

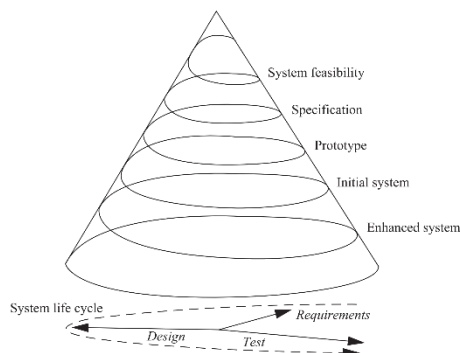
گزارش تمرین ۱

فهرست

| | |
|--|----|
| فهرست | ۲ |
| | ۳ |
| سوال ۱-الف: مدل توسعه نرم افزار Spiral چیست و چه تفاوت ها و شباهت هایی با فرایندهای توسعه معرفی شده در درس دارد؟ | |
| بخش ۱ | ۴ |
| | ۶ |
| سیستم جرم-فنر-دمپر | ۷ |
| بخش ۲ | |
| پیاده سازی سیستم | ۷ |
| شبیه سازی سیستم | ۹ |
| ورودی: عدد ثابت | ۹ |
| ورودی: تابع پله | ۱۰ |
| ورودی: تابع سینوسی | ۱۰ |
| تغییر مکان اولیه و سرعت اولیه | ۱۱ |
| تاثیر ضریب ثابت دمپر بر نوسان جسم | ۱۱ |
| تاثیر جرم بر نوسان جسم | ۱۲ |
| پیاده سازی کنترلر PD | ۱۳ |
| پیاده سازی حلقه بازخوردی سیستم | ۱۳ |
| شبیه سازی سیستم با کنترلر | ۱۴ |
| تاثیر پارامترهای Kp و Kd | ۱۴ |

بخش ۱

سوال ۱-الف: مدل توسعه نرم افزار Spiral چیست و چه تفاوت‌ها و شباهت‌هایی با فرایندهای توسعه معرفی شده در درس دارد؟



مدل spiral یکی از انواع متدهای توسعه سیستم‌های نهفته و سایبرفیزیکی است. این مدل فرآیند توسعه را تبدیل به یک فرآیند چرخه‌ای و تکاملی می‌کند. در هر چرخه بخشی از پروژه طراحی، تست و پیاده‌سازی می‌شود. در چرخه‌های اولیه توسعه، طراحان و مهندسان پروژه به صورت سطح بالا و انتزاعی محصول را طراحی می‌کنند. بعد از طی شدن هر چرخه، سطح انتزاع کاهش می‌یابد و جزئیات بیشتری از محصول طراحی و پیاده‌سازی می‌شود. در نتیجه هر قدر به چرخه‌های پایانی نزدیک‌تر می‌شویم، فرآیند توسعه هر چرخه به خاطر جزئیات بیشتر طولانی‌تر می‌شود. این چرخه‌ای بودن توسعه محصول باعث می‌شود تا مهندسان پروژه دید و درک بهتری از کل محصول پیدا کنند.

در مدل توسعه آبشاری، محصول یکپارچه فرض می‌شد و فرآیند طراحی، پیاده‌سازی و تست آن به صورت یک جا صورت می‌گرفت. در مدل Vee، کل معماری محصول قبل از پیاده‌سازی صورت می‌گرفت. بعد از پیاده‌سازی و تولید یک نسخه اولیه، محصول وارد فرآیند تست می‌شد و هر کجا باگی دیده می‌شد، یک بازگشت به عقب صورت می‌گرفت. ولی در مدل spiral محصول به سطح‌های کلی تا جزئی تقسیم می‌شود. هر چرخه شامل فرآیندهای طراحی، پیاده‌سازی و تست می‌باشد و در هر چرخه یک نسخه از محصول تولید می‌شود.

