<< به نام خالق دوست >>

دانشگاه شهید بهشتی
دانشکده مهندسی و علوم کامپیوتر

تمرین سری ۱

مبانی سیستم‌های نهفته و بی‌درنگ

** پاسخ سوال یک بعد از سوال دو قرار گرفته است. **

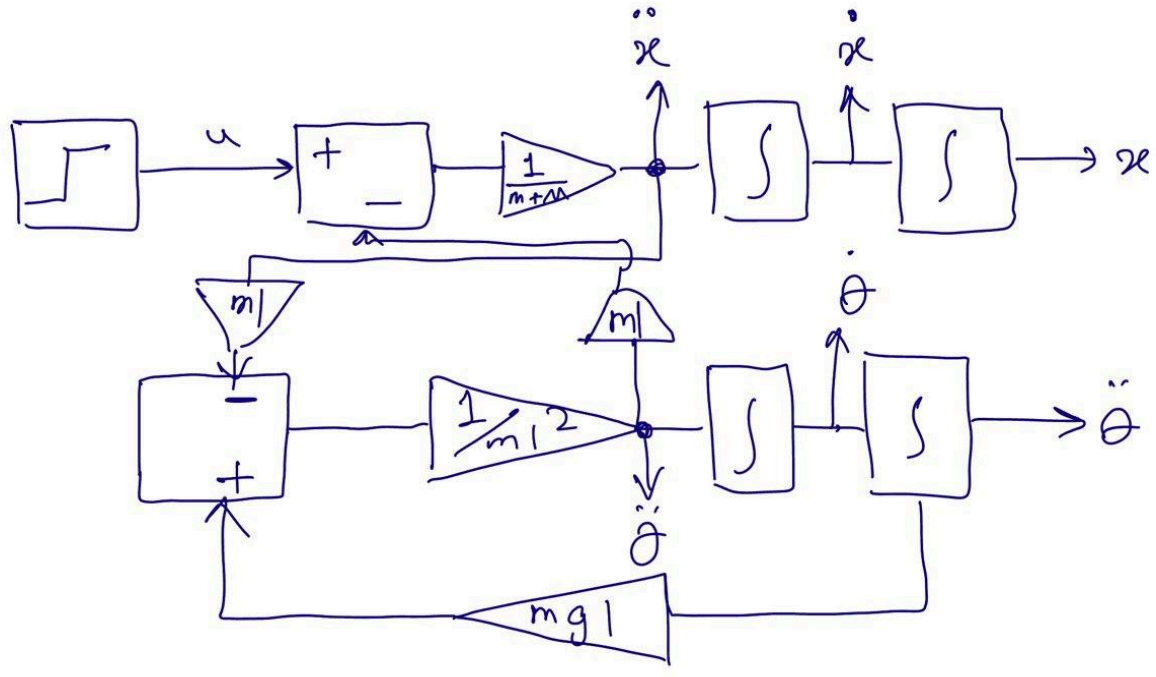
سوال 2:

الف) در این مرحله می‌خواهیم سیستم plant را پیاده سازی کنیم. برای اینکار با استفاده از دو فرمول معادله‌هایی مینویسیم که در دو طرف مشتق دوم θ و x قرار دارند. به عبارتی داریم:

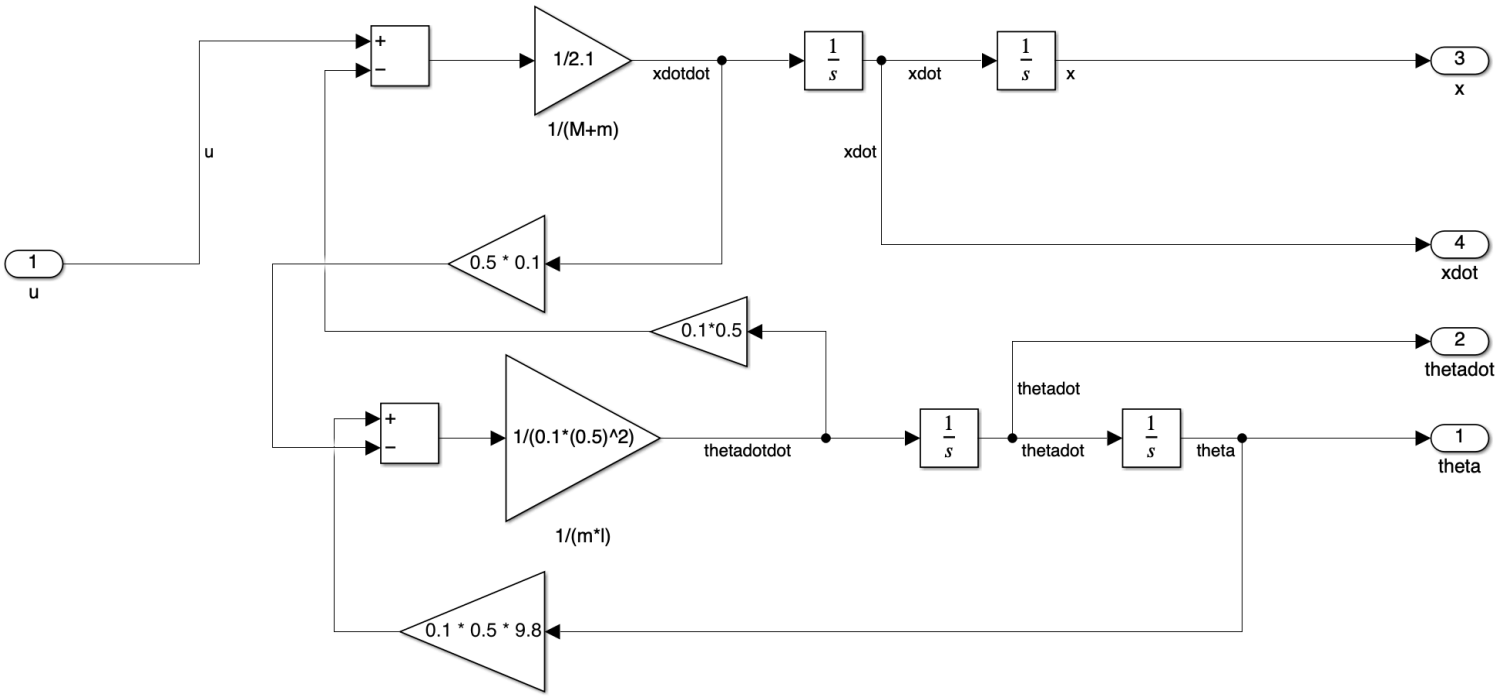
$$\frac{u - \ddot{\theta} lm}{m + M} = \ddot{x}$$

$$\frac{\theta lgm - \ddot{x} lm}{ml^2} = \ddot{\theta}$$

به این ترتیب با قرار دادن یک انتگرال گیر با ورودی مشتق دوم θ و x بار اول به مشتق θ و x و با قرار دادن یک انتگرال گیر دیگر به θ و x میرسیم. در نتیجه حالا که تمام پارامترهای لازم برای مدلسازی سیستم را داریم میتوانیم آن را شبیه سازی کنیم. ابتدا شماتیک کلی از سیستم میکشیم و سپس آن را در سیمولینک رسم میکنیم:



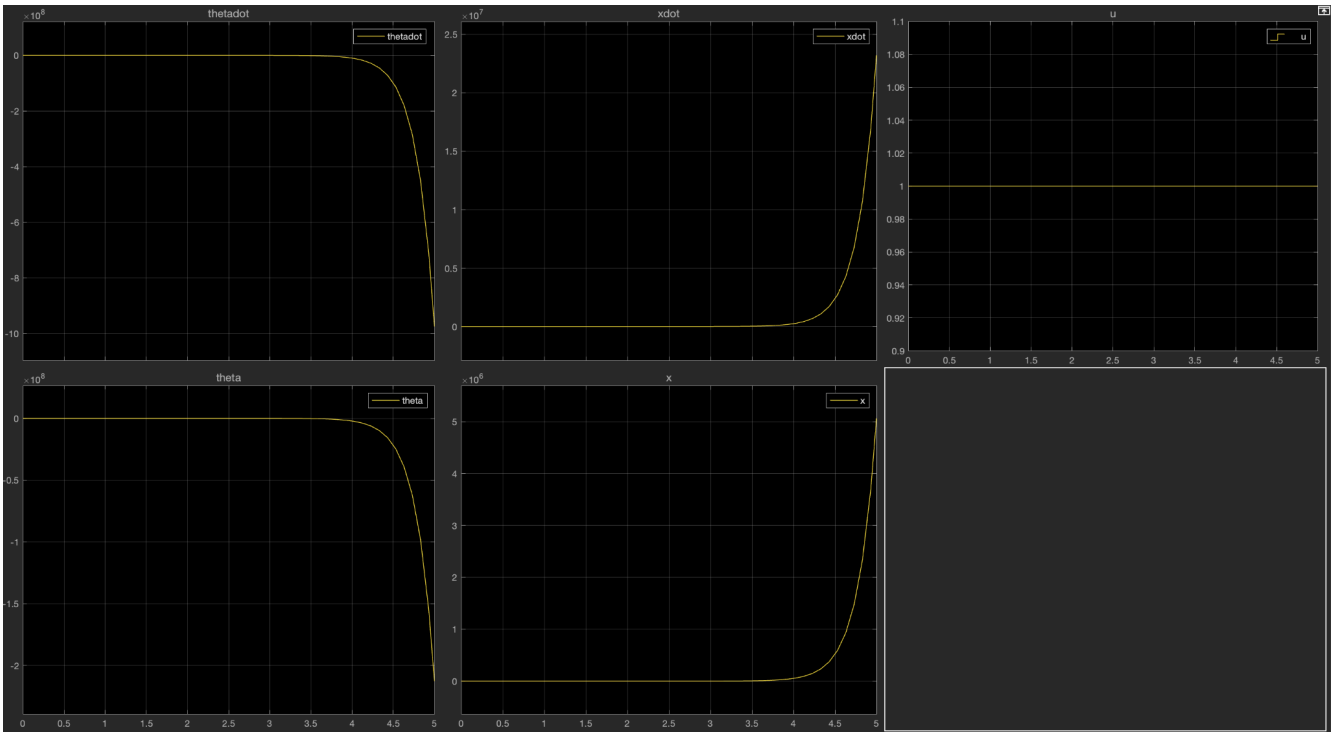
مدل سیمولینک نیز طبق این شماتیک به این شکل است:

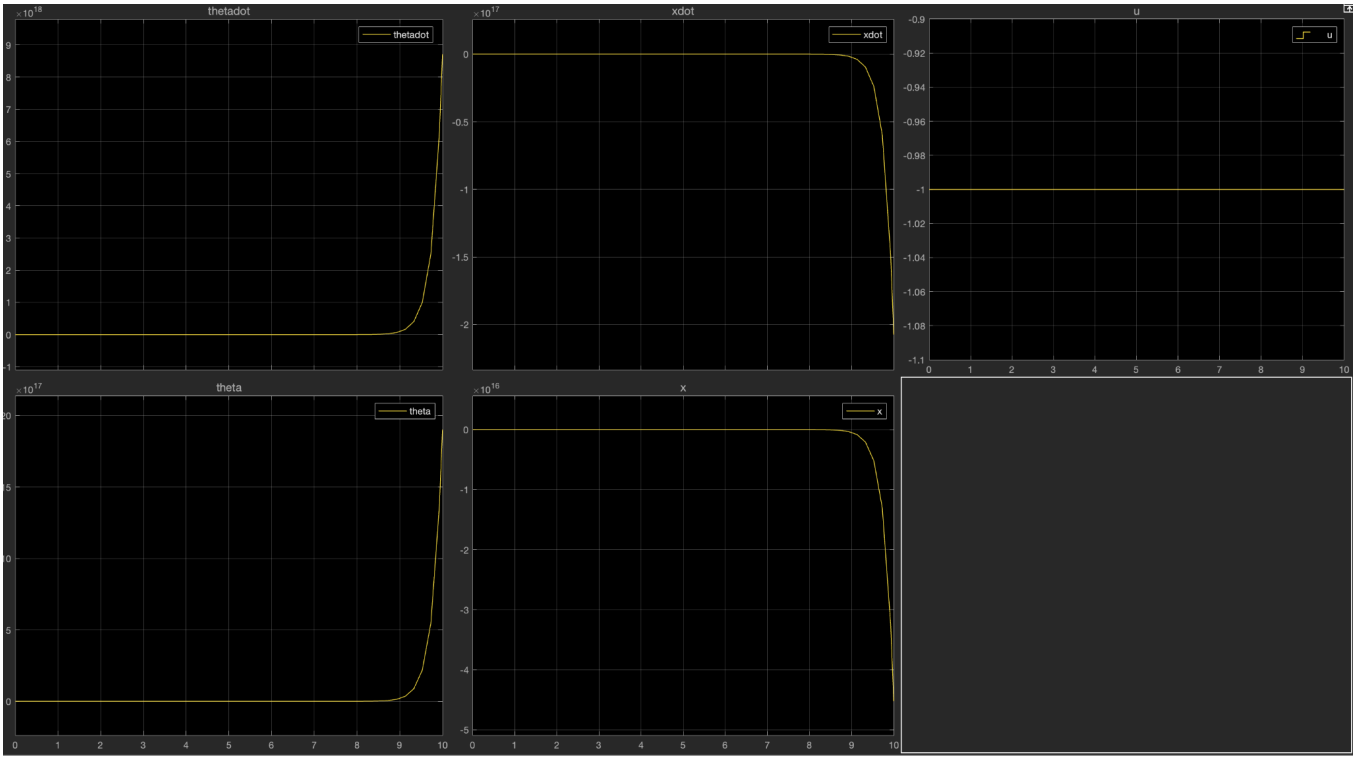


در صورت اجرای شبیه ساز به ازای ورودی‌های نیروی مختلف داریم:

- **تابع ثابت:**

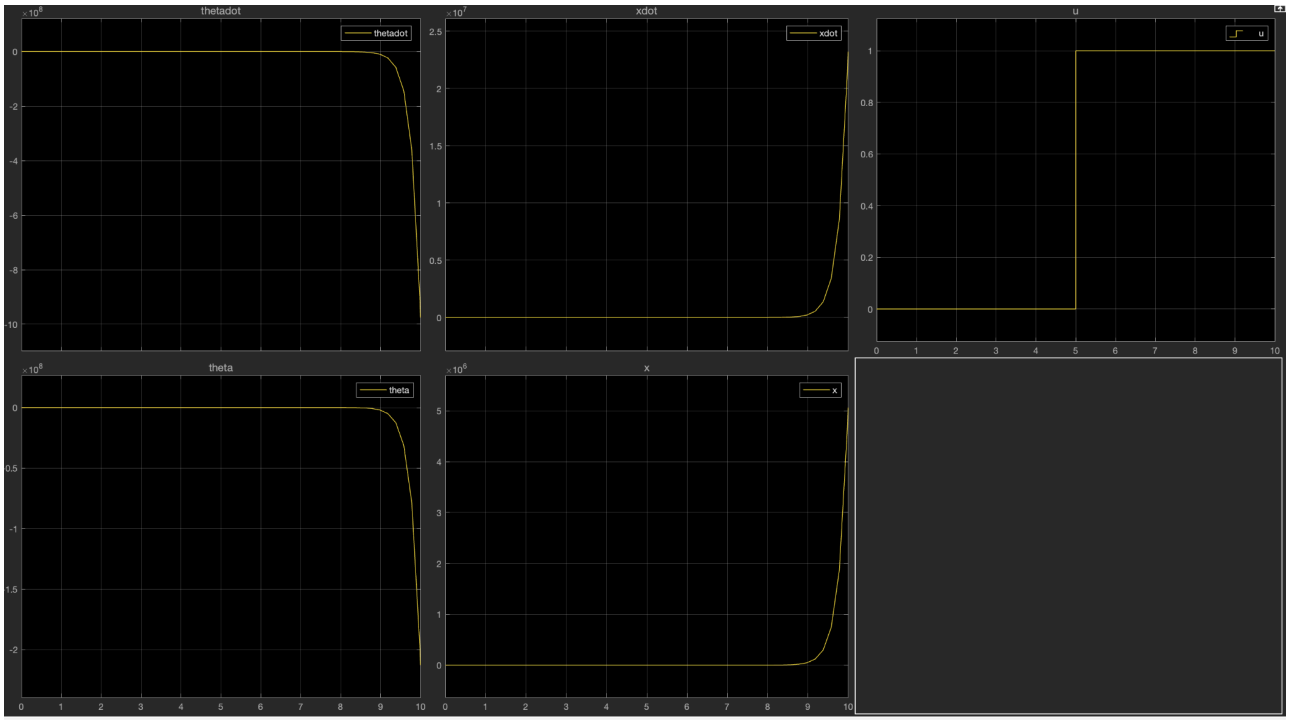
همانطور که دیده میشود چون هیچ سیستم کنترلی نداریم، سرعت و به واسطه آن جابجایی (x و مشتق x) به شدت تغییر میکند. به تبع این جابجایی، زاویه و سرعت زاویه‌ای نیز در جهت مخالف جابجایی (طبق اینرسی) منفی میشود یعنی به سرعت در جهت مخالف می‌چرخد. برای یک عدد ثابت منفی همانطور که دیده میشود جهت جابجایی و چرخش پاندول برعکس است.





