بسم الله الرحمن الرحیم

گزارش کار آزمایش شماره2 – آزمایشگاه کنترل خطی

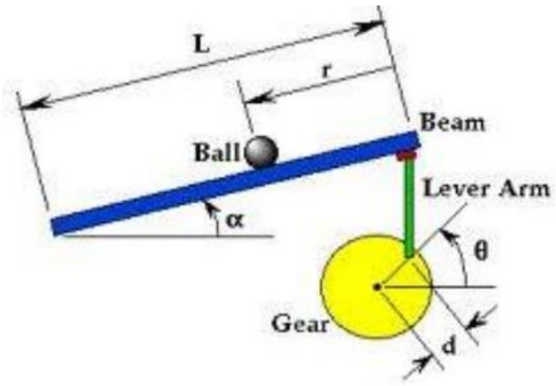
اعضای گروه: سیده لیلا حسینی 9923024 مریم مقتدری 9923073 زهرا لطیفی 9923069

نام آزمایش: توپ و میله (Ball and Beam)

هدف: کنترل موقعیت توپ با استفاده از کنترلر PV و Lead

شرح آزمایش:

در این آزمایش، تمام طراحیها برای سیستم گوی و میله انجام میگیرد که تصویر 1، مدل ریاضی آن را نشان می دهد:



تابع تبدیل این سیستم، به نحوی که در دستورکار آمده، محاسبه شده و در اختیار ما قرار گرفت:

$$\frac{\theta_l(s)}{V_m(s)} = \frac{\eta_g \eta_m K_t K_g}{J_{eq} R_m s^2 + \left(B_{eq} R_m + \eta_g \eta_m K_t K_m {K_g}^2\right)s}$$

$$\frac{X(s)}{V_m(s)} = \frac{\theta_l(s)}{V_m(s)} \frac{\alpha(s)}{\theta_l(s)} \frac{X(s)}{\alpha(s)}$$

کنترل کننده *lead* :

در این بخش از آزمایش قصد داریم کنترل توپ را توسط کنترل کننده *lead* انجام دهیم.شرایط خواسته شده از ما برای این سیستم، بالازدگی (*over shoot*) *5%* و زمان نشست *(settling time)* 0.1 ثانیه می باشد.

تابع تبدیل داده شده به ما به شکل روبه رو است: $\frac{\theta}{V_m} = \frac{64.2}{s^2+36.425s}$

$$P.O. = 100e^{-\pi\frac{\zeta}{\sqrt{1-\zeta^2}}} = 5 \rightarrow \zeta = 0.7$$

$$t_s = \frac{4}{\omega_n\zeta} = 0.1 \rightarrow \omega_n = 57.92$$

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

$$s_d = -\omega_n\zeta \pm j\omega_n\sqrt{1-\zeta^2} \rightarrow s_d = -40 \pm j41.987$$

تابع تبدیل کنترل‌کنند 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:

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

$$\rightarrow (2k+1)\pi = \frac{\pi}{2} - \theta_p - \left(\pi - \tan^{-1}\left(\frac{41.987}{40-36.42}\right)\right) - \left(\pi - \tan^{-1}\left(\frac{41.987}{40}\right)\right)$$

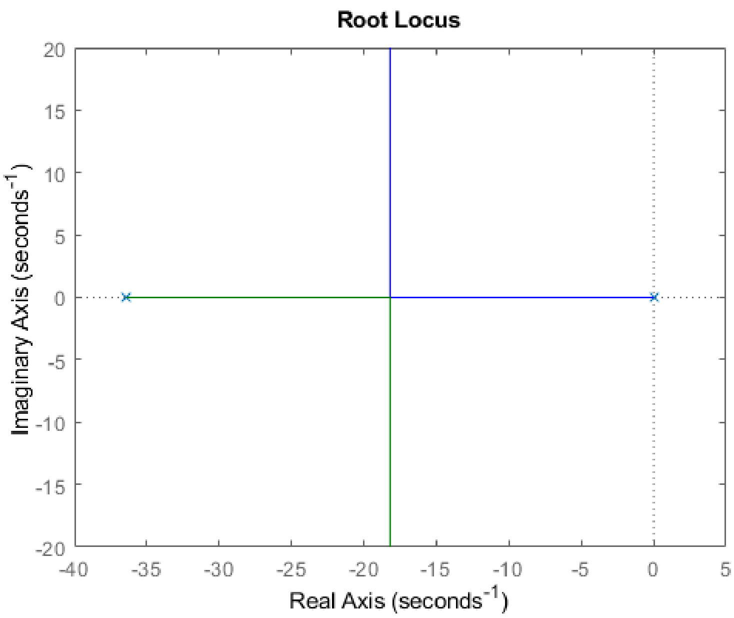
$$\rightarrow \theta_p = 41.516° \rightarrow \tan(\theta_p) = \frac{41.987}{p-40} \rightarrow p = 87.442$$