چرخه‌ای و تکاملی بودن مدل spiral شبیه به مدل Scrum است. در مدل Scrum نیازمندی‌ها به چرخه‌هایی که sprint نامیده می‌شود تقسیم می‌شوند.

سوال ۱-ب: معنی مهندسی هم‌روند (engineering concurrent) در روند طراحی چیست؟ مثال ذکر شده را مختصراً توضیح دهید.

زمانی که پروژه بزرگ می‌شود، افرادی که روی پروژه کار می‌کنند به ناچار مجبور هستند صرفاً روی وظیفه‌ای که برای قسمت مشخصی از پروژه به آن‌ها محول شده است تمرکز کنند. اسناد طراحی و پیاده‌سازی مرحله قبل را از فرد دیگری تحویل می‌گیرند و صرفاً با داشتن نتایج آن‌ها کار خود را پیش می‌برند و تعاملی بین این دو نفر صورت نمی‌گیرد.

مهندسی همروند می‌خواهد فرآیند توسعه را بهبود ببخشد. زمان صرف شده برای طراحی را کاهش بدهد و اتفاق "طراحی کن و بده بغلی" را از بین ببرد. مهندسی همروند می‌خواهد بین تیم‌ها با وظایف مختلف مثل طراح، مهندس نرم‌افزار و سخت‌افزار، بازاریاب و... تعامل بیشتری برقرار کند.

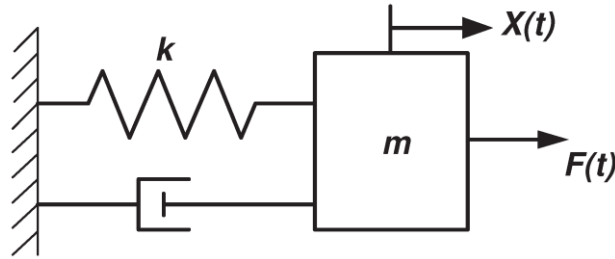
مهندسی همروند می‌خواهد زیرسیستم‌های مختلف پروژه به صورت همروند پیاده‌سازی شوند تا زمان صرف شده کاهش پیدا کند. پیاده‌سازی همروند چالش به اشتراک گذاری تجربه را به وجود می‌آورد. زمانی که پروژه ترتیبی پیاده‌سازی می‌شد، نتایج و تجربه‌ها به تیم بعدی برای توسعه منتقل می‌شد ولی در پیاده‌سازی همروند باید ساز و کاری مشخص و مدیریت شده برای انتقال اطلاعات و تجربه‌ها بین تیم‌ها صورت بگیرد.

در مثال ذکر شده، شرکت AT&T در پروژه ساخت Telephone switching system با چالش زمانی مواجه بوده است و می‌خواست با بهبود و باز طراحی متد توسعه محصول خود، زمان طراحی را کاهش دهد و کیفیت را بالا ببرد. این شرکت ۷ گام را طی کرده است که به اختصار توضیح داده می‌شود:

این شرکت در مقایسه با رقبای خودش ۳۰٪ سرعت طراحی پایین‌تری داشته است. برای بهبود فرآیند توسعه، مدیران و آنالیزورهایی به تیم اضافه کرد تا فرآیند توسعه بین تیم‌های مختلف بهتر مدیریت شود. سپس با بررسی روندنمای طراحی و تولید محصول کنونی به اشکالات آن پی بردند. از جمله این که در اکثر بخش‌ها، تسک‌ها به صورت ترتیبی بین تیم‌ها انجام می‌شدند؛ افراد داخل یک تیم تصمیماتی می‌گرفتند که نتایج و عواقب آن را روی بخش‌های دیگر پروژه نمی‌سنجیدند؛ تسک‌هایی وجود داشت که به نفرات قبلی وابسته بود و تا زمانی که یک تسک کامل نشده بود، نفر بعدی نمی‌توانست آن را انجام دهد. در نهایت مدل توسعه جدید را معرفی کردند و بعد از تست آن با یک پروژه فرضی، از آن در خط تولید محصول اصلی استفاده کردند. این متد توسعه جدید، زمان طراحی و تولید را از ۱۸-۳۰ ماه به ۱۱ کاهش داد.

