بسم تعالی

آزمایشگاه کنترل

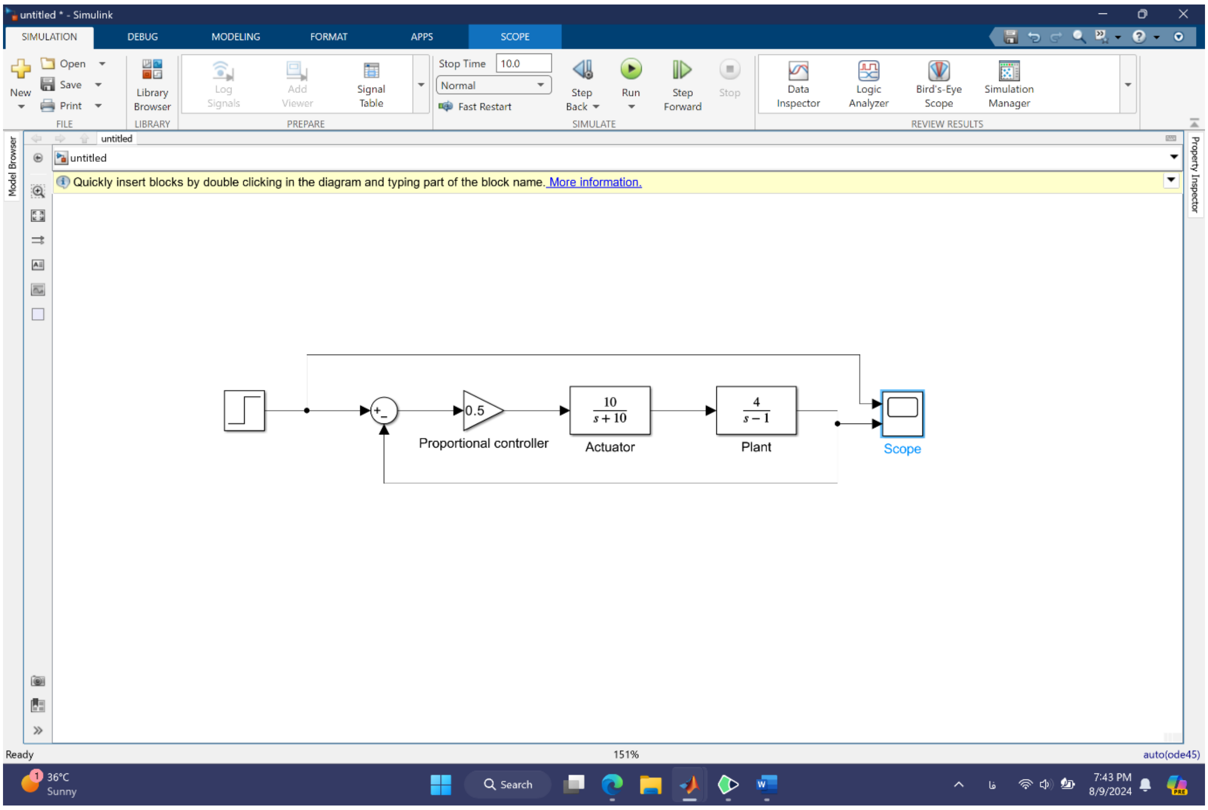
تمرین شماره ۲

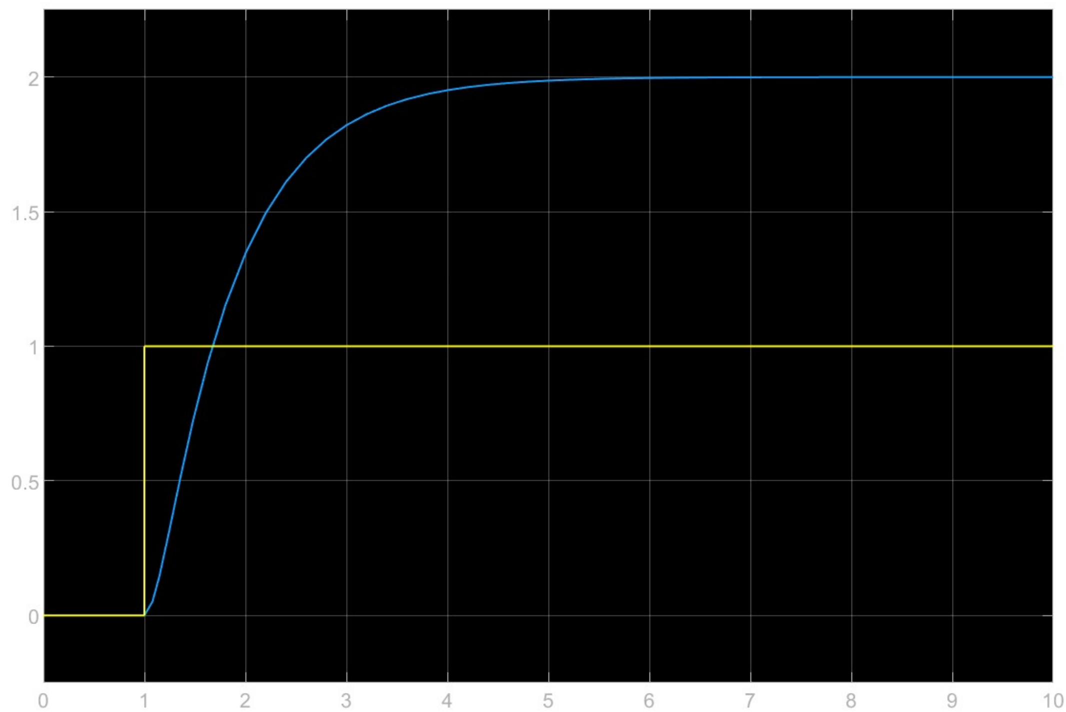
امیرحسین زاهدی ۹۹۱۰۱۷۰۵

تابستان ۱۴۰۳

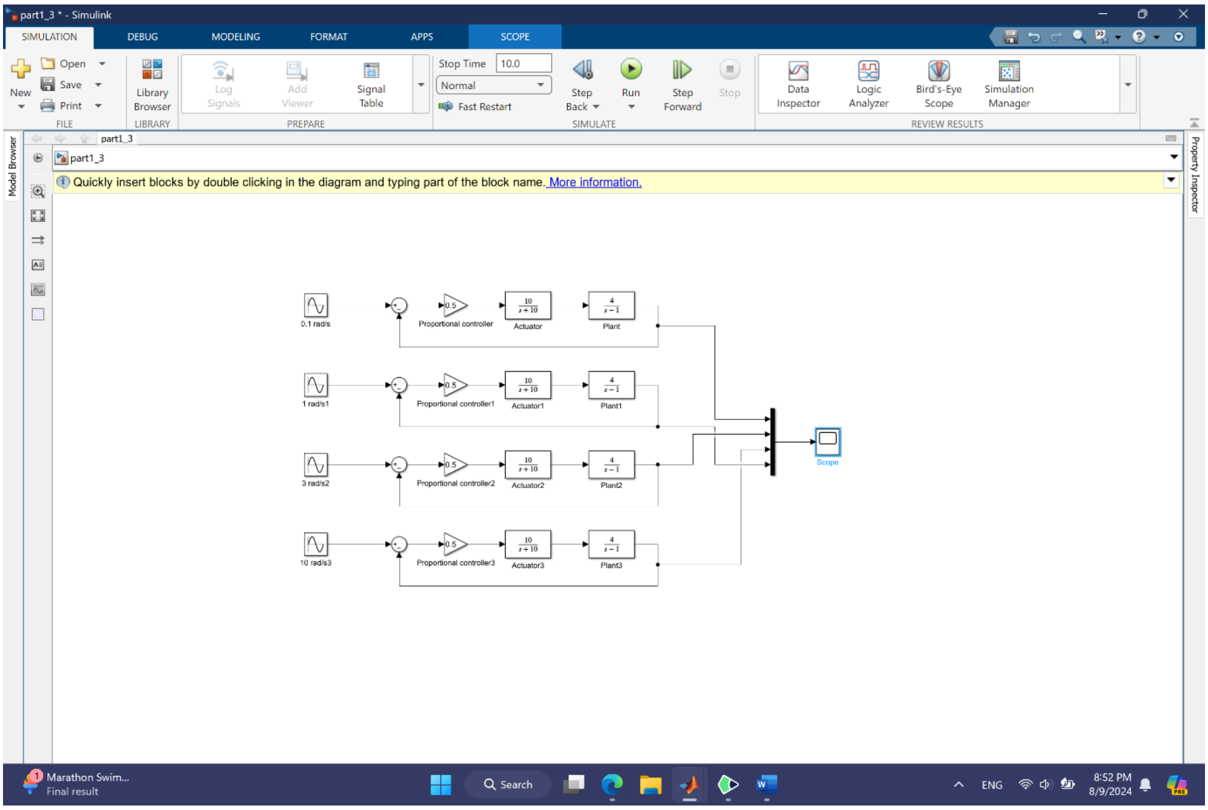
بخش ۱: شبیه سازی مدل با سیمولینک