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);
```

$$|C_{Lead}(s_d)G(s_d)| = 1 \Rightarrow k = 57.433497593838865$$

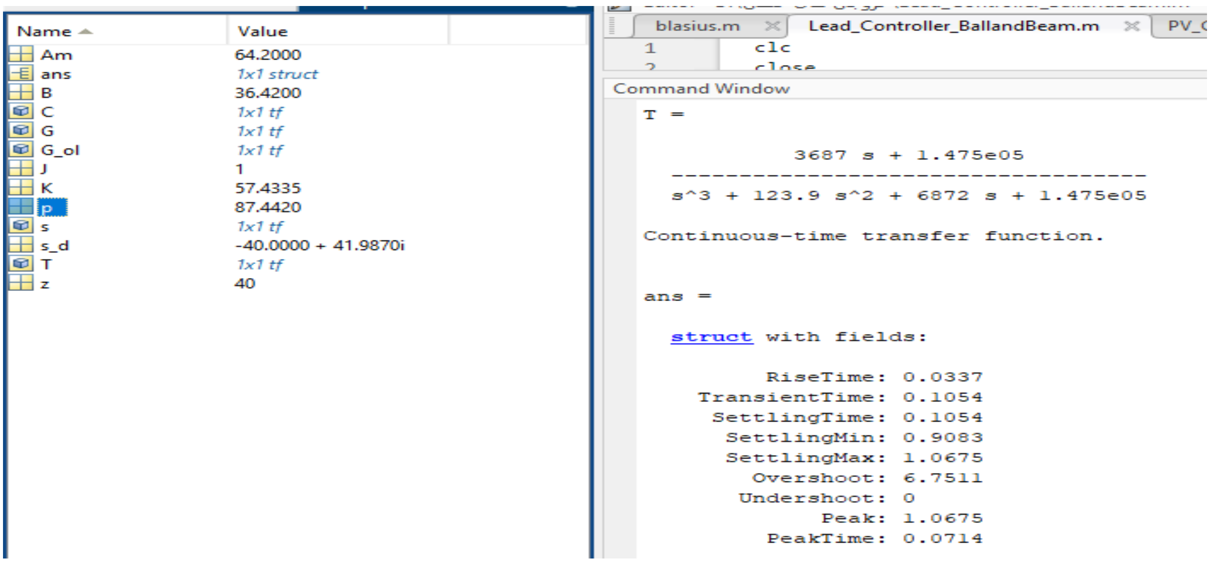
نمودار مکان ریشه با کمک دستور *rlocus* متلب رسم می کنیم:

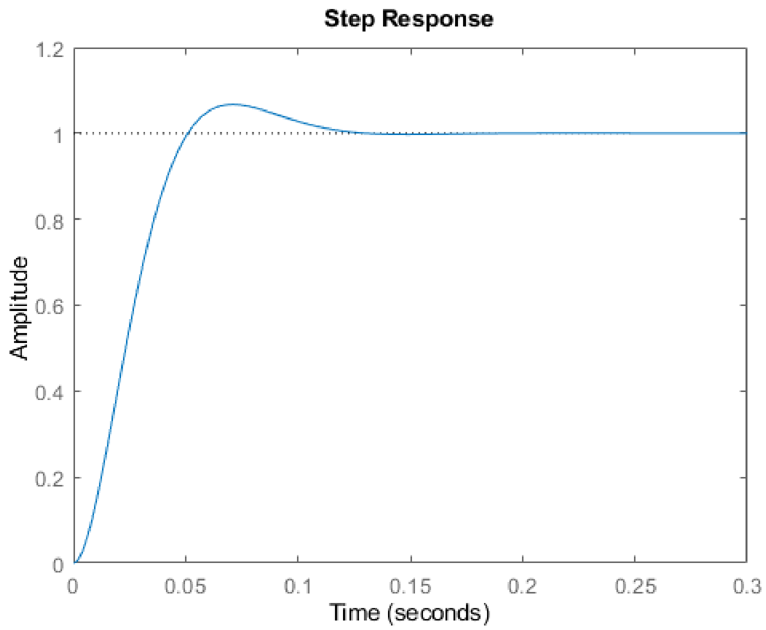


در نتیجه محاسبات بالا برای کنترل‌کننده Lead خود به تابع تبدیل زیر می‌رسیم:

$$C_{Lead}(s) \Rightarrow 57.4335\frac{s+40}{s+87.442}$$

شبیه‌سازی با نرم افزار Matlab انجام گرفت که در نتیجه آن برای Overshoot و Settling time مقادیر زیر حاصل شد:

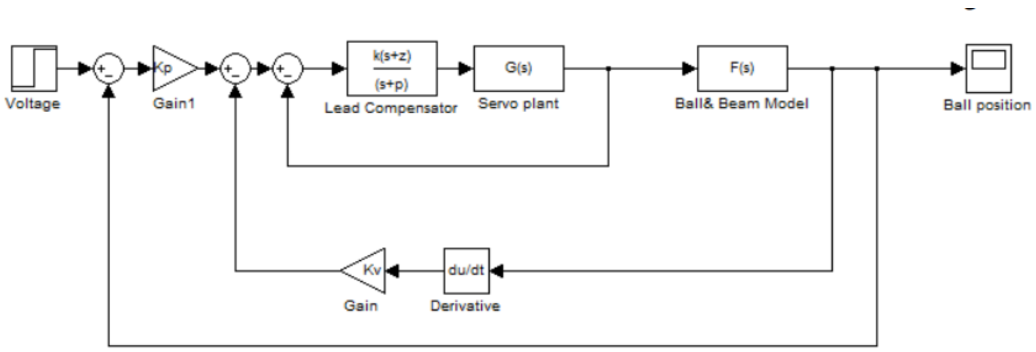




پاسخ پله کنترل کننده lead

طراحی کنترل کننده PV:

قبل از طراحی به بررسی این نوع کنترل کننده بر روی سیستم می پردازیم:



شکل بالا حلقه کنترلی سیستم را نشان می دهد که از 2 حلقه تشکیل شده است. کنترل کننده خارجی وظیفه ی ایجاد سیگنال زاویه مطلوب برای کنترل کننده داخلی را بر عهده دارد.

ازبلوک دیاگرام بالا تابع تبدیل زیر بدست می آید:

$$\frac{Y(s)}{R(s)} = \frac{k_p G(s) G_{lead} F(s)}{1 + G_{lead} G(s) + G_{lead} G(s) F(s) k_v (\frac{dv}{dt})}$$

حلقه داخلی که حلقه موتور است بسیار سریع است و می توان از آن چشم پوشی کرد و آن را با گین واحد تقریب زد:

$$\frac{\mathrm{Y(s)}}{\mathrm{R(s)}} = \frac{\mathrm{k_p F(s)}}{2+\mathrm{G_{lead} G(s) F(s) k_v (\frac{dv}{dt})}}$$

حال به طراحی کنترلر PV می پردازیم:

شرایط مطلوب خواسته شده از ما برای این سیستم ، بالازدگی(*overshoot*) کمتر از 4.6% و زمان پیک(peak time) 1.5 ثانیه می باشد.

$$\mathrm{P.O.} = 100\mathrm{e}^{-\pi\frac{\zeta}{\sqrt{1-\zeta^2}}} = 4.6 \rightarrow \zeta = 0.707$$

$$\mathrm{t_p} = \frac{\pi}{\omega_\mathrm{d}} = 1.5 \rightarrow \omega_\mathrm{d} = 2.094$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2} \rightarrow \omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}} = \frac{2.094}{\sqrt{1-(0.707)^2}} \cong 2.961$$

با توجه به ω_n و ζ بدسه آمده می توان تابع تبدیل سیستم را به صورت زیر نوشت:

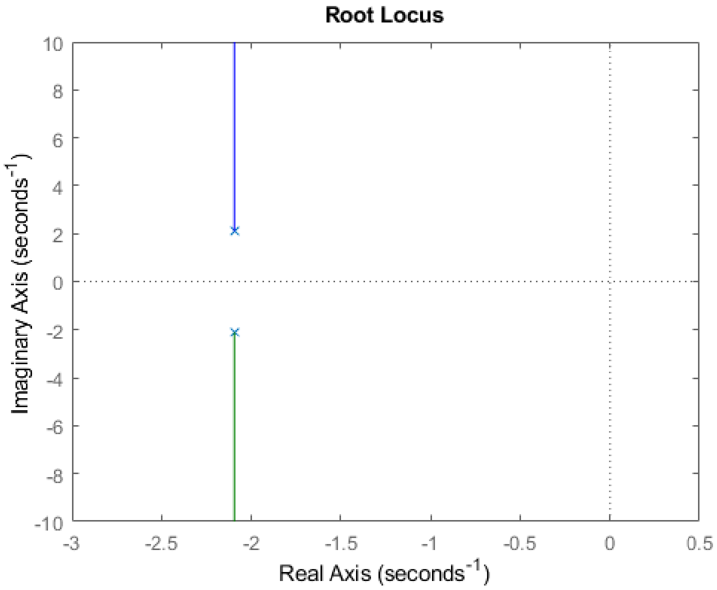
$$T(s) = \frac{Y(s)}{R(s)} = \frac{{\omega_n}^2}{s^2 + 2\omega_n \zeta s + {\omega_n}^2} \rightarrow \frac{Y(s)}{R(s)} = \frac{8.767}{s^2 + 4.187s + 8.767}$$

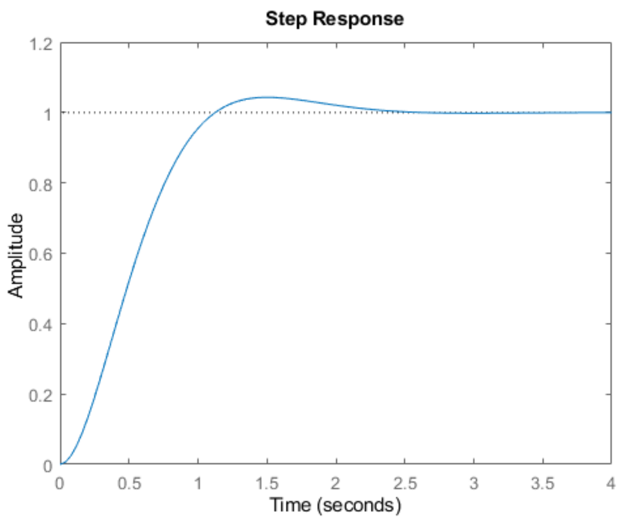
تابع به دست آمده را با تابعی که به ما داده شد برابر قرار می دهیم:

$$\frac{Y(s)}{R(s)} = \frac{8.767}{s^2 + 4.187s + 8.767} = \frac{7k_p}{s^2 + 7k_v s + 7k_p}$$

$$\Longrightarrow \begin{cases} 7k_p = 8.767 \rightarrow k_p = 1.2524\,, \\ 7k_v = 4.187 \rightarrow k_v = 0.5981 \end{cases}$$

نمودار مکان ریشه *pv* با کمک دستور *rlocus* متلب رسم می کنیم:

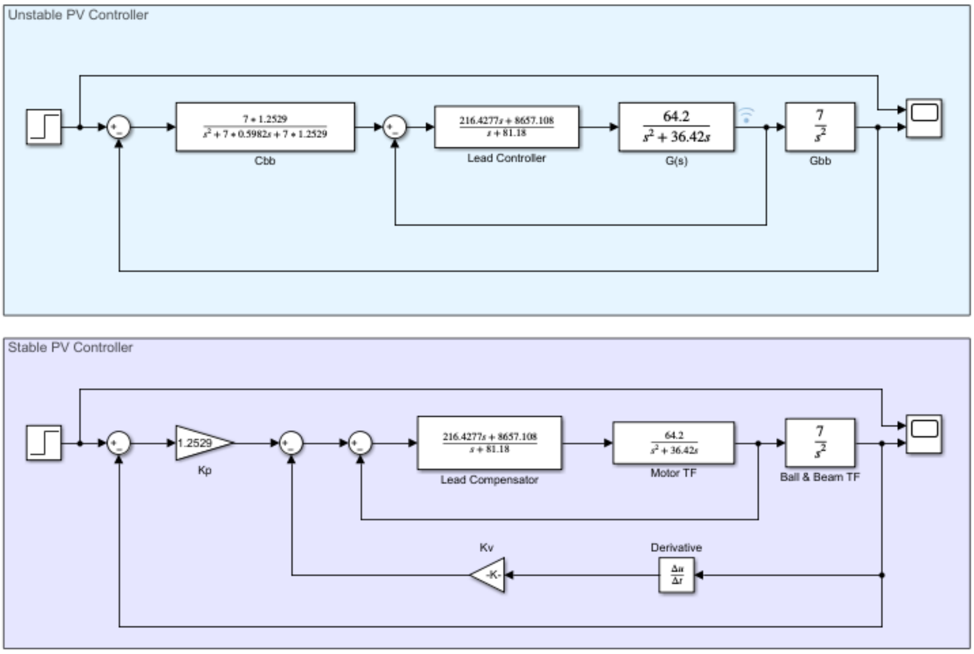


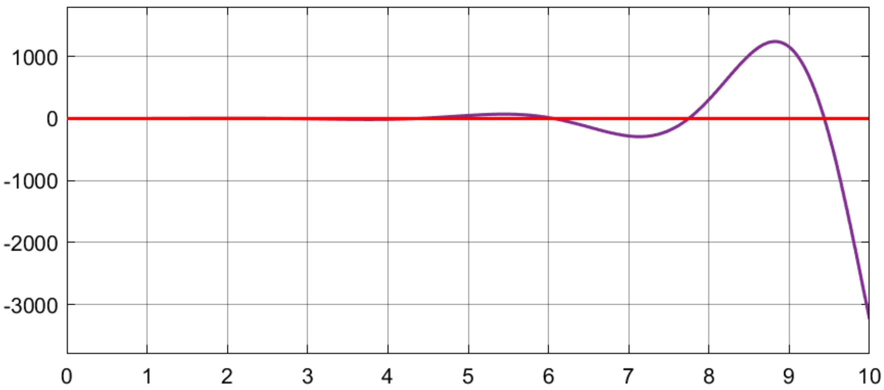


پاسخ پله کنترل کننده pv

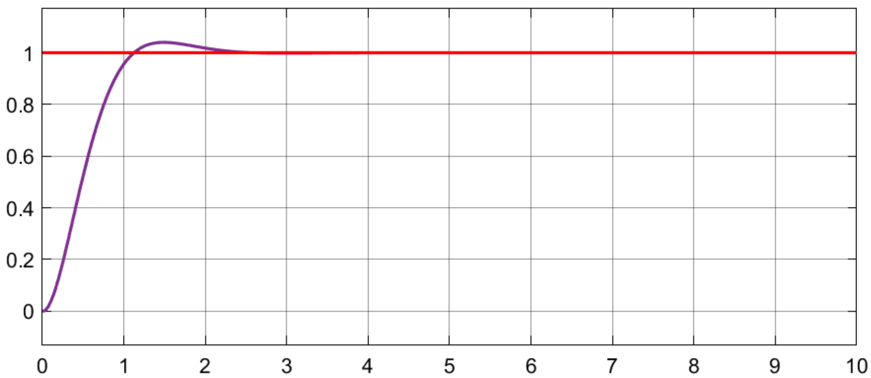
شبیه سازی در Simulink

پس از انجام مراحل شرح داده شده، یک بار هم در محیط Simulink شبیه سازی انجام دادیم:





خروجی سیستم ناپایدار



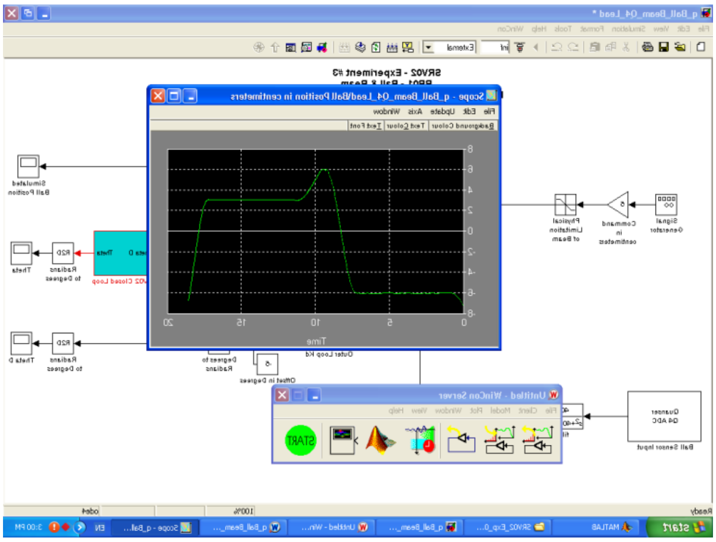
خروجی سیستم پایدار

نتیجه گیری

1-بعد از طراحی کنترل کننده و مشاهده پاسخ، آیا مقدار overshoot و time rise سیگنالهای position ball simulated وposition ball Measured 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 تقریب هایی است که در محاسبات استفاده می شوند(به عنوان مثال $\sin\alpha = \alpha$)

همچنین در پیاده سازی عواملی مثل نویز و خطای فیزیکی دستگاه در اختلاف نسبت به مقدار تئوری دخیل است میتوان به جای تقریب با کمک ژاکوبین از خطی سازی ترم غیرخطی $\sin\alpha$ استفاده کرد یا به جای کنترل خطی از کنترل غیرخطی استفاده کرد یا میتوان از نوع دیگری کنترل کننده استفاده کرد.