بخش ۲

سیستم جرم-فنر-دمپر



معادله مدل یک سیستم جرم-فنر-دمپر:

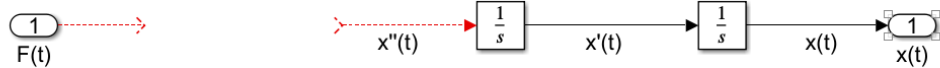
$$F(t) = k \cdot x(t) + b \cdot \dot{x}(t) + m \cdot \ddot{x}(t)$$

پیاده‌سازی سیستم

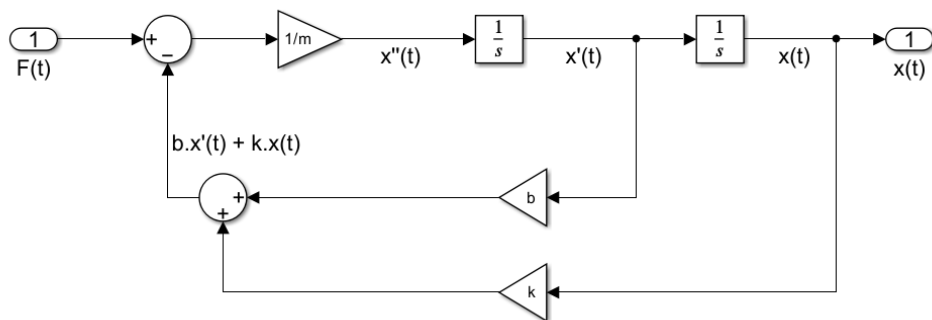
جهت پیاده‌سازی مدل اشاره شده، ابتدا ورودی و خروجی سیستم را مشخص می‌کنیم.