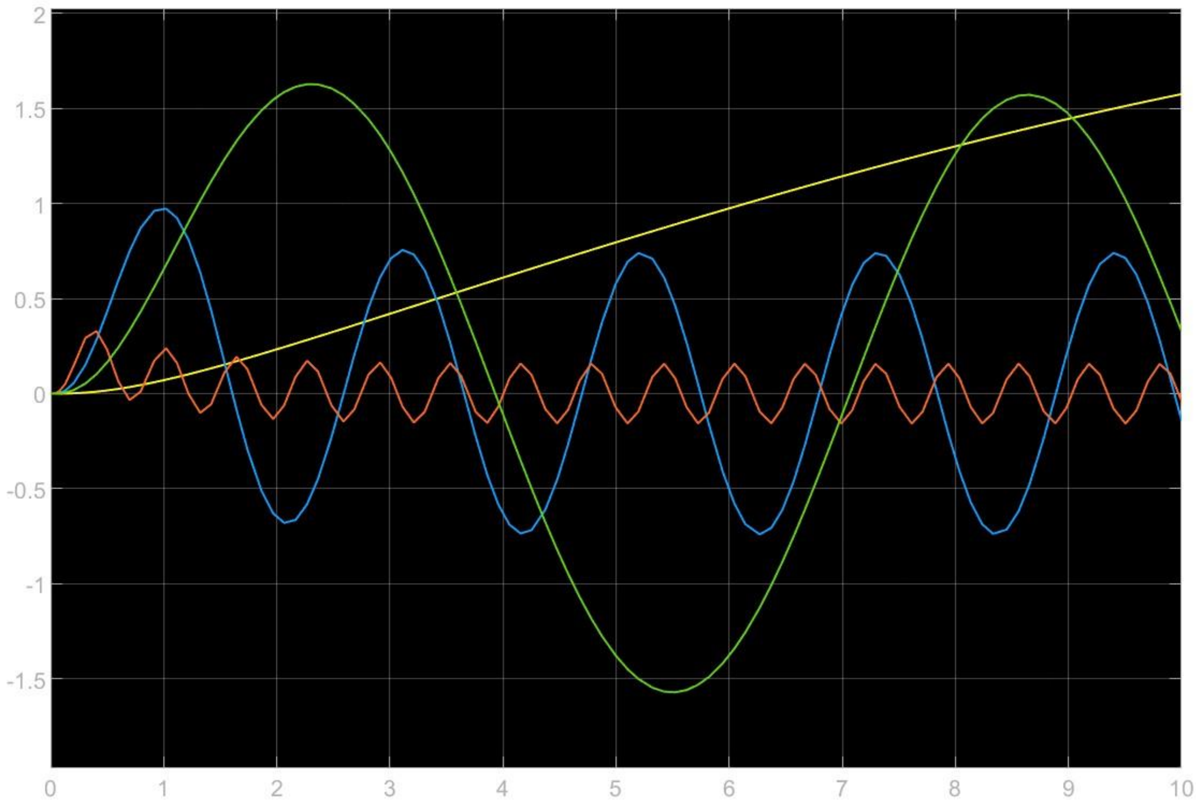
الف و ب) مدل داده شده را در سیمولینک رسم کرده و به آن ورودی پله را اعمال میکنیم. خروجی حاصله را در این مدل حلقه بسته نمایش میدهیم. ورودی پله به همراه خروجی نمایش داده شده است:



