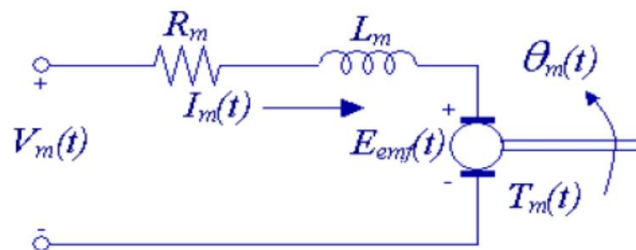


نام آزمایش: کنترل موقعیت موتور (Servo motor)

هدف: کنترل موتور با استفاده از جبران ساز PD و Lead و PI

شرح آزمایش:

در این آزمایش، تمام طراحی‌ها برای سروو موتوری انجام می‌گیرد که تصویر 1، شماتیک الکتریکی مدار آرمیچر آن را نشان می‌دهد:



1) شماتیک موتور

تابع تبدیل این موتور، به نحوی که در دستور کار آمده، محاسبه شده و در اختیار ما قرار گرفت.

$$G(s) = \frac{\theta_l(s)}{V_m(s)} = \frac{A_m}{s(J_{eq}s + B_{eq})} \quad ; \quad \begin{cases} A_m = 0.129 \\ J_{eq} = 0.00213 \\ B_{eq} = 0.0844 \end{cases}$$

#### • طراحی کنترل کننده PD

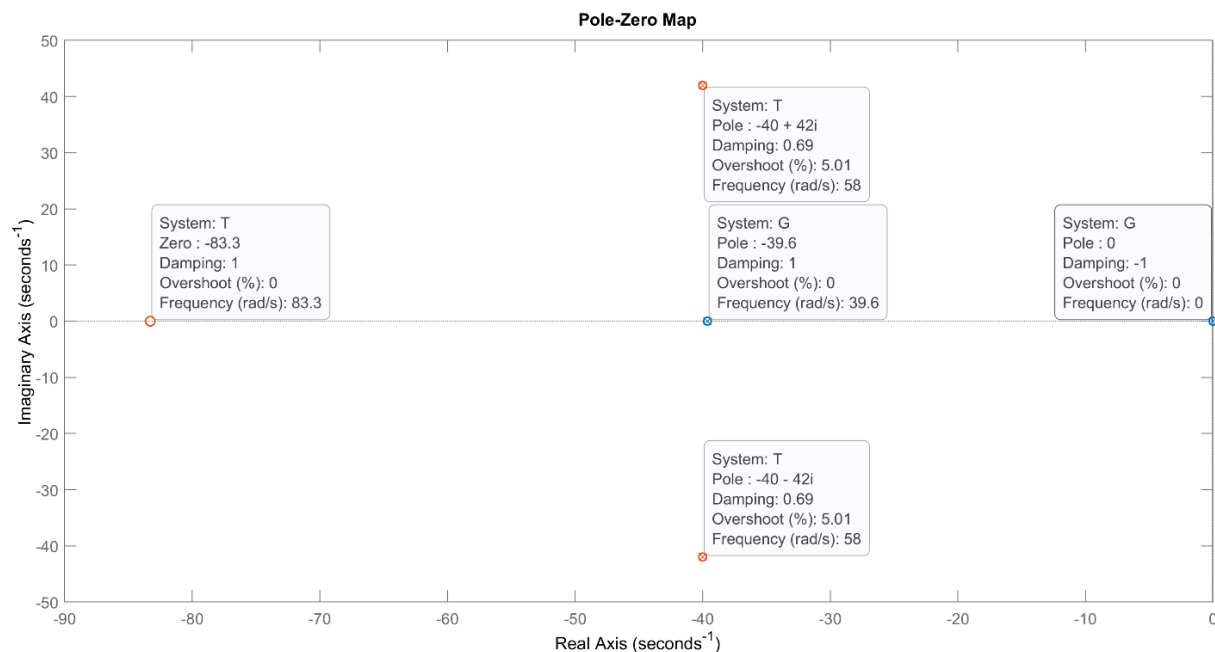
در این بخش از آزمایش قصد داریم کنترل سروو موتور را توسط کنترل کننده PD انجام دهیم. شرایط مطلوب خواسته شده از ما برای این سیستم، بالازدگی (Overshoot) کمتر از 5% و زمان نشست (Settling Time) 0.1 ثانیه می‌باشد. (زمان نشست با معیار قرار گرفتن در بازه 2% مقدار نهایی در نظر گرفته شده است).

$$P.O.V = 5 = 100e^{-\pi \frac{\xi}{\sqrt{1-\xi^2}}} \Rightarrow \xi = 0.689$$

$$t_s = 0.1 = \frac{4}{\xi \omega_n} \Rightarrow \omega_n = 57.99$$

$$\omega_d = \omega_n \sqrt{1 - \xi^2} = 41.987$$

$$s_d = -\xi \omega_n \pm j \omega_n \sqrt{1 - \xi^2} \Rightarrow s_d = -40 \pm 41.987j \rightarrow z = 40$$



(2) نمایش همزمان صفرها و قطب‌های سیستم جبران نشده و جبران شده توسط کنترل کننده PD

با کمک مقادیر به دست آمده، به ترسیم مکان ریشه تابع تبدیل کنترل نشده می‌پردازیم. باتوجه به اینکه از ما خواسته شده کنترل کننده PD طراحی کنیم، پس انتظار داریم مکان از قطب‌های مطلوب عبور نکند زیرا اگر عبور می‌کرد، طراحی کنترل کننده P کافی بود.

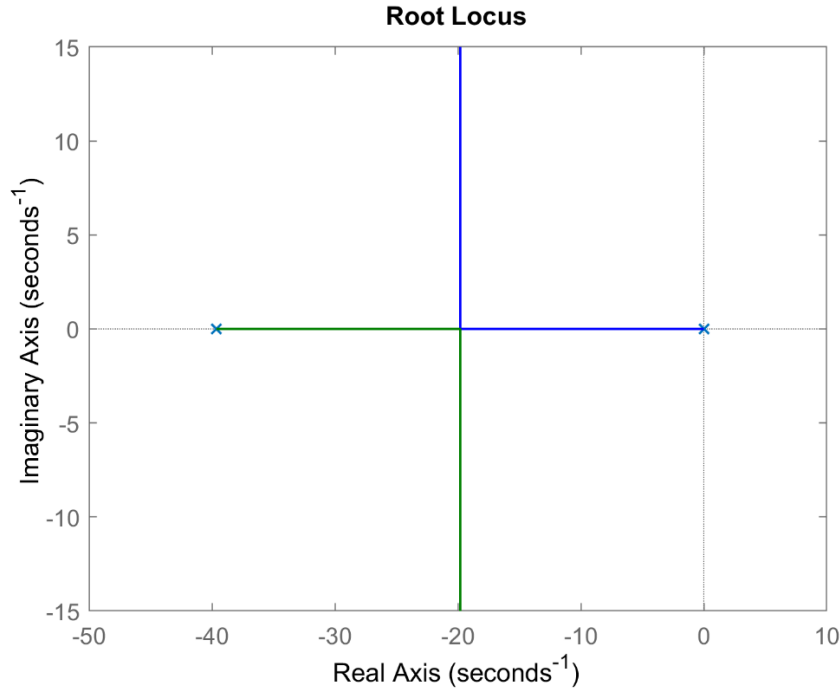
$$\Delta(s) = 0 \rightarrow s_1 = 0, \quad s_2 = -39.62$$

مکان مجانب:

$$\sigma = \frac{\sum P_i - \sum Z_i}{RD} = \frac{\sum P_i - \sum Z_i}{n - m} = \frac{0 - 39.62}{2} = -19.81$$

زاویه مجانب:

$$\varphi = \frac{(2k+1)\pi}{n-m} = \frac{(2k+1)\pi}{2} = \frac{\pi}{2}, \frac{3\pi}{2}, \dots$$