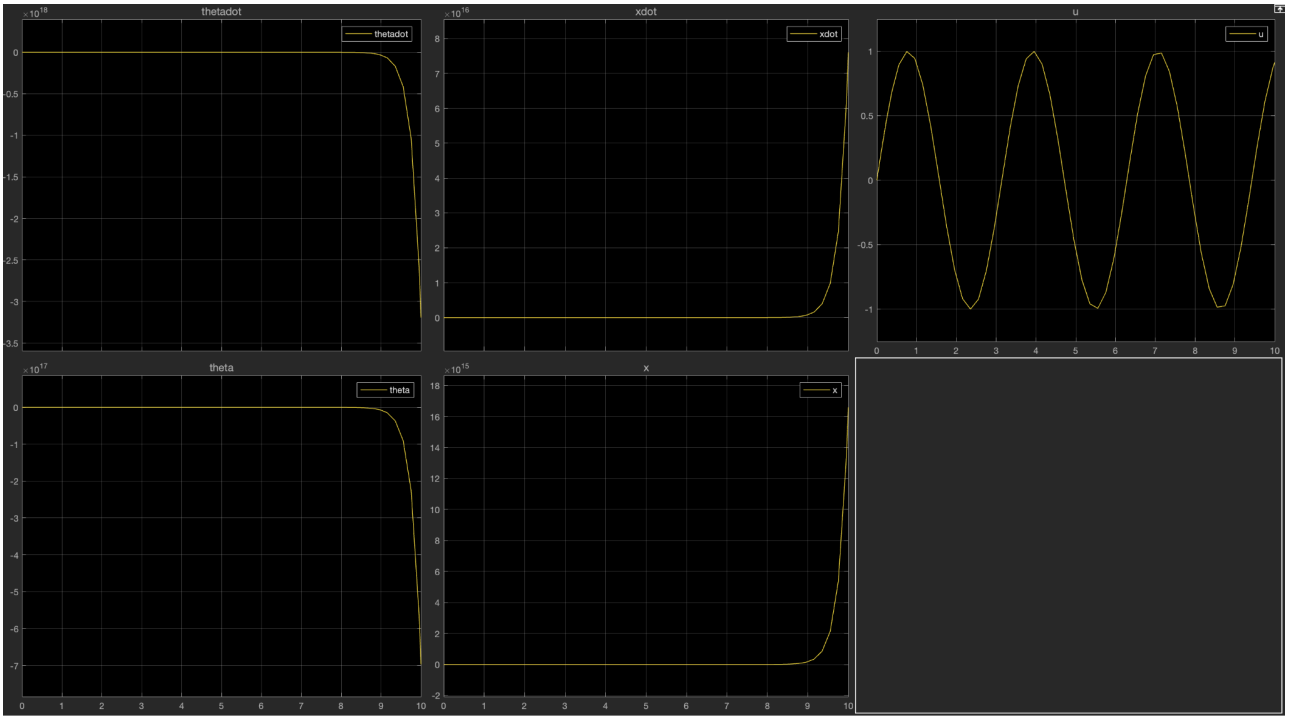
- **تابع پله‌ای:**

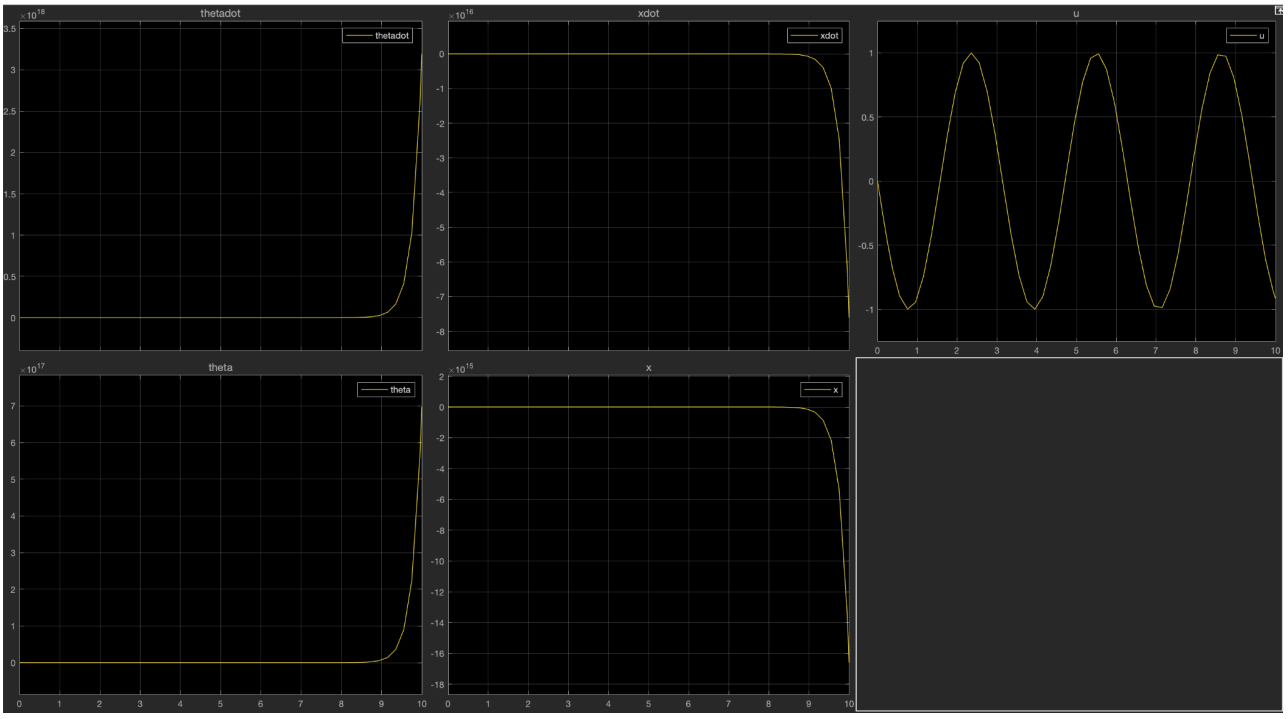
تابع پله‌ای استفاده شده در ثانیه 5 از مقدار صفر به مقدار 1 میرسد و با یک تاخیری این تاثیر در جابجایی و زاویه پاندول نشان داده میشود. در اینجا نیز با توجه به نبود کنترلر جابجایی و زاویه با سرعت فزاینده (مشتقات هم زیاد میشوند) افزایش پیدا میکنند.