2-از چه کنترل کننده های دیگری می توانستید استفاده کنید؟ دلیل استفاده از کنترل کننده PV را توضیح دهید

بله می توانستیم از کنترلر *lead-lag* استفاده کنیم. مزیت *pv* این است که مشتق در مسیر *forward* نیست و در مسیر فیدبک حلقه دوم قرار دارد.اگر ورودی تغییر ناگهانی داشته باشد(تابع پله) مشتق گیر در این حالت باعث آسیب دیدن سیستم نمی شود .

3-آیا میتوان به جای استفاده از کنترل کننده Lead برای کنترل موقعیت موتور نیز از کنترل کننده ی PV استفاده نمود؟ اگر بله به لحاظ تئوری طراحی و شبیه سازی نمایید و همچنین مالحظات عملی را نیز بررسی نمایید. اما اگر پاسخ منفی است؛ چرا؟

کنترلر *PV* و *Lead* ویژگی های متفاوتی دارند و از هریک به منظوری خاص در مدار استفاده می‌شود. از کنترلر *Lead* برای بهبود پایداری و عملکرد موتور استفاده می‌شود، در صورتی که از کنترلر *PV* برای *tracking* دقیق‌تر موقعیت و سرعت مطلوب موتور می‌توان استفاده کرد. بنابراین علیرغم اینکه از لحاظ تئوری استفاده از کنترلر *PV* برای موتور امکان‌پذیر است، اما به دلیل اینکه ممکن است پایداری سیستم را تحت تاثیر قرار دهد بهتر است از این کنترلر برای کنترل موقعیت موتور استفاده نکنیم، زیرا با توجه به سرعت بالای موتور، در صورت ناپایدار بودن سیستم عمر مفید آن به دلیل اعمال ضربه‌های ناگهانی به شدت کاهش خواهد یافت.