3) مکان ریشه سیستم جبران نشده

تابع تبدیل کنترل کنند PD در حالت کلی به شکل زیر است:

$$C_{PD}(s) = k(s + z) = k_d s + k_p$$

که Z محل صفر کنترل کننده و k بهره آن است و باید آن‌ها را محاسبه کنیم. برای پیدا کردن محل Z باید شرط زاویه بررسی شود:

$$(2k + 1)\pi = \sum \theta_{Zeros} - \sum \theta_{Poles} \Rightarrow (2k + 1)\pi = \theta_z - (90.51 + 133.61)$$

$$\Rightarrow \theta_z \cong 44.11^\circ \Rightarrow \tan \theta_z = \frac{41.98}{z - 40} = 0.9697 \Rightarrow z = 83.29$$

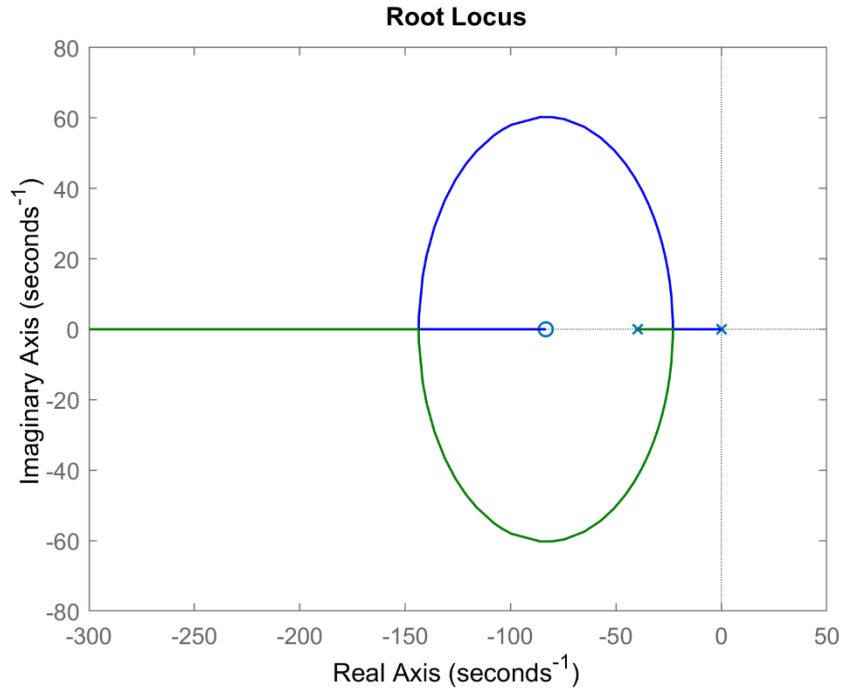
حال برای پیدا کردن مقدار k، شرط اندازه را بررسی می کنیم:

$$|C_{PD}(s_d)G(s_d)| = 1 \Rightarrow |k(s_d + z)G(s_d)| = 1$$

$$\Rightarrow k = 0.6666 \Rightarrow k_d = 0.6666, \quad k_p = 55.621$$

در نتیجه محاسبات بالا برای کنترل کننده PD خود به تابع تبدیل زیر می‌رسیم:

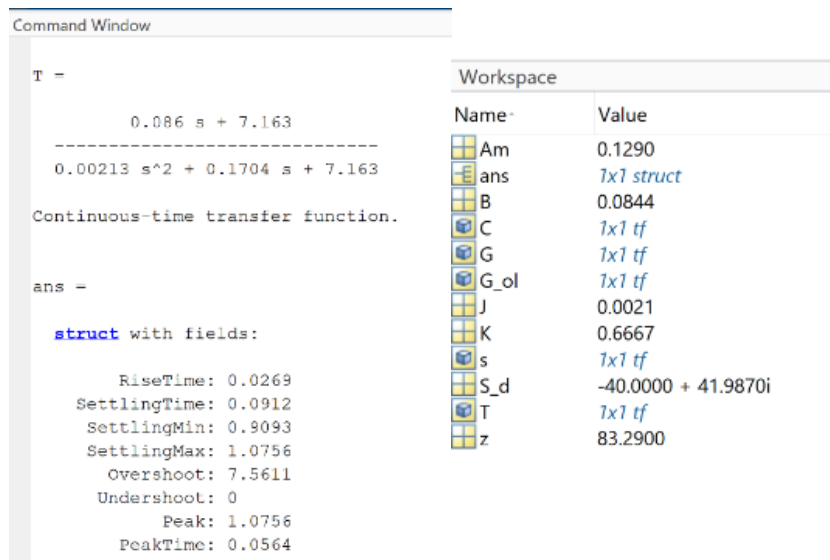
$$C_{PD}(s) = 0.6666s + 55.621$$



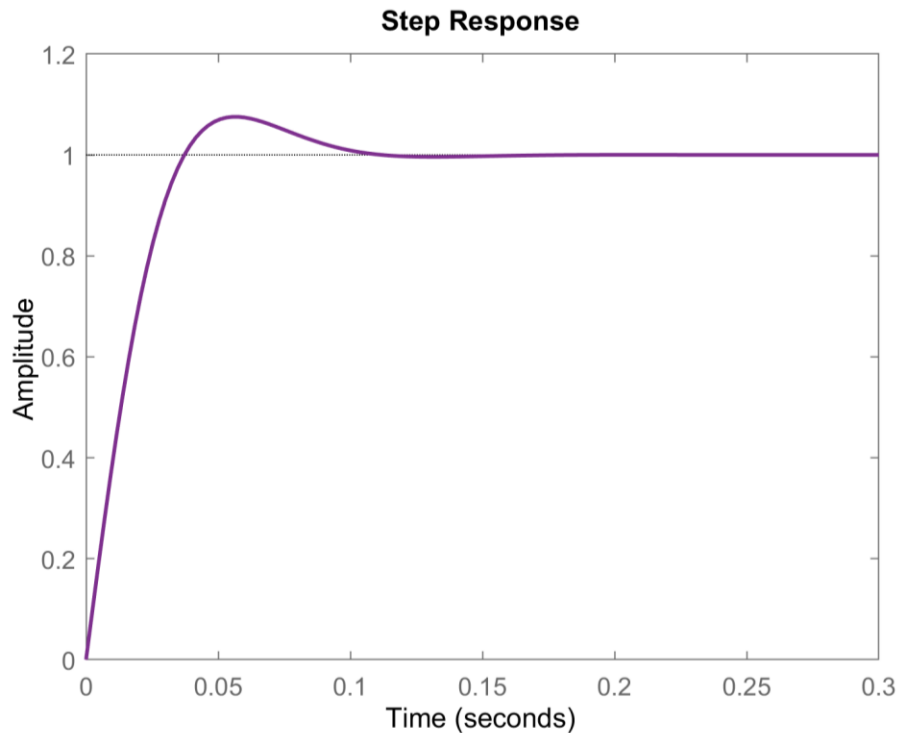
(4) مکان ریشه سیستم حلقه باز جبران شده توسط کنترل کننده PD

این جبران‌ساز به صورت سری با تابع تبدیل سروو موتور قرار گرفته و پس از گرفتن فیدبک منفی، پاسخ سیستم جدید به ورودی پله بررسی شد. شبیه‌سازی با نرم افزار Matlab انجام گرفت که در نتیجه آن برای Overshoot و Settling time مقادیر زیر حاصل شد:

$$t_s = 0.0912 \quad \text{و} \quad P.O.V = \%7.5611$$



(5) داده‌ها و نتایج شبیه سازی پس از اعمال کنترل کننده PD



(6) پاسخ پله سیستم جبران شده توسط کنترل کننده PD

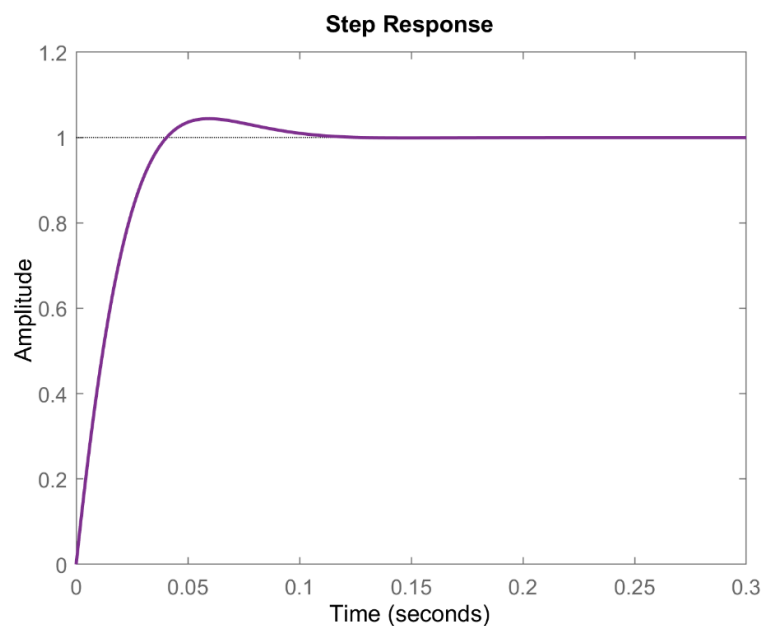
مشاهده می شود که بالازدگی مطلوب ما حاصل نشده است. علت این است که شرایط، به ازای سیستم غیر خطی داده شده اما ما با کنترل کننده خطی سیستم را طراحی کردیم. برای رفع این مشکل، با تغییر  $k$  یا  $z$  محاسبه شده، مطلوب مسئله حاصل خواهد شد. ما مقدار  $z$  را به 60.29 تغییر داده و به نتایج مطلوب زیر رسیدیم:

$$k_d = 0.8263$$

$$k_p = 53.55$$

$$t_s = 0.0877$$

$$P.O.V = \%4.4219$$



(7) پاسخ پله سیستم جبران شده توسط کنترل کننده PD/اصلاح شده

## • طراحی کنترل کننده Lead

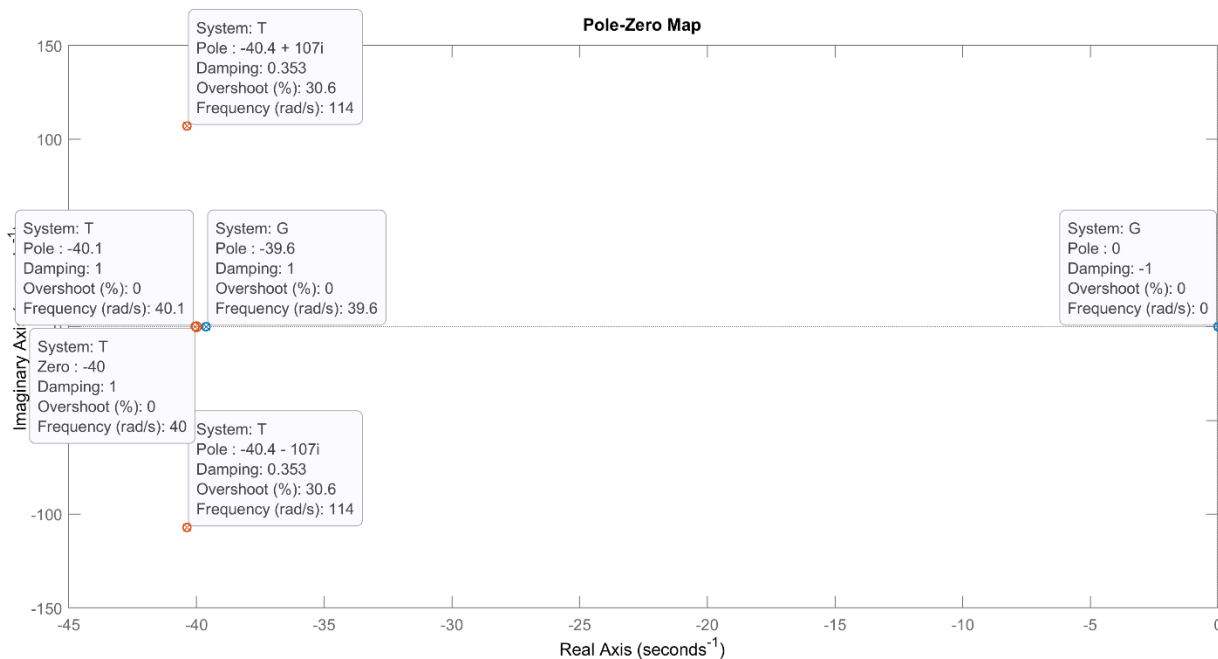
در این بخش از آزمایش قصد داریم کنترل سروو موتور را توسط کنترل کننده Lead انجام دهیم. شرایط مطلوب خواسته شده از ما برای این سیستم، بالازدگی (Overshoot) کمتر از 30% و زمان نشست (Settling time) 0.1 ثانیه می باشد. (زمان نشست با معیار قرار گرفتن در بازه 2% مقدار نهایی در نظر گرفته شده است.)

$$P.O.V = 30 = 100e^{-\pi \frac{\xi}{\sqrt{1-\xi^2}}} \Rightarrow \xi = 0.35$$

$$t_s = 0.1 = \frac{4}{\xi \omega_n} \Rightarrow \omega_n = 114.28$$

$$\omega_d = \omega_n \sqrt{1-\xi^2} = 107.05$$

$$s_d = -\xi \omega_n \pm j \omega_n \sqrt{1-\xi^2} \Rightarrow s_d = -40 \pm 107.05j \rightarrow z = 40$$



8) نمایش همزمان صفرها و قطب‌های سیستم جبران نشده و جبران شده توسط کنترل کننده Lead

با کمک مقادیر به دست آمده، به ترسیم مکان ریشه تابع تبدیل کنترل نشده می پردازیم.

$$\Delta(s) = 0 \rightarrow s_1 = 0, \quad s_2 = -39.62$$

مکان مجانب:

$$\sigma = \frac{\sum P_i - \sum Z_i}{RD} = \frac{\sum P_i - \sum Z_i}{n - m} = \frac{0 - 39.62}{2} = -19.81$$

زاویه مجانب:

$$\varphi = \frac{(2k + 1)\pi}{n - m} = \frac{(2k + 1)\pi}{2} = \frac{\pi}{2}, \frac{3\pi}{2}, \dots$$

تابع تبدیل کنترل کنند Lead در حالت کلی به شکل زیر است:

$$C_{Lead}(s) = k \frac{s + z_e}{s + p_e}$$

برای پیدا کردن محل  $p_e$  باید شرط زاویه را بررسی کنیم:

$$(2k + 1)\pi = \sum \theta_{Zeros} - \sum \theta_{Poles}$$

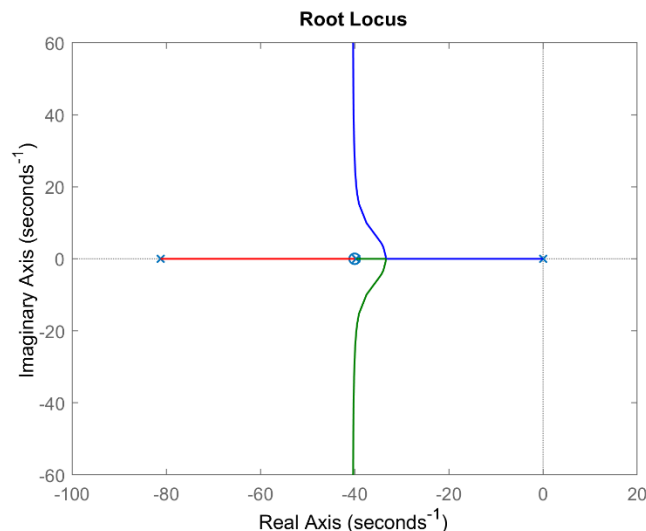
$$\Rightarrow (2k + 1)\pi = \frac{\pi}{2} - \theta_p - \left( \pi - \tan^{-1} \left( \frac{107.05}{40 - 39.62} \right) \right) - \left( \pi - \tan^{-1} \left( \frac{107.05}{40} \right) \right)$$

$$\Rightarrow \theta_p \cong 69.32^\circ \Rightarrow \tan \theta_p = \frac{107.05}{p - 40} \Rightarrow p = 81.18$$