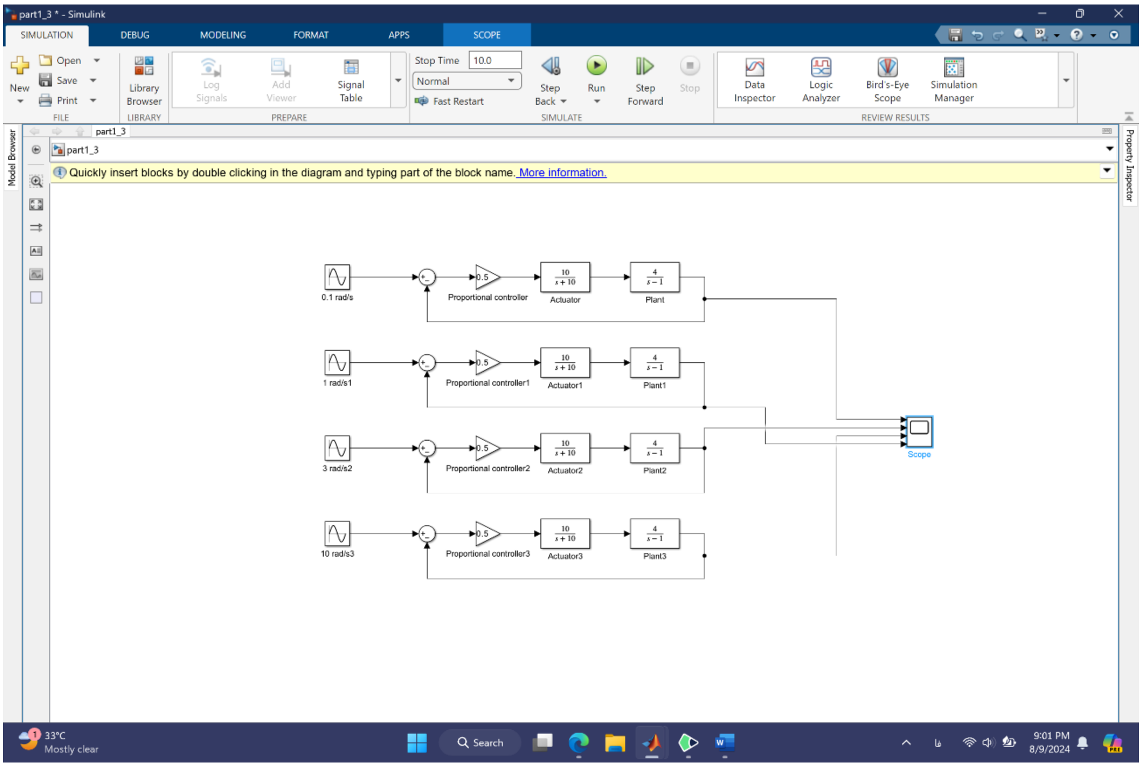


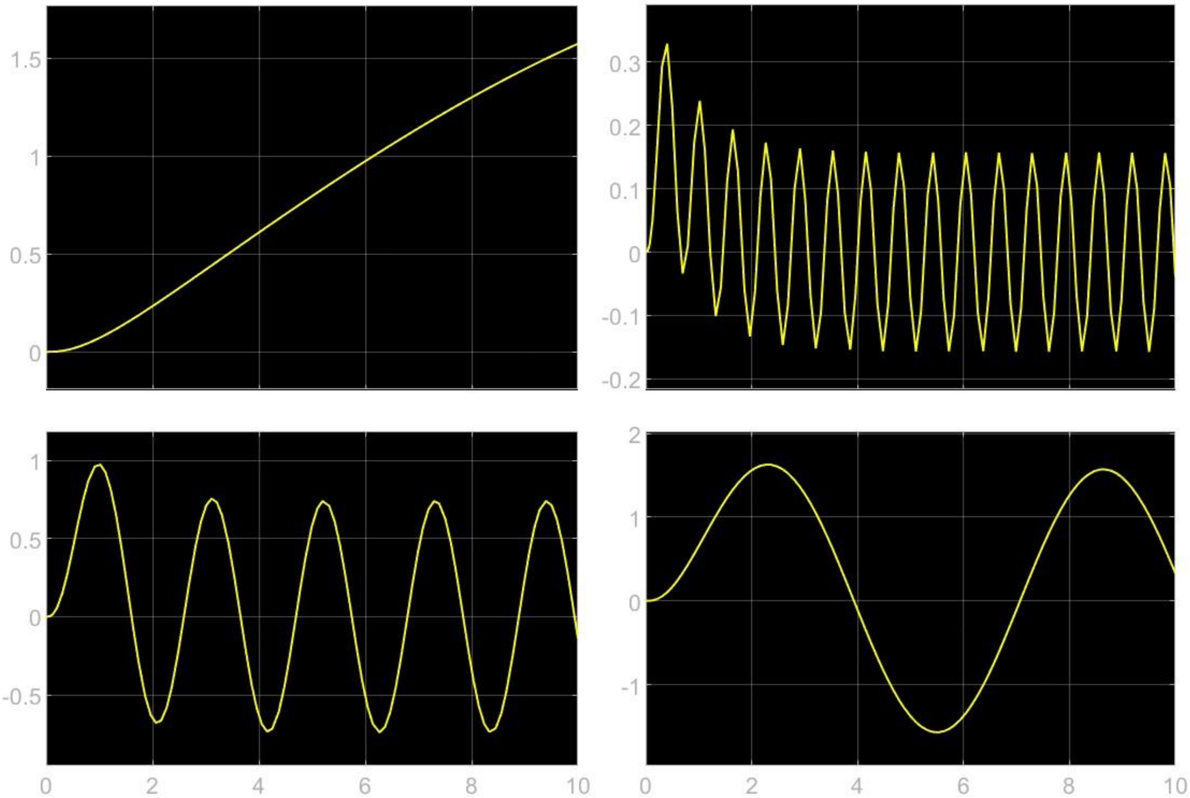
ج) در این بخش ۴ ورودی سینوسی را با فرکانس های ۰.۲، ۱، ۳ و ۱۰ رادیان بر ثانیه با استفاده از Mux به صورت همزمان پلات می کنیم.خروجی به شکل زیر است:





اینبار در ادامه همین بخش، ورودی ها را به صورت جداگانه پلات می کنیم.

