$PM_{desired} - PM_u + 10 = \boxed{\phi_M = 42.28^\circ}$$

$$\rightarrow \boxed{\alpha = \frac{1-\sin\phi_m}{1+\sin\phi_M} = 0.1956}$$

$$|KG_{(s)}|\big|_{\omega_g} = 1 \rightarrow 1.783 \times \frac{0.9944}{\sqrt{\omega_g^2+(2.471)^2}\sqrt{\omega_g^2+(0.4162)^2}\sqrt{\omega_g^2+0.1724^2}}$$

$$\omega_g = 0.7665 \simeq \omega_m$$

$$T = \frac{1}{\omega_m\sqrt{\alpha}} = \frac{1}{0.766 \times \sqrt{0.1956}} \simeq 2.95$$

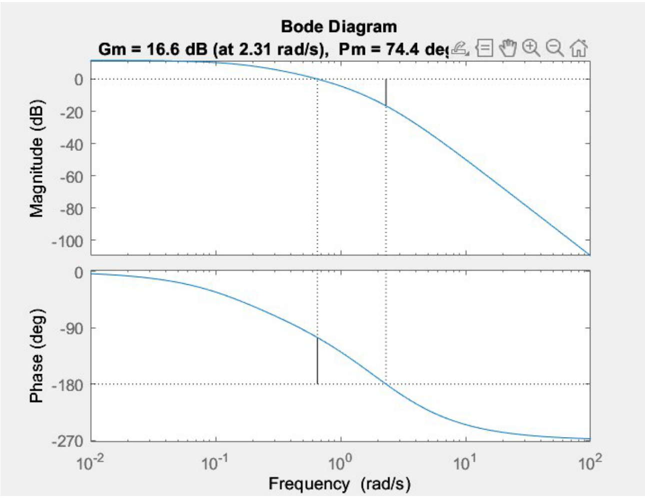
$$\rightarrow \boxed{G_{c(s)} = 0.7665 \times \frac{(2.95s+1)}{(0.1956 \times 2.95)s+1}}$$

شکل 37) تئوری کنترل کننده

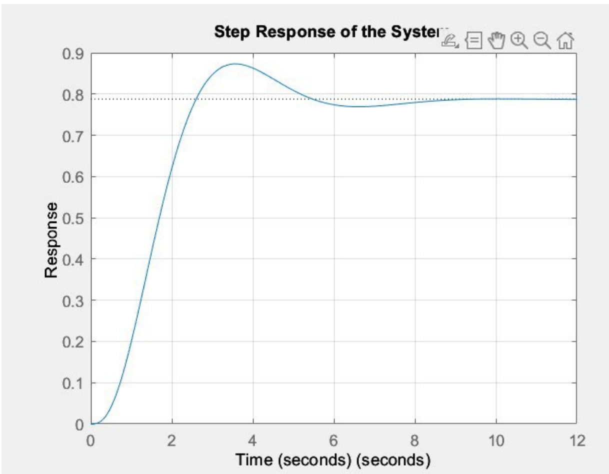
- سپس در متلب کد زیر را میزنیم :

```
G_3 = 0.9944 * tf(1, conv([1 2.471], conv([1 0.4162], [1 0.1724])));

G_c3 = 0.662 * tf([2.95, 1], [0.577, 1]);

G_t3 = G_3 * G_c3;
G_f3 = feedback(G_t3, 1);

% Step Response
figure;
step(G_f3);
grid on;
title('Step Response of the System');
xlabel('Time (seconds)');
ylabel('Response');

% Root Locus
figure;
rlocus(G_t3);
grid on;
title('Root Locus of G_t2');
xlabel('Real Axis');
ylabel('Imaginary Axis');


figure;
margin(G_t3);
```

شکل 38) کد متلب



شکل 39) نمودار بود



شکل 40) پاسخ پله سیستم

پاسخ پله بالا تقریبا با اصلی یکسان است اما در زمان بینهایت به مقدار حدود 0.8 میل کرده که کمی با پاسخ پله اصلی متفاوت است و خطای ماندگار دارد.