| ورودی | خروجی |
|--------|--------|
| $f(t)$ | $x(t)$ |

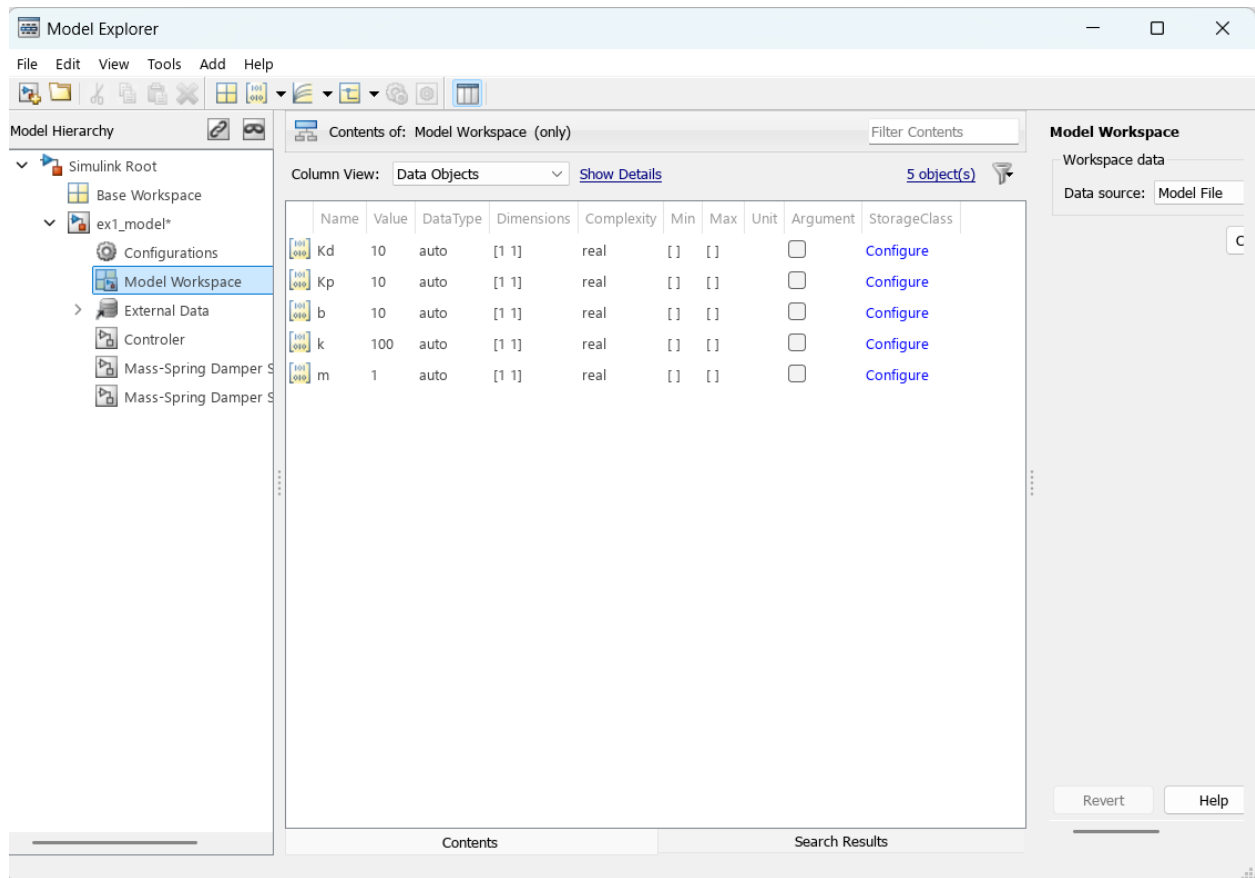
سیگنال‌های \dot{x} و \ddot{x} را می‌توان با استفاده از دو قطعه انتگرال‌گیر ساخت. حال کافیست در معادله سیستم، \ddot{x} را تنها کنیم و با اجزای دیگر، سیگنال را بسازیم.



$$\ddot{x}(t) = \frac{1}{m} (F(t) - b \cdot \dot{x}(t) - k \cdot x(t))$$



مقادیر ثابت را به صورت متغیر تعریف میکنیم و از قسمت Model Workspace به آنها مقدار می‌دهیم.



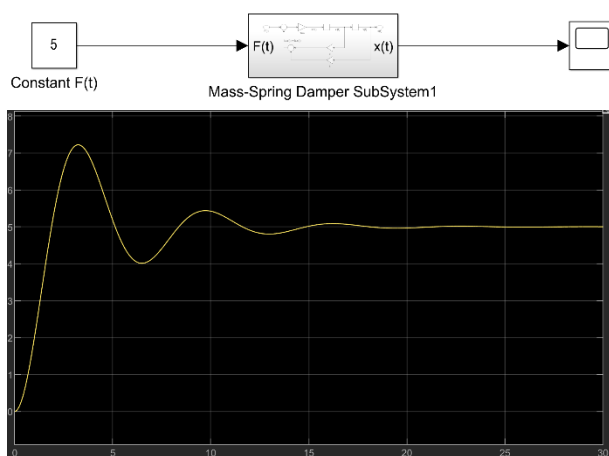
| b | k | m |
|-------------------|-------------------|-------|
| $10 \frac{Ns}{m}$ | $100 \frac{N}{m}$ | $1kg$ |

شبیه‌سازی سیستم

ورودی: عدد ثابت

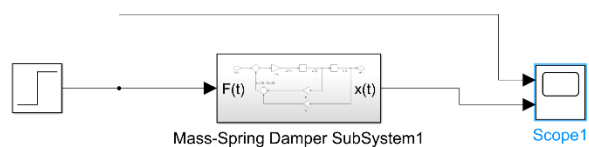


در این قسمت از شبیه‌سازی، عدد ثابت صفر نیوتن به عنوان نیروی ورودی به سیستم اعمال می‌شود. این بدان معنا است که سیستم باید حالت پایداری خود را حفظ کند. با مشاهده نمودار جابجایی سیستم، ساکن بودن جسم دیده می‌شود.