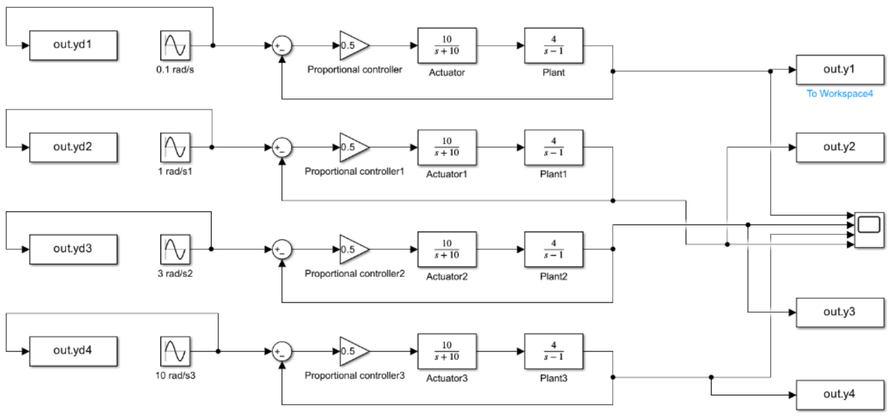


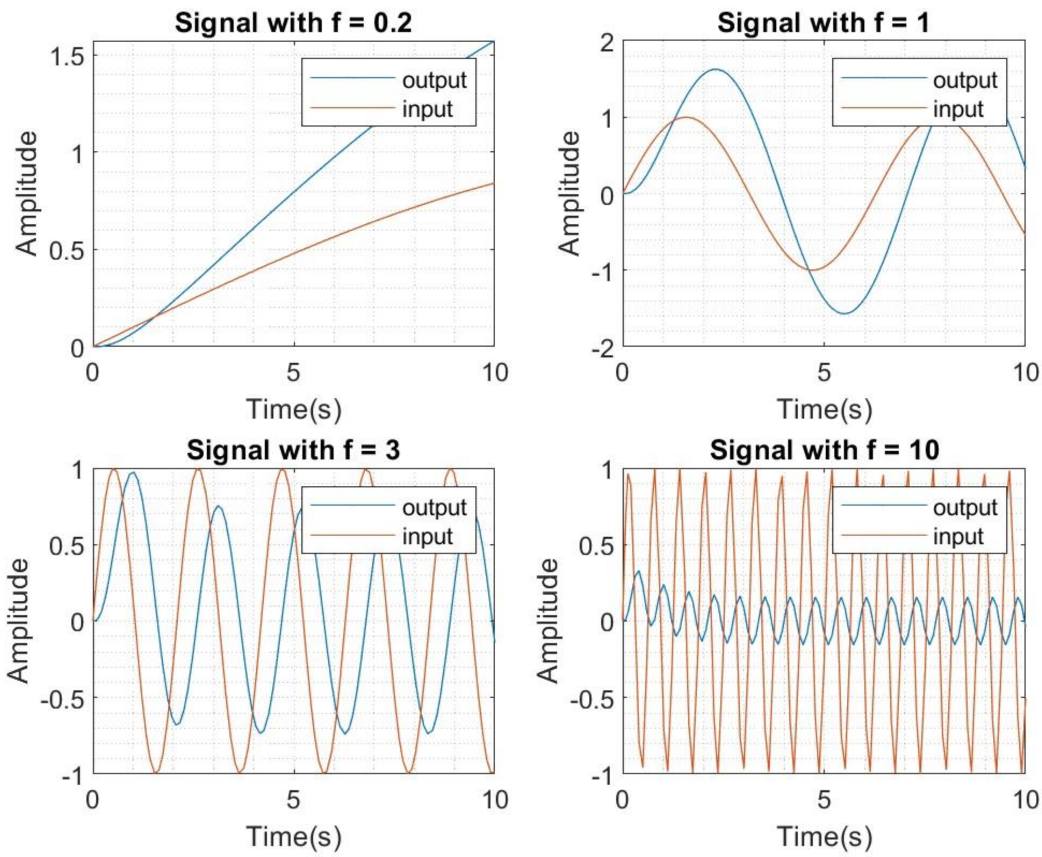


مشاهده می شود که با دامنه یکسان ورودی ها، هر چه فرکانس افزایش می یابد، دامنه خروجی کم تر می شود. ورودی ها نیز بر حسب تعداد نوسان مشخص هستند که چه فرکانسی دارند.

د) در این قسمت با استفاده از بلاک to workspace ورودی ها و خروجی های سیستم را به محیط کار اسکریپت متلب انتقال می دهیم و با استفاده از تکرار کد زیر برای هر ۴ سیگنال ورودی، ورودی و خروجی ها را در نهایت بر روی یک فیگر دو در دو رسم می کنیم که به شکل زیر می شود:

```
figure;
subplot(2,2,1);
plot(out.y1)
hold on
plot(out.yd1)
xlabel('Time(s)')
ylabel('Amplitude')
title('Signal with f = 0.2')
grid minor
legend('output','input')
```



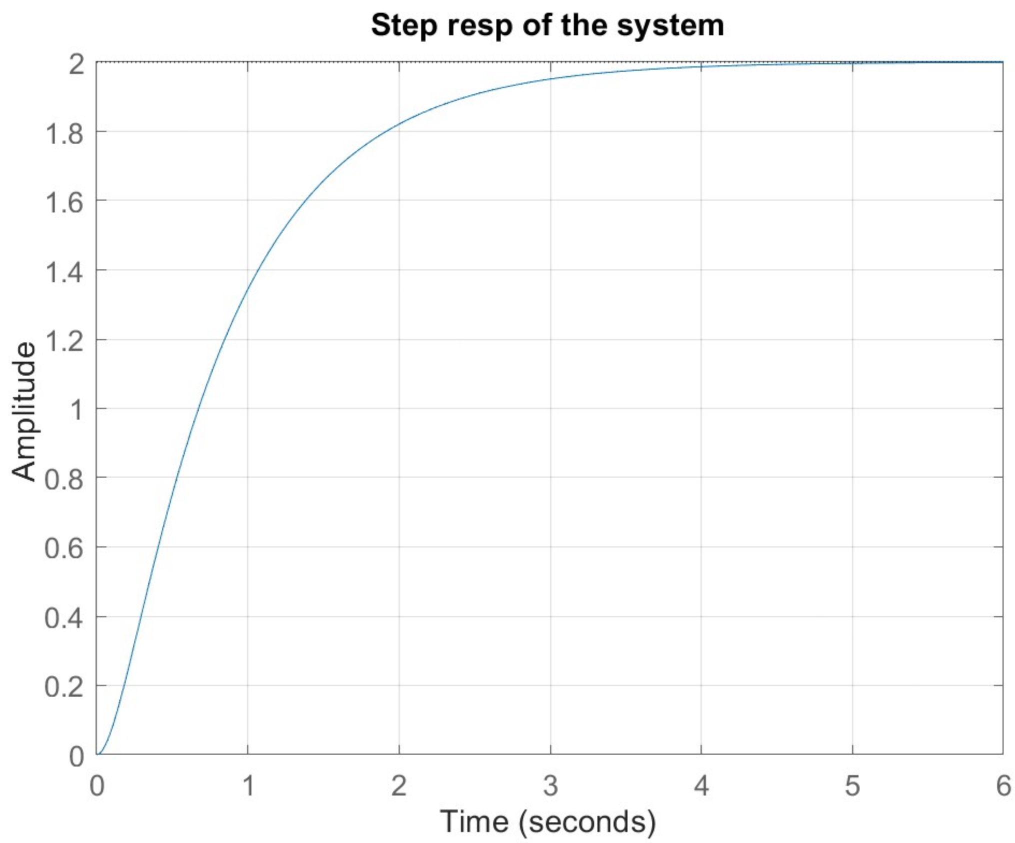


۵) ابتدا توابع actuator و plant را پیاده کرده و سری می کنیم، سپس با استفاده از تابع فیدبک، ترنسفر فانکشن کل سیستم را بدست می آوریم. پس از آن ورودی پله را اعمال کرده و خروجی را چاپ می کنیم.