برای پیدا کردن مقدار  $k$ ، شرط اندازه را بررسی می‌کنیم. ما این کار را به کمک دستور `rlocfind` متلب انجام دادیم:

```
K=rlocfind(G*((s+z)/(s+p)),s_d);
```

$$|C_{Lead}(s_d)G(s_d)| = 1 \Rightarrow k = 216.4277$$



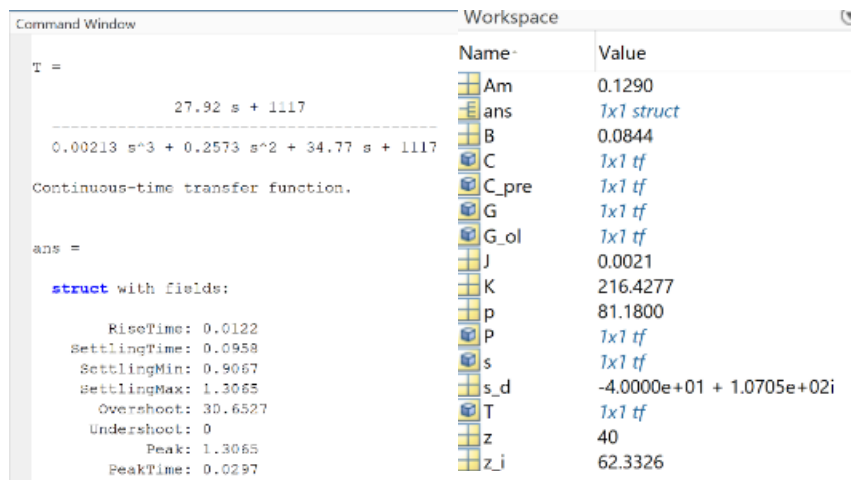
9) مکان ریشه سیستم حلقه باز جبران شده توسط کنترل کننده Lead

در نتیجه محاسبات بالا برای کنترل کننده Lead خود به تابع تبدیل زیر می‌رسیم:

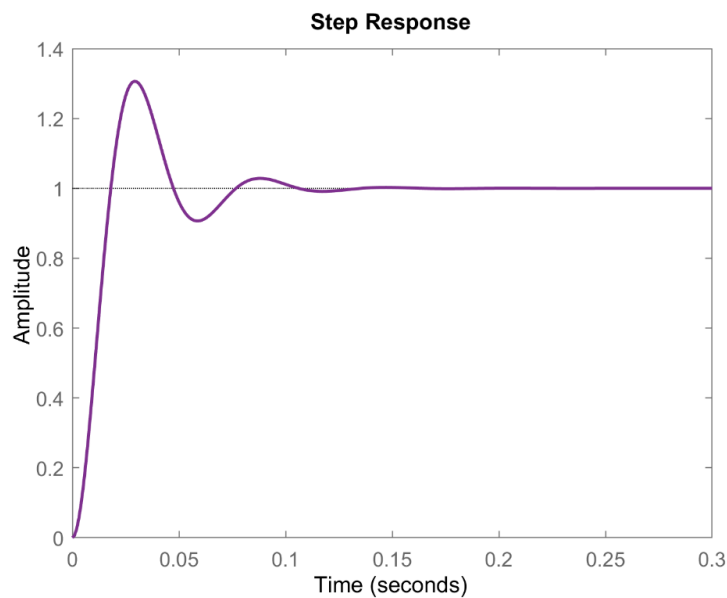
$$C_{Lead}(s) \Rightarrow 216.4277 \frac{s + 40}{s + 81.18}$$

این جبرانساز به صورت سری با تابع تبدیل سروو موتور قرار گرفت و پس از گرفتن فیدبک منفی، پاسخ سیستم جدید به ورودی پله بررسی شد. شبیه‌سازی با نرم افزار Matlab انجام گرفت که در نتیجه آن برای Settling time و Overshoot مقادیر زیر حاصل شد:

$$t_s = 0.0958 \text{ و } P.O.V = \%30.6527$$



(10) داده‌ها و نتایج شبیه‌سازی پس از اعمال کنترل کننده Lead



(11) پاسخ پله سیستم جبران شده توسط کنترل کننده Lead



مشاهده می‌شود که بالازدگی آن به میزان بسیار اندکی از مطلوب ما فاصله دارد. علت این است که کنترل‌کننده Lead یک صفر به سیستم اضافه می‌کند و اضافه شدن صفر سمت چپ، باعث افزایش سرعت پاسخ و افزایش Overshoot می‌شود. در این آزمایش، اختلاف محسوس نبود اما اگر اختلاف قابل توجهی وجود داشت، برای رفع این مشکل می‌توان یک prefilter طراحی کرد و آن را با سیستم حلقه بسته حاصل پس از جبران‌سازی، سری کرد. تابع تبدیل این prefilter به نحوی است که صفر تابع تبدیل حلقه بسته را حذف کرده و بالازدگی را کاهش می‌دهد. ما با تکه کد زیر، این فرآیند را انجام دادیم.

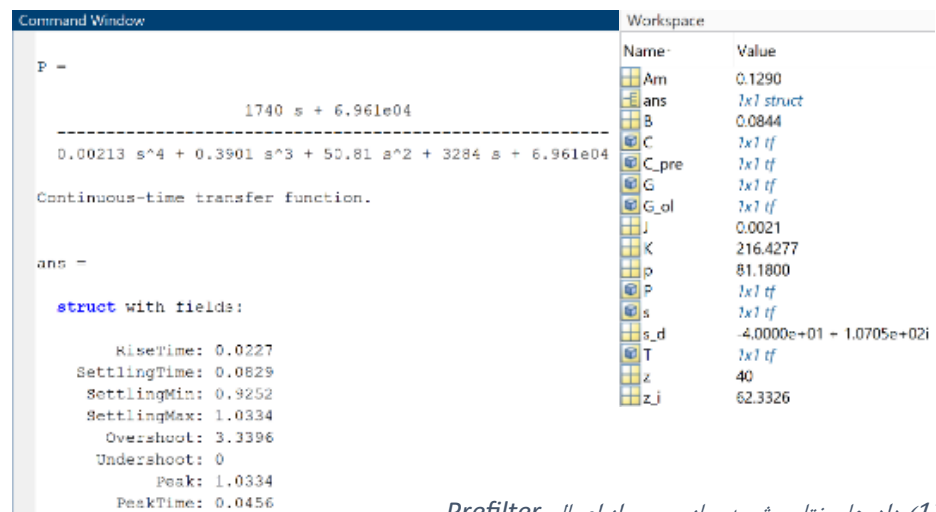
```
z_i = 1117/17.92;
C_pre = z_i / (s + z_i);
P = series (C_pre, T);
```

دیدیم که نتایج جدیدی حاصل شد:

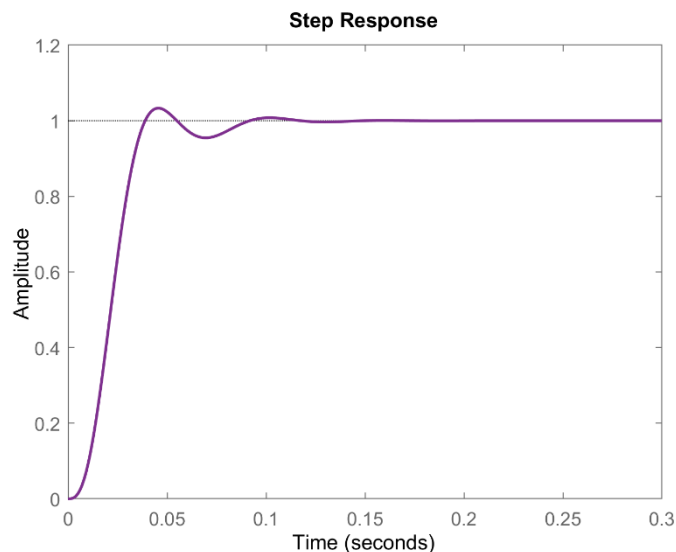
$$C_{pre}(s) = \frac{z_i}{s + z_i}$$

$$t_s = 0.0829$$

$$P.O.V = \%3.3396$$



(12) داده‌ها و نتایج شبیه سازی پس از اعمال Prefilter



(13) پاسخ پله سیستم پس از اعمال Prefilter

## • طراحی کنترل کننده PI

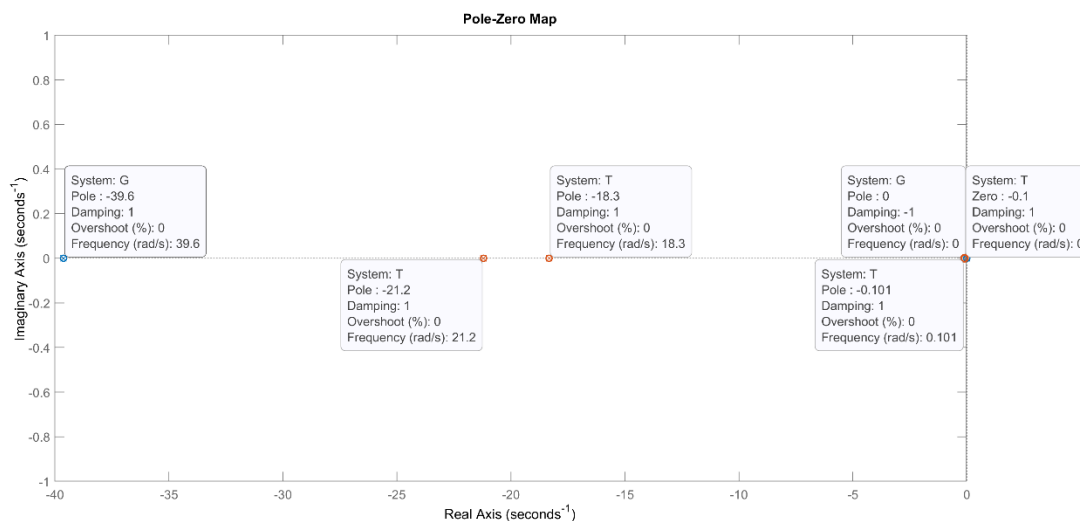
این نوع کنترل کننده حالت خاصی از کنترل کننده پس فاز به شمار می رود، به همین دلیل نوع سیستم را افزایش می دهد و و خطای حالت ماندگار را حذف می کند؛ به همین دلیل، بر خلاف کنترل کننده هایی که پیش تر بررسی شد، در طراحی کنترل کننده PI نیازی به محاسبه بهره های کنترلی نیست.

برای طراحی این کنترل کننده، صفر کنترل کننده را نزدیک به مبدا انتخاب می کنیم. علت این کار آن است که با نزدیک بودن صفر و قطب به همدیگر، سهم زاویه ای این دوقطبی در نمودار مکان ریشه بسیار ناچیز خواهد بود و تغییر فاز قابل توجهی ایجاد نمی کند. با توجه به اینکه قطب در مبدا و صفر نزدیک به مبدا تاثیر چندانی بر شرط اندازه نیز نخواهد داشت، به طور کلی می توان گفت پارامترهای پاسخ گذرا با کنترل کننده PI تقریباً مشابه پارامترهای سیستم اصلی است، اما خطاهای حالت پایدار به دلیل این واقعیت که نوع سیستم کنترل بازخورد یک بار افزایش یافته است، به شدت بهبود می یابد. تابع تبدیل کنترل کننده Lead در حالت کلی به شکل زیر است:

$$C_{PI}(s) = k_p + \frac{K_I}{s} = k \frac{s + z}{s}$$

ضمناً مطابق توضیحات بالا، فرض می کنیم:

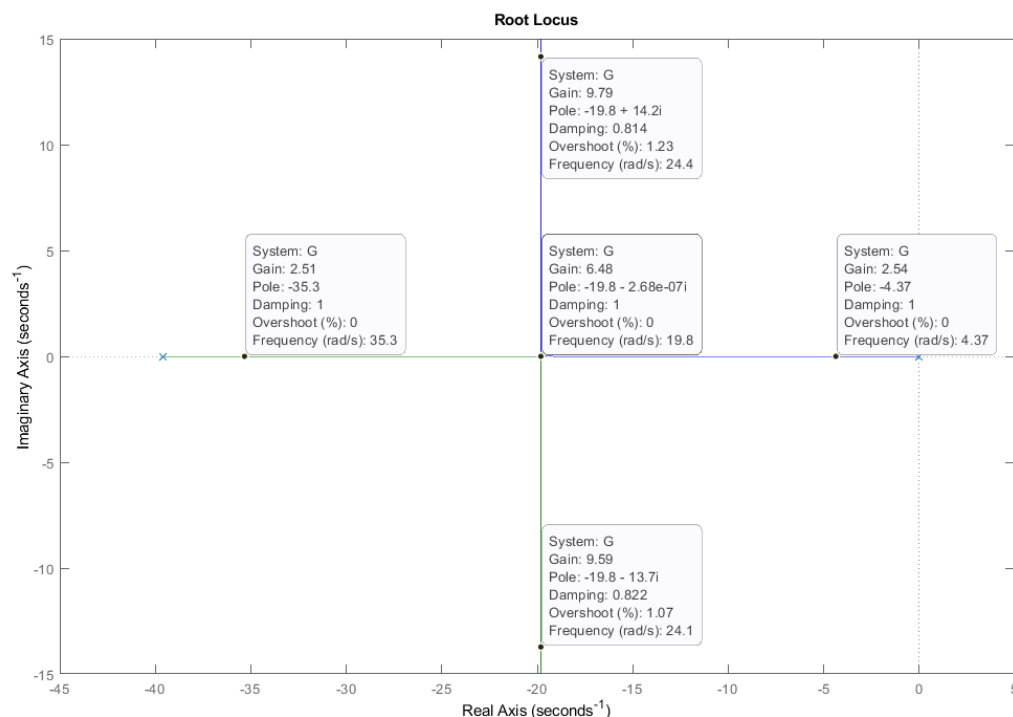
$$z = 0.1$$



در تصویر بالا شاهد نمایش همزمان صفرها و قطب های سیستم جبران نشده و جبران شده توسط کنترل کننده PI هستیم.

برای انتخاب مقدار بهره K، نمودار مکان ریشه سیستم جبران نشده را رسم می کنیم (که مراحل آن قبل تر مرور شد). با توجه به اینکه مکان ریشه در هیچ نقطه ای محور موهومی را قطع نکرده است و کنترل کننده جز در نزدیکی مبدا مختصات تغییری در مکان ریشه سیستم ایجاد نخواهد کرد، با حرکت بر روی محور حقیقی، بیشترین بهره

ممکن را انتخاب می‌کنیم. ملاحظه می‌شود که در نزدیکی شاخه‌های مکان ریشه، بهره به بیشترین میزان خود می‌رسد و با نزدیک شدن به قطب‌ها مقدار آن کاهش پیدا می‌کند.