4- آیا با عدم صرف نظر از حلقه ی داخلی همچنان میتوان از PV برای حلقه ی خارجی استفاده نمود؟ اگر بله به لحاظ تئوری طراحی و شبیه سازی نمایید. اما اگر پاسخ منفی است یک کنترل کننده برای این منظور پیشنهاد داده و عملکرد آن را بررسی کنید.

امکان استفاده وجود دارد، اما با توجه به اینکه در این صورت تابع تبدیل نهایی پیچیده خواهد شد، در صورتی که نخواهیم از تقریب استفاده کنیم محاسبه مقدار دقیق آن دشوار خواهد بود و در صورتی که از تقریب استفاده کنیم، جواب با حالت قبل یکسان خواهد شد. به این ترتیب باید از طراحی دستی استفاده کنیم:

$$T_s = 0.1 = \frac{4}{\zeta\omega_n} \rightarrow \zeta\omega_n = 40$$

$$P.O < 4.7\% \rightarrow \zeta > \frac{1}{\sqrt{2}} \rightarrow 40\sqrt{2} > \omega_n$$

$$s_d = -\zeta\omega_n \pm \omega_n\sqrt{1-\zeta} = -40 \pm j40$$

شرط زاویه:

$$\frac{\pi}{2} - \left(\arctan\frac{40}{p-40} + \frac{\pi}{2} + \arctan\frac{40-36.42}{40} + \frac{\pi}{2} + \arctan\frac{40}{40}\right) = \pi$$

$$\rightarrow p = 87.863$$