کد این بخش به همراه خروجی:

```
s = tf('s');
actuator = tf(10, [1 10]);
plant = tf(4, [1 -1]);
T = series(actuator,plant);
trans_func = feedback(0.5*T,1);

step(trans_func)
title('Step resp of the system')
xlabel('Time')
ylabel('Amplitude')
grid minor
```



بخش ۲: طراحی کنترلر

الف) برای طراحی کنترلری که بتواند خطای حالت دائم را به ۰ برساند، راه های مختلفی وجود دارد که حتی با کنترلر p ساده که صرفا یک ضریب را به سیستم اضافه میکند، نیز قابل انجام است ولی در آن صورت باید Kp را به بی نهایت سوق داد. اما با استفاده از کنترل کننده pi که در آزمایشگاه نیز استفاده شد، می توانیم این محقق را راحت تر به سرانجام برسانیم که در بخش بعدی محاسبات آن به صورت کامل آورده شده اند. این کنترل کننده حاوی یک ضریب و یک انتگرال گیر در سیستم با فیدبک است که نیازمند تعیین ضرایب Ki و Kp و همچنین بتا هستیم. البته می توان بتا را صفر در نظر گرفت.

به صورت کلی برای تحقق خطای حالت دائم صفر کنترل کننده pi مناسب است که در بخش بعدی طراحی می شود.

بخش بعدی به صورت دست نویس آماده شده و در آن ضرایب مورد نیاز در کنترل کننده با پیدا شدن رنج تغییرات مناسب آن ها تعیین می شوند.

ب)

$$\text{Pi controller}: \quad \frac{(\beta k_p + \frac{k_i}{s})G(s)}{1+(k_p+\frac{k_i}{s})G(s)} \Rightarrow G(s)=\frac{4}{s-1} \Rightarrow$$

$$H(s)=\frac{(\beta k_p+\frac{k_i}{s})\frac{4}{s-1}}{1+(k_p+\frac{k_i}{s})\frac{4}{s-1}}=\frac{4(\beta k_p+\frac{k_i}{s})}{s-1+4(k_p+\frac{k_i}{s})}=\boxed{\frac{4(\beta k_p s+k_i)}{s^2+(4k_p-1)s+4k_i}}$$

$$\Rightarrow \beta=0 \longrightarrow H(s)=\frac{4k_i}{s^2+(4k_p-1)s+4k_i} \Rightarrow \omega_n^2=4k_i \Rightarrow \boxed{\omega_n=2\sqrt{k_i}}$$

$$2\omega_n\zeta=4k_p-1 \Rightarrow 4\sqrt{k_i}\,\zeta=4k_p-1 \Rightarrow \boxed{\zeta=\frac{4k_p-1}{4\sqrt{k_i}}}$$

$$T_s<4s \Rightarrow \frac{4}{\omega_n\zeta}<4 \Rightarrow \omega_n\zeta>1 \Rightarrow 2k_p-0.5>1 \Rightarrow$$

$$\boxed{k_p>0.75}$$

$$P.O.<10\% \Rightarrow 100\,e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}<10 \Rightarrow \frac{-\pi\zeta}{\sqrt{1-\zeta^2}}<\ln 0.1 \Rightarrow$$

$$\zeta>-\frac{\ln 0.1}{\pi}\sqrt{1-\zeta^2} \Rightarrow \zeta>0.733\sqrt{1-\zeta^2} \Rightarrow$$

$$\zeta^2>0.537(1-\zeta^2) \Rightarrow 1.537\zeta^2>0.537 \Rightarrow \zeta^2>0.35$$

$$\Rightarrow \zeta>0.591 \Rightarrow \boxed{\frac{k_p}{\sqrt{k_i}}-\frac{1}{4\sqrt{k_i}}>0.591}$$

$$e(\infty)=\lim_{s\to 0} s(1-H(s))Y_d(s) \Rightarrow Y_d(s)=U(s)=\frac{1}{s} \Rightarrow e(\infty)=\lim_{s\to 0}(1-H(s))$$

$$=\lim_{s\to 0}\left(1-\frac{4k_i}{s^2+(4k_p-1)s+4k_i}\right)=\lim_{s\to 0}\left(1-\frac{4k_i}{4k_i}\right)=\boxed{0}$$

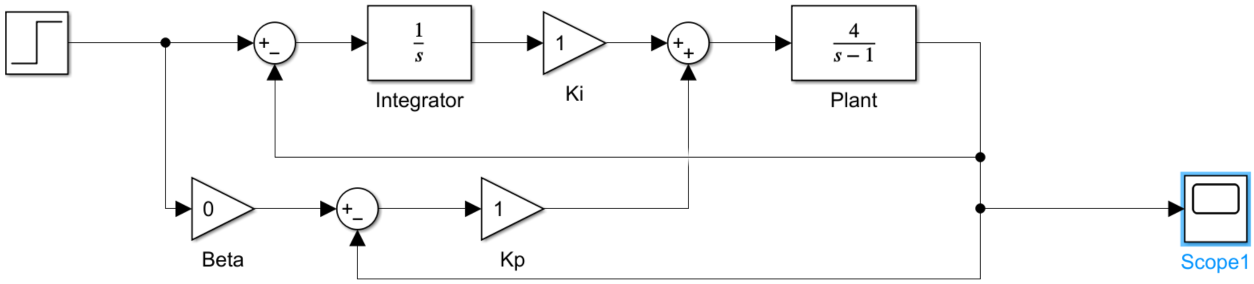
$$k_p>0.75 \Rightarrow \boxed{k_p=1} \qquad \frac{k_p}{\sqrt{k_i}}-\frac{1}{4\sqrt{k_i}}>0.591 \Rightarrow \frac{0.75}{\sqrt{k_i}}>0.591 \Rightarrow$$