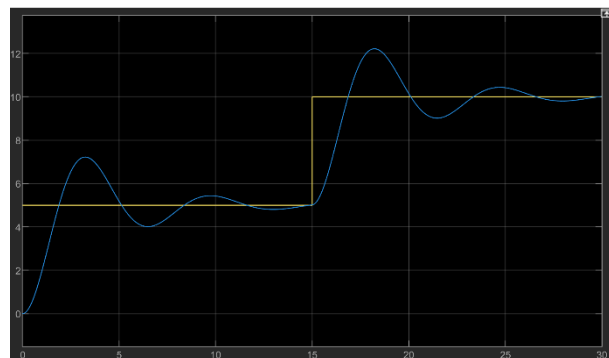


اگر نیرویی ثابت مخالف صفر به سیستم اعمال شود، پس از یک نوسان، سیستم باید در نقطه‌ای غیر از نقطه تعادل به پایداری برسد.

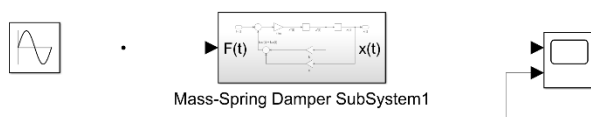
ورودی: تابع پله



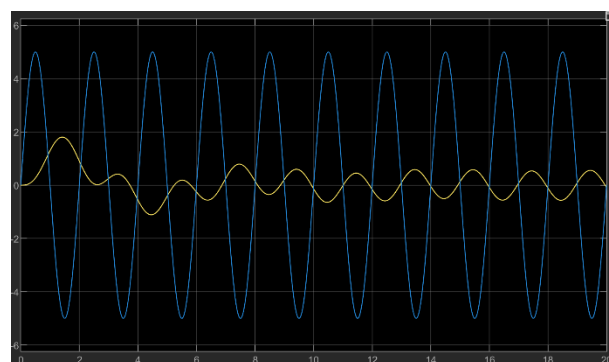
به کمک تابع پله می‌توان رفتار سیستم را هنگام تغییر مقدار نیروی ورودی بهتر مشاهده کرد. در این شبیه‌سازی ابتدا نیروی ۵ نیوتونی و سپس نیروی ۱۰ نیوتونی به سیستم اعمال شده است.



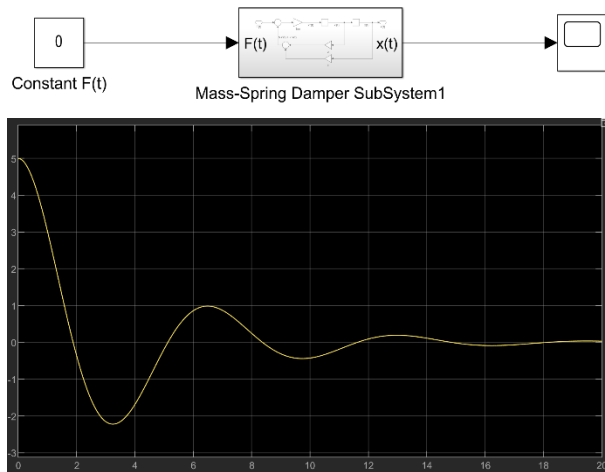
ورودی: تابع سینوسی



در این شبیه‌سازی نیرو به صورت تابع سینوسی $5\sin(\pi t)$ به سیستم وارد شده است.