با توجه به توضیحات بالا گین را برابر با 57 فرض می کنیم و خواهیم داشت:

$$C_{lead} = 57\frac{s+40}{s+87.863}$$

$$T_{cl_{srvo}} = \frac{C_{lead} G_{srvo}}{1 + C_{lead} G_{srvo}}$$

$$= \frac{57(s+40)(64.2)}{\big((s+87.863)(s^2+36.42s)\big) + \big(57(s+40)(64.2)\big)}$$

$$= \frac{3659s + 146376}{s^3 + 124.28s^2 + 3185s}$$

$$G_{plant} = T_{cl_{srvo}} G_{bb} = \frac{3659s + 146376}{s^3 + 124.28s^2 + 3185s} \frac{7}{s^2}$$

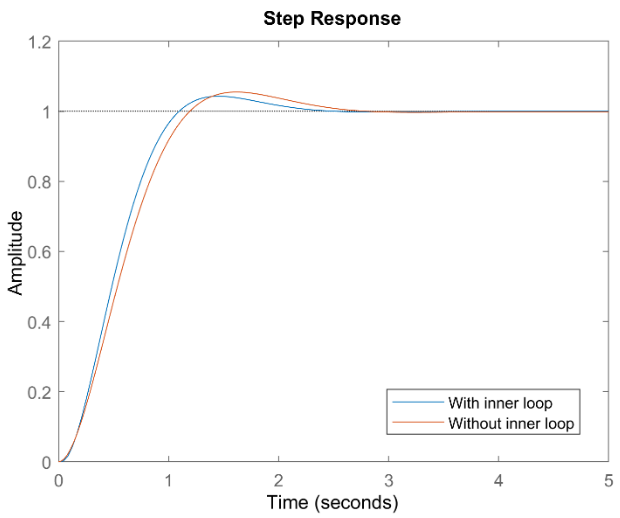
$$T(s) = \frac{Y(s)}{R(s)} = \frac{G(s)\,.K_P}{1 + \ G(s)\,.K_P + G(s).\ K_V s}$$

$$= \frac{7(3659s + 146376)}{s^2(s^3 + 124.28s^2 + 3185s) + 7(3659s + 146376)}$$

$$= \frac{7(s+40)}{\frac{s^3}{3659}(s+35.877)(s+88.023) + 7(s+40)}$$

$$\approx \frac{7}{\frac{s^3}{3659}(s+88.023) + 7}$$

5- بر اساس پاسخی که به سوال قبلی می‌دهید با قراردادن منحنی‌های خروجی بر روی یکدیگر؛ اثر ایده‌آل در نظر گرفتن حلقه داخلی را بررسی کنید.




```
     RiseTime: 0.6910
 SettlingTime: 1.9411
  SettlingMin: 0.9025
  SettlingMax: 1.0432
    Overshoot: 4.3183
   Undershoot: 0
         Peak: 1.0432
     PeakTime: 1.4458
```