14) بررسی بهره، میرایی و overshoot در نقاط مختلف مکان ریشه جبران نشده

با حرکت بر روی شاخه‌های نمودار، اگرچه بهره افزایش می‌یابد اما میرایی کاهش پیدا کرده و مقدار بالازدگی افزایش پیدا خواهد کرد. بنابراین:

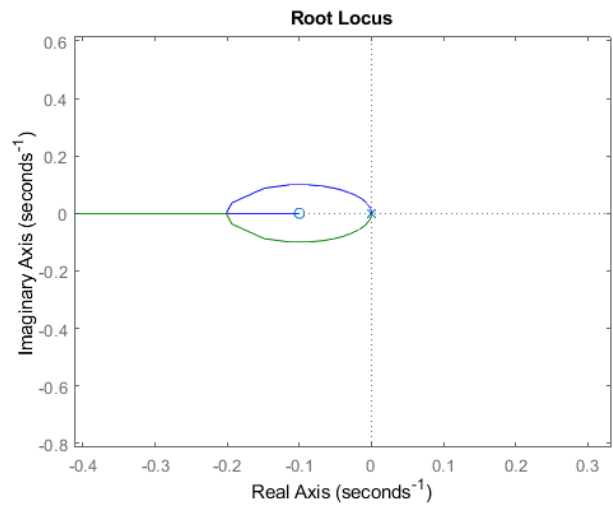
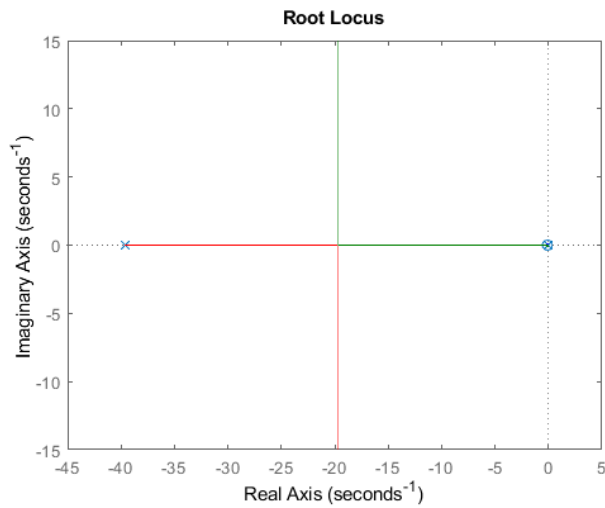
$$k = 6.48$$

نهایتاً تابع تبدیل کنترل‌کننده PI طراحی شده، به شکل زیر درمی‌آید:

$$C_{PI}(s) = 6.48 \frac{s + 0.1}{s}$$

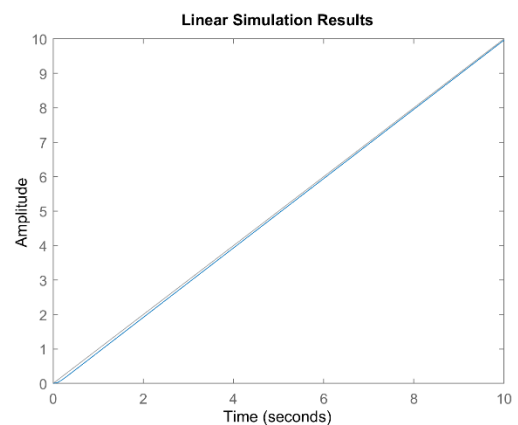
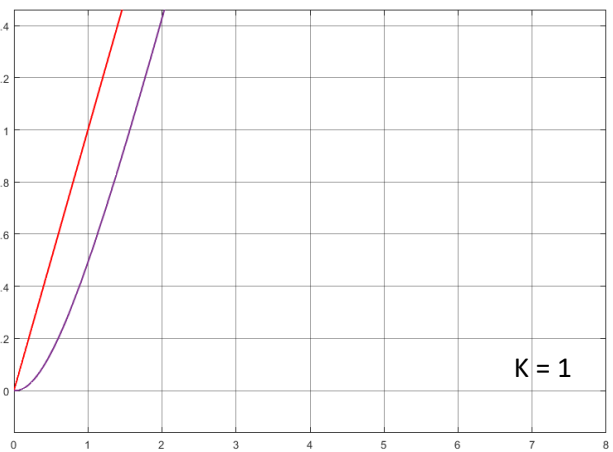
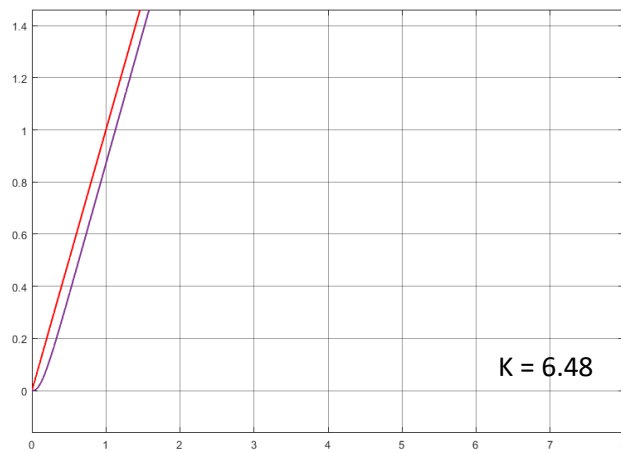
در ادامه همانند مراحل قبل، توابع تبدیل حلقه باز جبران‌شده و حلقه بسته را محاسبه می‌کنیم.

ضمناً با توجه به صفر بودن خطای حالت دائمی در پاسخ به ورودی پله، برای بررسی خطای حالت دائمی در پاسخ به ورودی شیب از شبیه‌سازی Simulink استفاده می‌کنیم که در بخش بعد شرح داده خواهد شد.



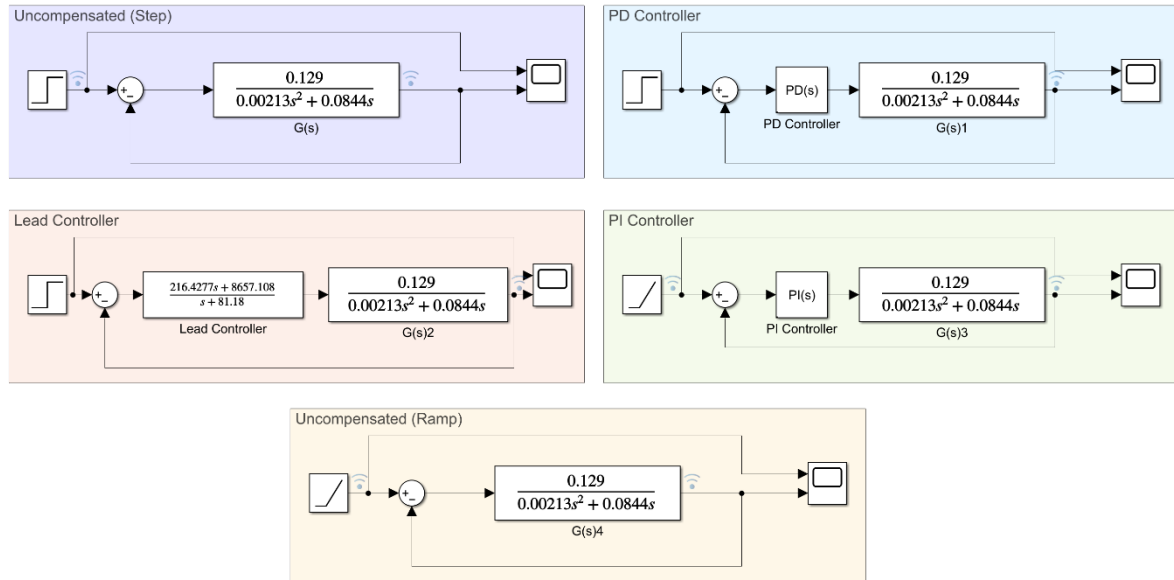
15) نمودار مکان ریشه حلقه باز، و رفتار آن در نزدیکی مبدا

16) تفاوت پاسخ ورودی شیب به ازای  $k=6.48$  و  $k=1$  و  $k=20$   
کاهش بهره موجب افزایش خطای خروجی شده است!