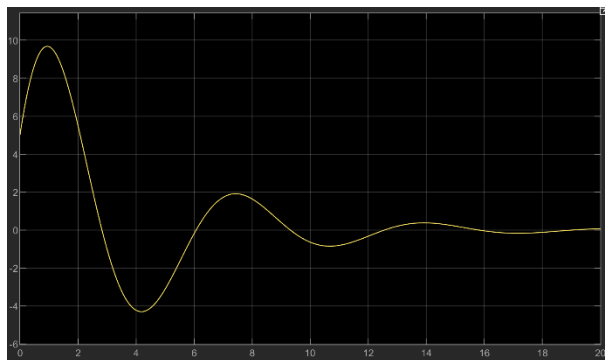
تغییر مکان اولیه و سرعت اولیه



در این شبیه‌سازی به سیستم نیروی خارجی وارد نشده است ولی هنگام شروع، جسم در نقطه تعادل خود قرار نداشته است. مشاهده می‌شود که جسم تمایل به حفظ پایداری در نقطه تعادل دارد.

مکان اولیه: ۵

سرعت اولیه: صفر

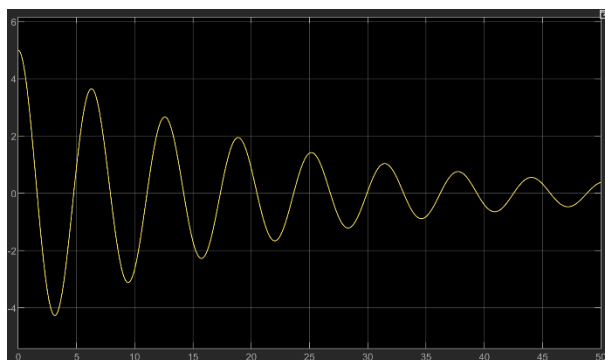


حال اگر به این جسم سرعت اولیه هم داده شود، مشاهده می‌شود که بر خلاف شبیه‌سازی بالایی، به دور شدن از نقطه تعادل خود ادامه می‌دهد تا اینکه دوباره پایدار می‌شود.

مکان اولیه: ۵

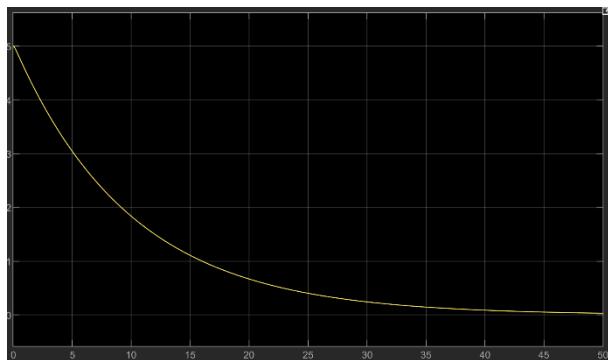
سرعت اولیه: ۱۰

تأثیر ضریب ثابت دمپر بر نوسان جسم



جسمی که بدون دمپر تنها به فنر متصل شده باشد، برای همیشه به نوسان خود ادامه می‌دهد. پس اگر ضریب ثابت دمپر را کم کنیم، جسم راحت‌تر نوسان می‌کند و دیرتر پایدار می‌شود. مانند این شبیه‌سازی:

نیروی خارجی و سرعت اولیه = صفر، مکان اولیه = ۵ و ضریب دمپر = ۰.۱



در شبیه‌سازی دوم، ضریب ثابت دمپر ۱۰۰ برابر شد یعنی برابر ۱۰ مقداردهی شد. مشاهده می‌شود که دمپر سفت شده و از نوسان جسم بیشتر جلوگیری می‌کند.

تاثیر جرم بر نوسان جسم

با ۱۰ برابر کردن جرم، جابجایی ۱۰ برابر شده است.