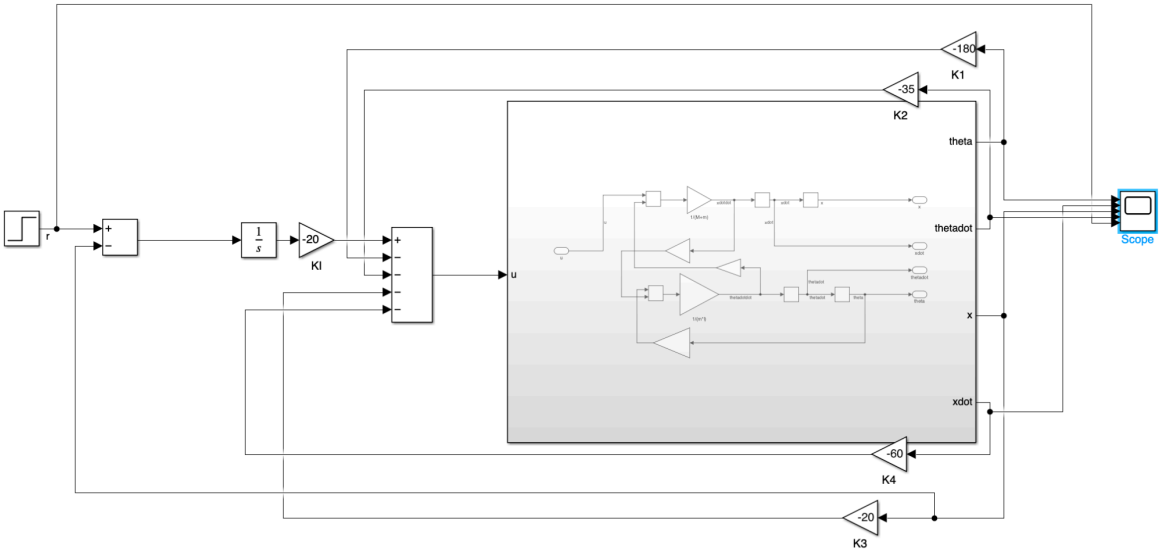
- تابع سینوسی:

در دو تصویر زیر توابع سینوسی به سیستم داده شده است که یکی از آنها ضریب -۱ دارد, همانطور که در تصاویر مشخص می‌باشد با تغییر در جهت شروع تابع سینوسی مشاهده می‌شود که جهت حرکت گاری نیز تغییر می‌کند. در این مثال‌ها نیروی سینوسی اولیه که به سیستم وارد می‌شود جهت نهایی حرکت گاری را مشخص می‌کند.