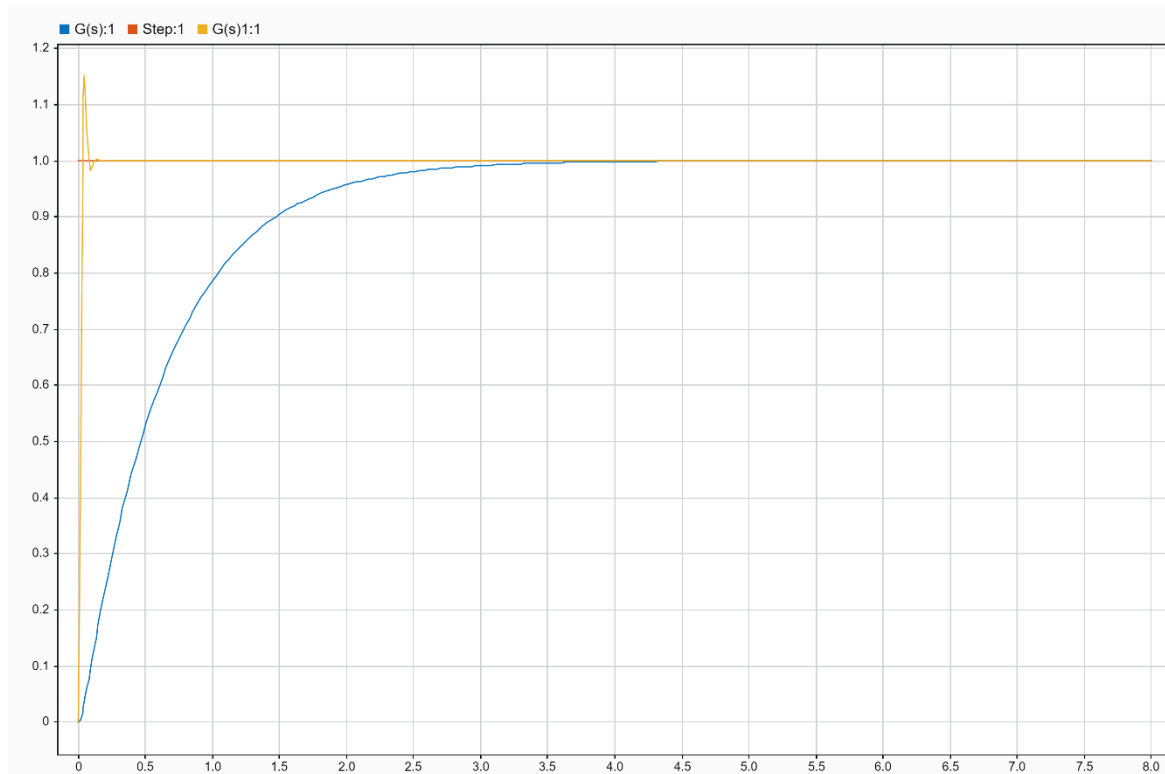


## • شبیه سازی در محیط Simulink

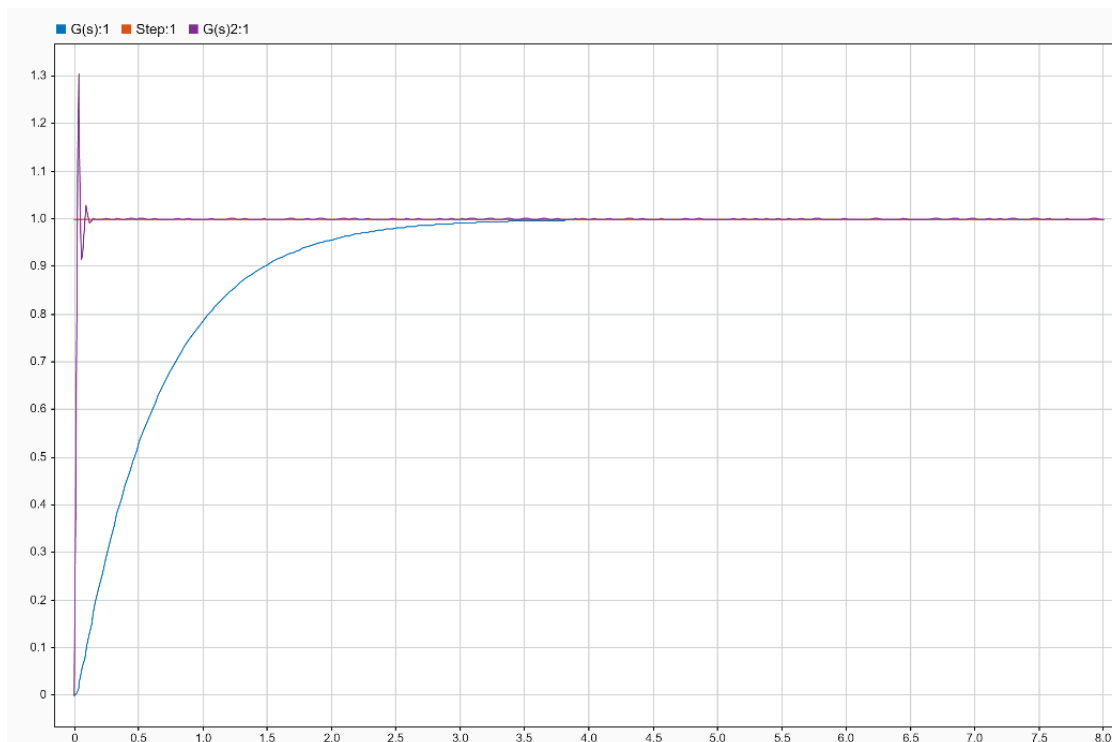
پس از انجام مراحل شرح داده شده، یک بار هم در محیط Simulink شبیه سازی انجام داده و به بررسی اثر حضور کنترل کننده ها و مقایسه آن ها با یکدیگر پرداختیم.