$$\sqrt{k_i}<1.27 \Rightarrow k_i<1.6 \Rightarrow \boxed{k_i=1}$$

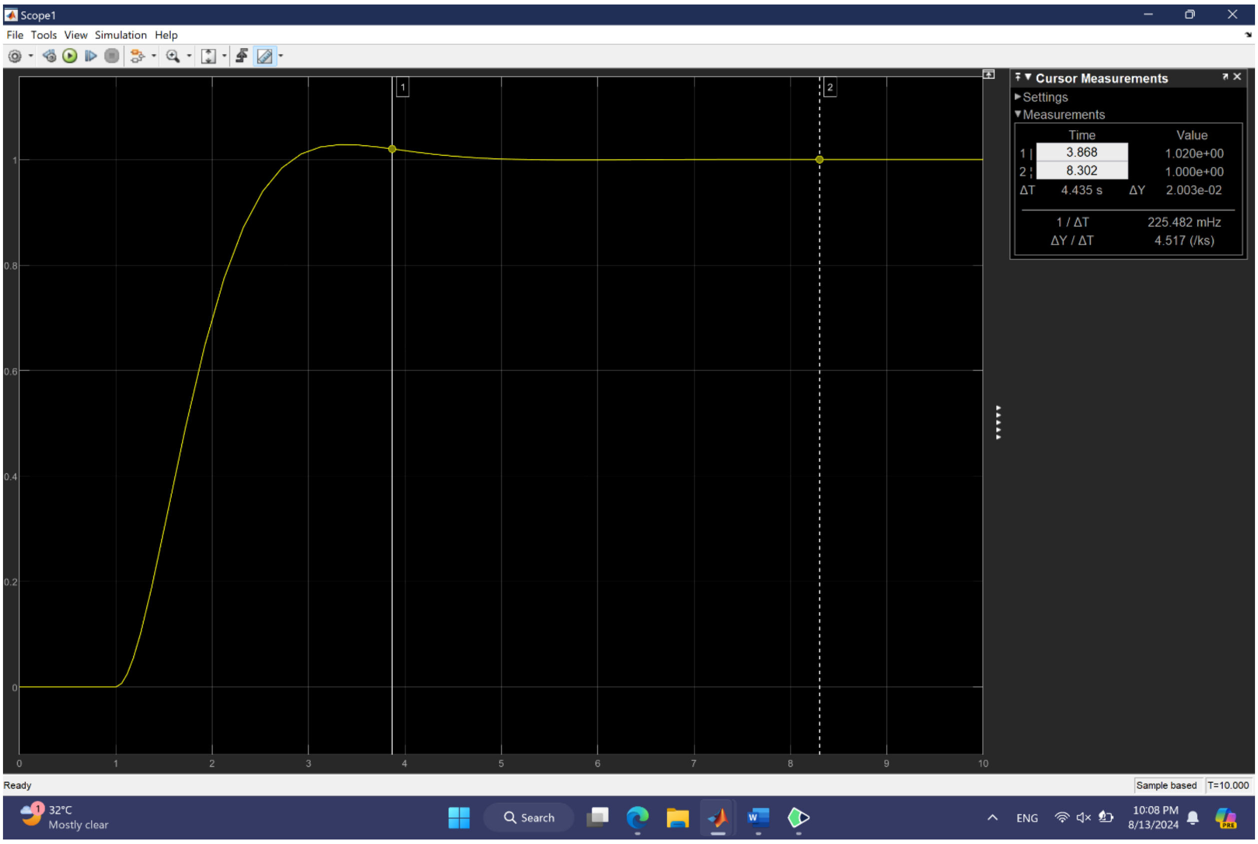
$$\Rightarrow \zeta=0.75,\ \omega_n=2 \Rightarrow \boxed{T_s=2.67} \quad \boxed{P.O.=2.8376\%} \quad \boxed{e(\infty)=0}$$

پس کنترل کننده Pi با $k_p=1$, $k_i=1$ را طراحی می‌کنیم و می‌بینیم که شرط‌ها را برقرار می‌کند.

ج) کنترل کننده بخش قبل را وارد سیمولینک می کنیم و شبیه سازی را با دادن ورودی پله انجام می دهیم. سیستم طراحی شده به همراه خروجی آن:



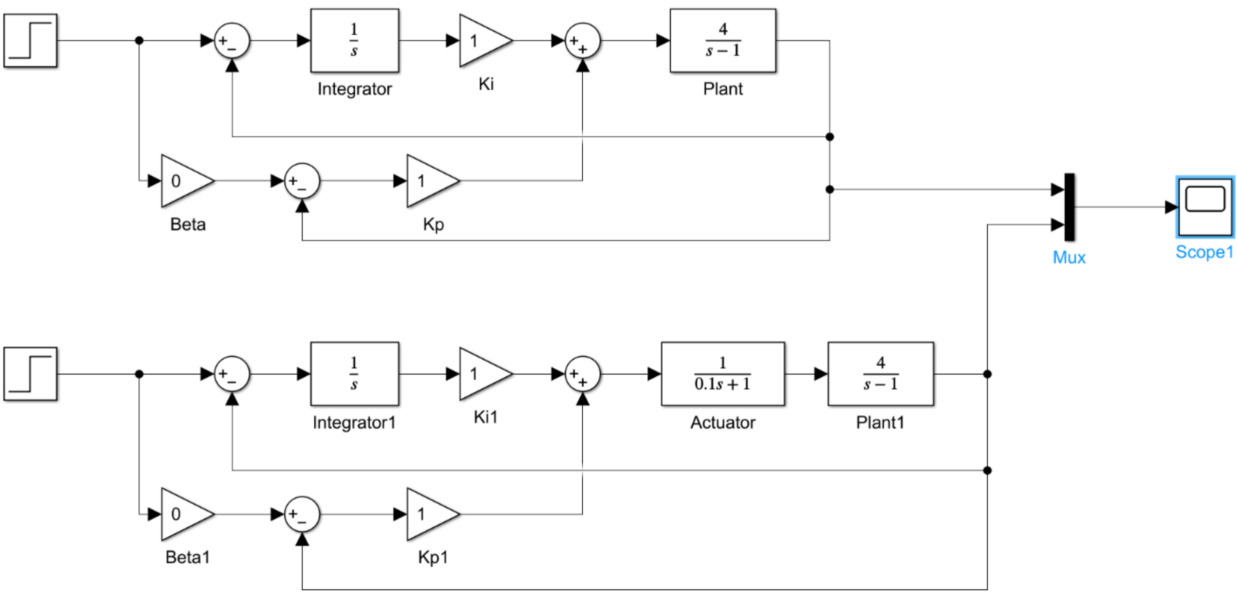
ضرایب و انگرال گیر در سیستم مشخص شده اند.