$$m=10 - ۱$$

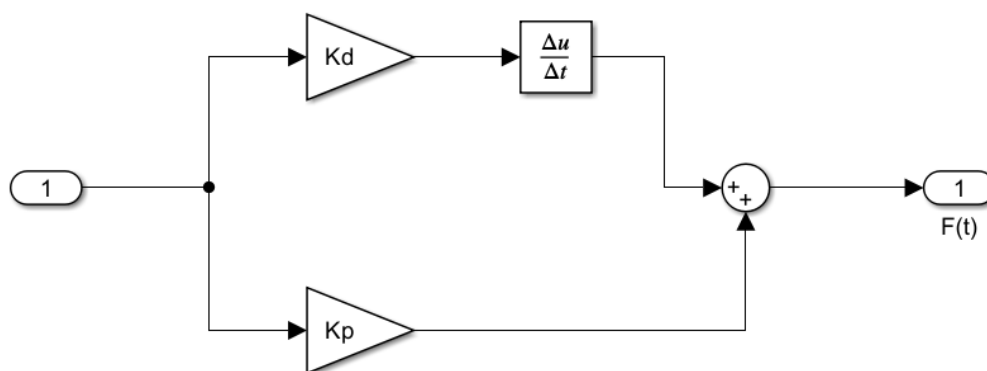


$$m=1 - ۲$$

پیاده سازی کنترلر PD

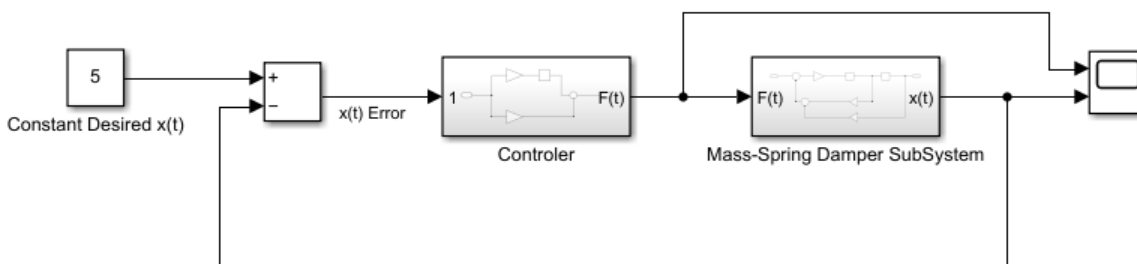
معادله کنترلر PD به صورت زیر است (منفی‌ها به داخل ضریب منتقل شدند):

$$F(t) = -K_p \cdot x(t) - K_d \cdot \dot{x} \rightarrow F(t) = K_p \cdot x(t) + K_d \cdot \dot{x}$$

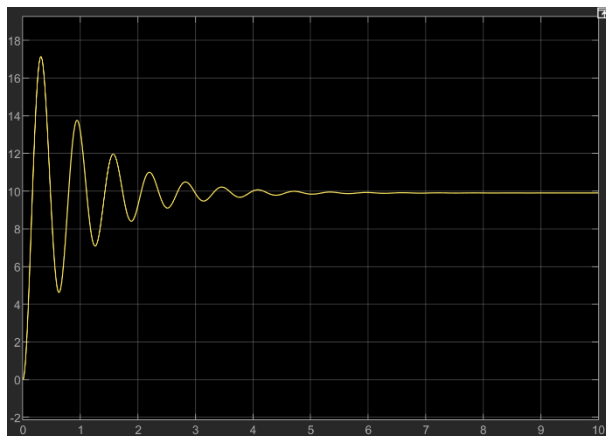


پیاده سازی حلقه بازخوردی سیستم

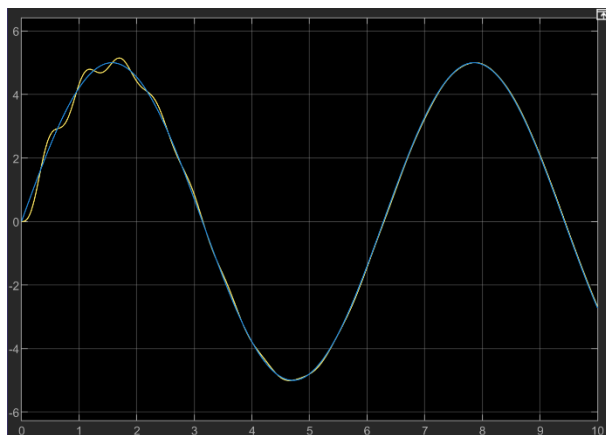
با متصل کردن زیرسیستم‌های پیشین و کم‌کننده به یکدیگر، سیگنال خطا و حلقه بازخورد را تشکیل می‌دهیم.