17) شبیه سازی سیستم در غیاب و حضور کنترل کننده های مختلف



18) پاسخ پله سیستم در غیاب و در حضور کنترل کننده PD



19) پاسخ پله سیستم در غیاب و در حضور کنترل کننده Lead

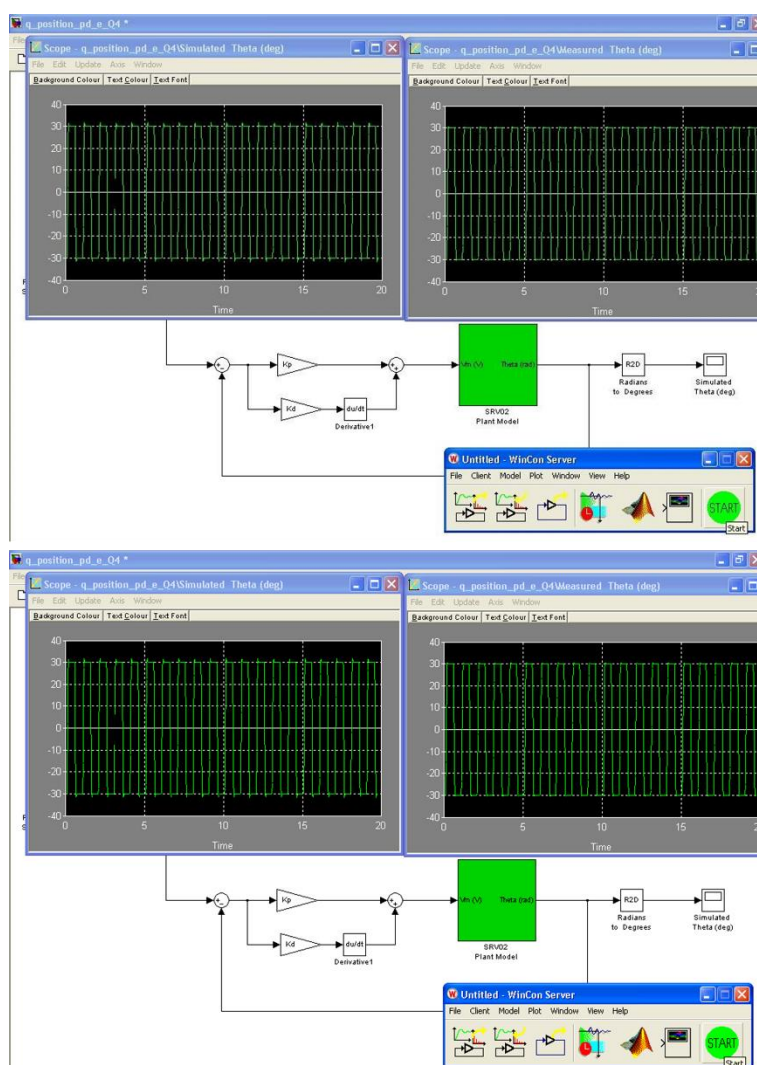


20) پاسخ شیب سیستم در غیاب و در حضور کنترل کننده PI

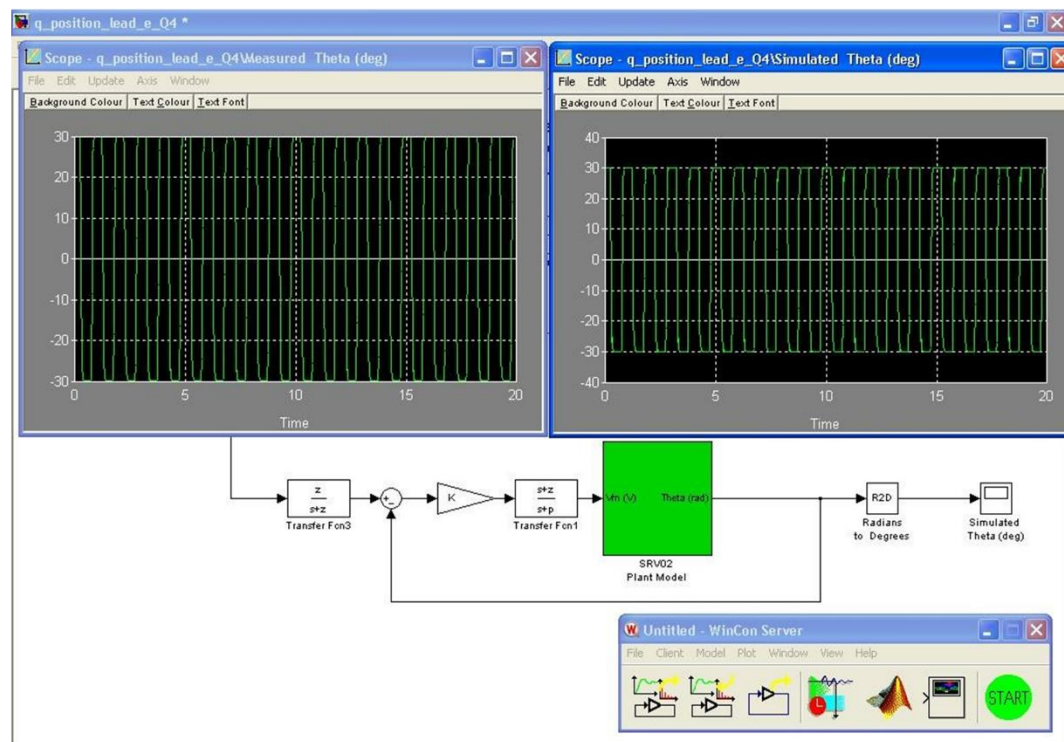
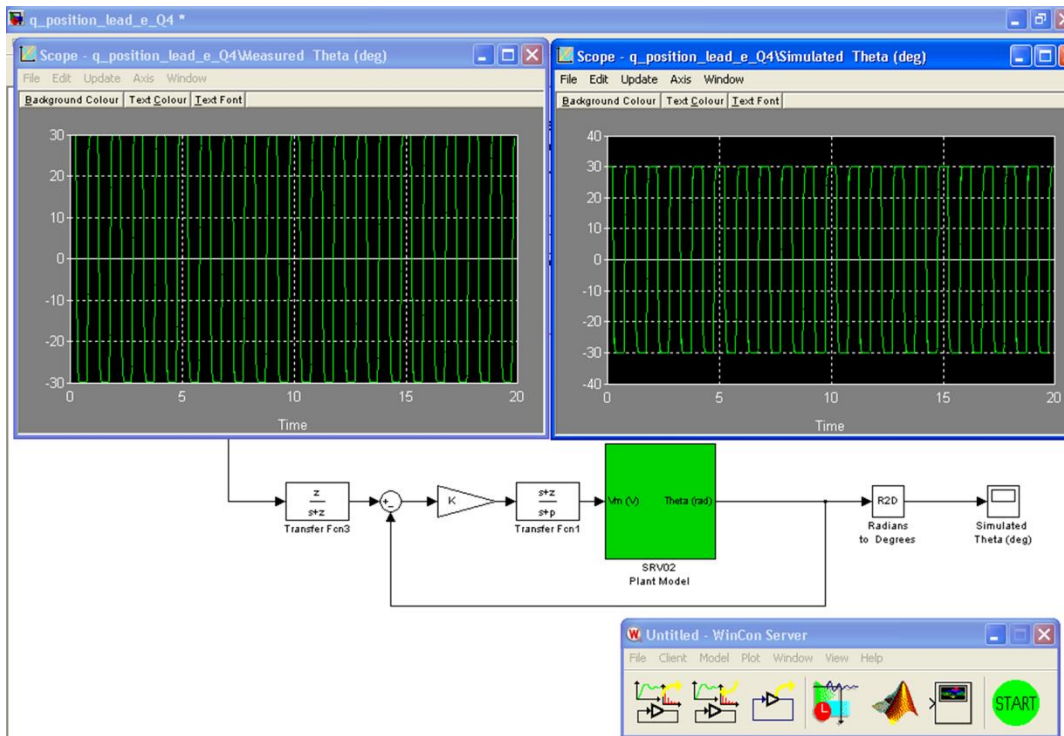
## • نتیجه گیری

1) بعد از طراحی کنترل کننده، آیا مقدار *overshoot* و *rise time* سیگنال‌های *Measured theta* و *Simulated theta* با هم و با مقدار مطلوب تفاوت داشت؟ اگر جواب مثبت است دلیل آن را توضیح دهید. به نظر شما آیا راه حلی وجود دارد که این اختلاف به حداقل برسد؟ توضیح دهید.

بله. علت این اختلاف در درجه اول تقریب‌هایی است که در محاسبات استفاده می‌شوند و در درجه دوم، خطایی است که افزودن هریک از کنترل‌کننده‌ها به سیستم اضافه می‌کند. همچنین در پیاده‌سازی عملی عواملی همچون نویز و خطای فیزیکی دستگاه در اختلاف نسبت به مقدار تئوری دخیل است. برای کمتر کردن اختلاف می‌توان از کنترلرهای دیگر (مثلاً PID) استفاده کرد. همچنین می‌توان تقریب‌های کمتری در محاسبات استفاده کرد و پارامترهای مربوط به موتور مانند  $J_{eq}$  که ممکن است در طول زمان تغییر کرده باشد را دوباره اندازه‌گیری کنیم.



21) سیگنال‌های *Measured theta* و *Simulated theta* مربوط به دو کنترل کننده PD طراحی شده



22) سیگنال‌های *Simulated theta* و *Measured theta* مربوط به دو کنترل کننده *Lead* طراحی شده



## 2) تفاوت میان کنترل کننده PD و Lead را در پیاده سازی عملی شرح دهید.

کنترل کننده Lead برای افزودن فاز به یک سیستم در محدوده فرکانسی معینی طراحی می شود؛ در صورتی که کنترل کننده PD، فاز را نیز اضافه می کند، و این فاز در هر فرکانس بالاتر از فرکانس نقطه شکست صفر کنترل کننده اضافه می شود.

هنگامی که سیگنال خطا به سرعت تغییر می کند، به بهبود عملکرد کنترل کمک می کند؛ اما همزمان می تواند اختلالات ناخواسته را در خروجی اندازه گیری شده تقویت کند. بنابراین، استفاده از عملکرد مشتق همیشه باید محدود باشد. دلیل این موضوع با بررسی تابع ریاضی مشتق مشهودتر خواهد شد:

$$x'(t) = \lim_{T \rightarrow 0} \frac{x(k+1) - x(k)}{T}$$

با توجه به اینکه T به صفر میل می کند، اگر سیگنال ورودی x به تابع مشتق را سیگنال نویز در نظر بگیریم، مشتق آن علیرغم کوچک بودن صورت، با توجه به مخرج کسر قابل توجه خواهد شد.

یک کنترل کننده Lead، بهره را با فرکانس بالا حذف می کند؛ در صورتی که کنترل کننده PD به دلیل خاصیت مشتق گیر بودن می تواند نویزهایی که در عمل قابل چشم پوشی هستند را به شدت تقویت کند. از همین رو در پیاده سازی های عملی، عملکرد کنترل کننده Lead را می توان با سری کردن یک کنترل کننده PD با یک فیلتر پایین گذر (که فرکانس شکست آن بالاتر از فرکانس صفر کنترل کننده PD انتخاب شده است) تحقق بخشید.