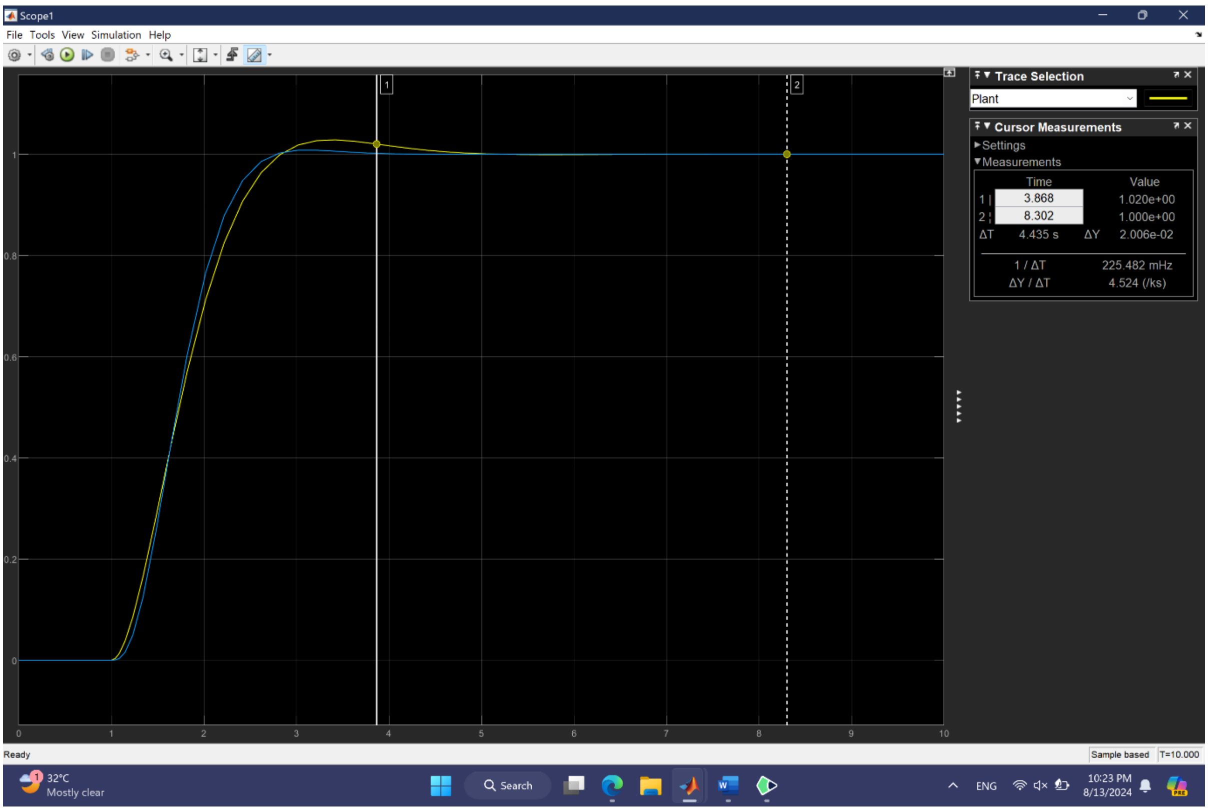


مشاهده می کنیم که اورشوت بسیار کم و در حدود ۳ الی ۵ درصد است که مطلوب است. همچنین با توجه به نقطه مشخص شده، ستلینگ تایم در حدود ۳.۸۷ ثانیه است که زیر ۴ است و شرط را رعایت می کند. همچنین با توجه به نقطه دوم و در ادامه آن می بینیم که خطا به سمت صفر میل پیدا کرده است. در نهایت می بینیم که کنترل کننده بسیار خوب عمل کرده و شروط را پیاده کرده است.

د) با توجه به صحت عملکرد کنترلر بخش قبل به تغییر نیازی نیست و ضرایب Kp و Ki به مقدار ۱ و همچنین ۰ بودن بتا مناسب هستند.

۵) سیستم دوم را با اکچویتور پیاده کرده و خروجی هر دو را در یک پلات مشاهده می کنیم.





نمودار آبی با اکچویتور است که مشاهده می شود تغییر چندانی رخ داده و واکنش ها تقریبا یکسان است زیرا که تابعی با ثابت زمانی کوتاهی (۰.۱) اضافه شده که تاثیر کمی در مدت طولانی دارد. مشاهده می کنیم که کنترل بهتری انجام شده هم به لحاظ اورشوت و هم به لحاظ ستلینگ تایم.