شبیه‌سازی سیستم با کنترلر



در اینجا مقدار نقطه پایداری مطلوب به سیستم عدد ثابت ۱۰ داده شده بود. مشاهده می‌شود کنترلر با مقادیر مناسب K_p و K_d جسم را در نقطه مورد مطلوب پایدار کرده است.

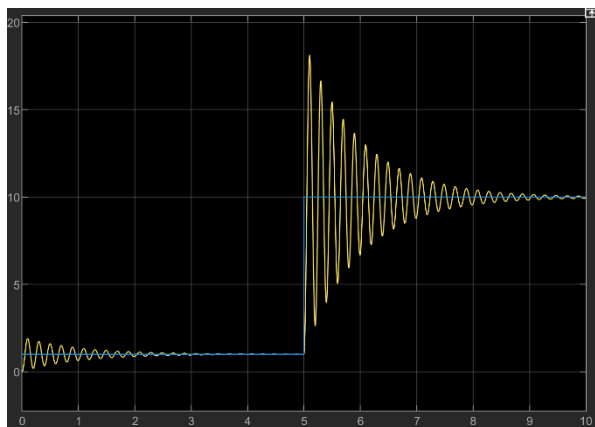


در شبیه‌سازی دیگری، این بار مکان مطلوب جسم یک تابع سینوسی بوده است. اینجا هم مشاهده می‌شود که کنترلر توانسته خروجی مطلوب را تولید کند.

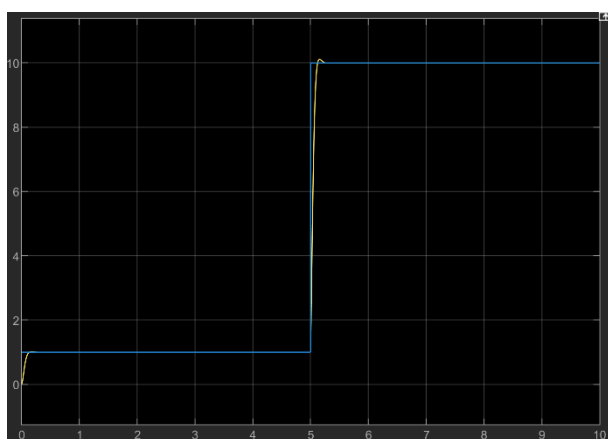
تاثیر پارامترهای K_p و K_d



پارامتر K_p یک gain برای ارور است. هر قدر مقدار این پارامتر بیشتر باشد، تاثیر ارور بیشتر است و کنترلر واکنش سریع‌تری نشان می‌دهد. به طور مثال تمام شبیه‌سازی‌ها بالا با مقدار $K_p = 100$ انجام شد. حال اگر همان شبیه‌سازی با $K_p = 5$ انجام شود، اتفاق زیر رخ می‌دهد:



بیشتر شدن بیش از حد این پارامتر می‌تواند باعث بروز خطا هم شود. در اینجا مقدار $K_p = 1000$ است. مشاهده می‌شود که تابع پله شوک شدیدی به سیستم وارد کرده است.



پارامتر K_d روی مشتق سیگنال جابجایی اعمال شده است. این پارامتر نرخ تغییرات ارور را تاثیر می‌دهد. آینده‌ی ارور سیستم را محاسبه می‌کند و می‌تواند پایداری سیستم را افزایش دهد. حال اگر به همان شبیه‌سازی بالا $K_d = 50$ اعمال شود، از واکنش سریع کنترلر به ارور جلوگیری می‌کند و خروجی کنترلر را نرم‌تر می‌کند.