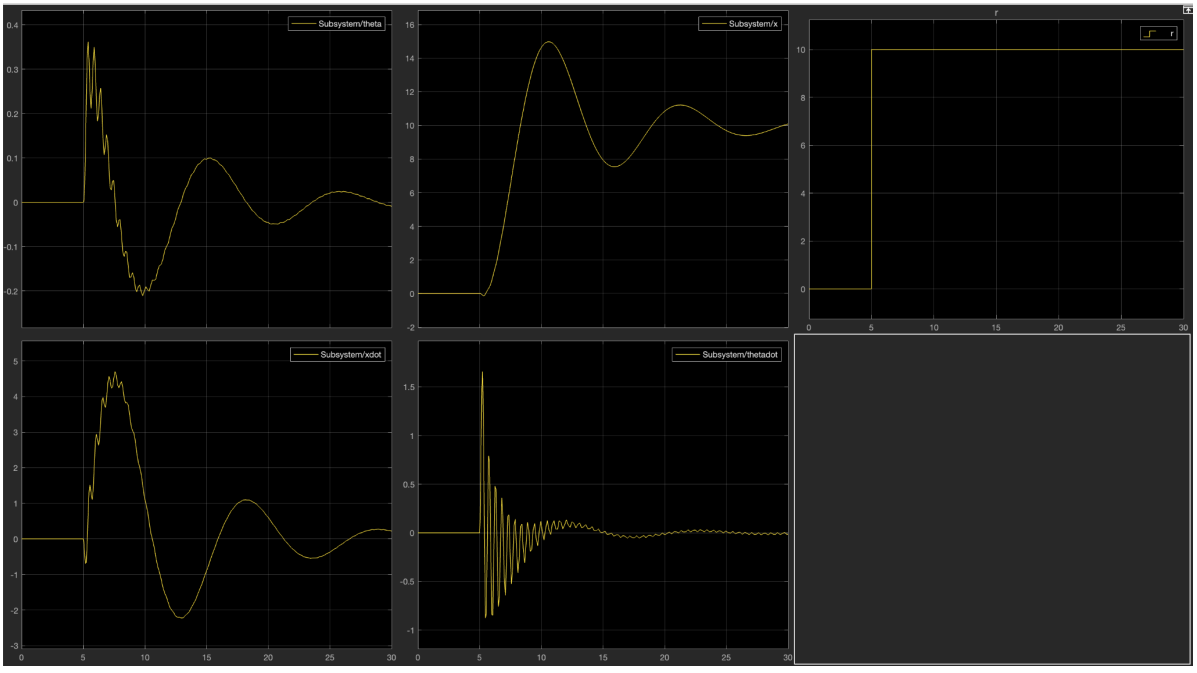


ب) در این قسمت به سیستم plant یک کنترلر نیز اضافه کردیم که از چهار خروجی plant به علاوه انتگرال مقدار error متغیر جابجایی (با احتساب چند ضریب) برای بهبود سیستم و در نتیجه رسیدن به هدف سیستم که ثابت نگه داشتن پاندول است، استفاده میکنیم. در ابتدا شکلی از مدل شبیه سازی شده بهبود یافته را میبینیم:



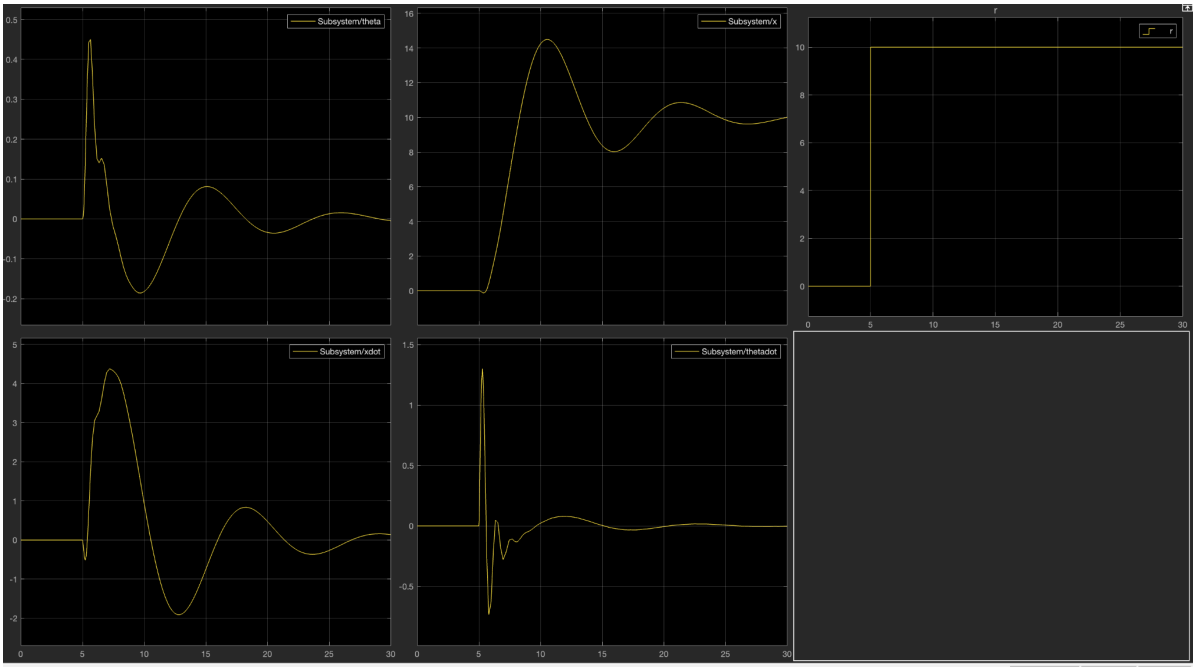
در ابتدای کار (و با توجه به راهنمایی‌های صورت مسئله) مقادیر تصادفی برای هر یک از Kها گذاشتیم. خروجی چیزی شبیه به عکس زیر شد که همانطور که مشاهده میکنیم از شکل مطلوب نمودار x خیلی دور است و همچنین تغییرات زاویه‌ای هم مطلوب ما نیست.

K_1 = -180, K_2 = -35, K_3 = -20, K_4 = -60, K_I=-20



در نتیجه مقادیر را به این شکل تغییر می‌دهیم:

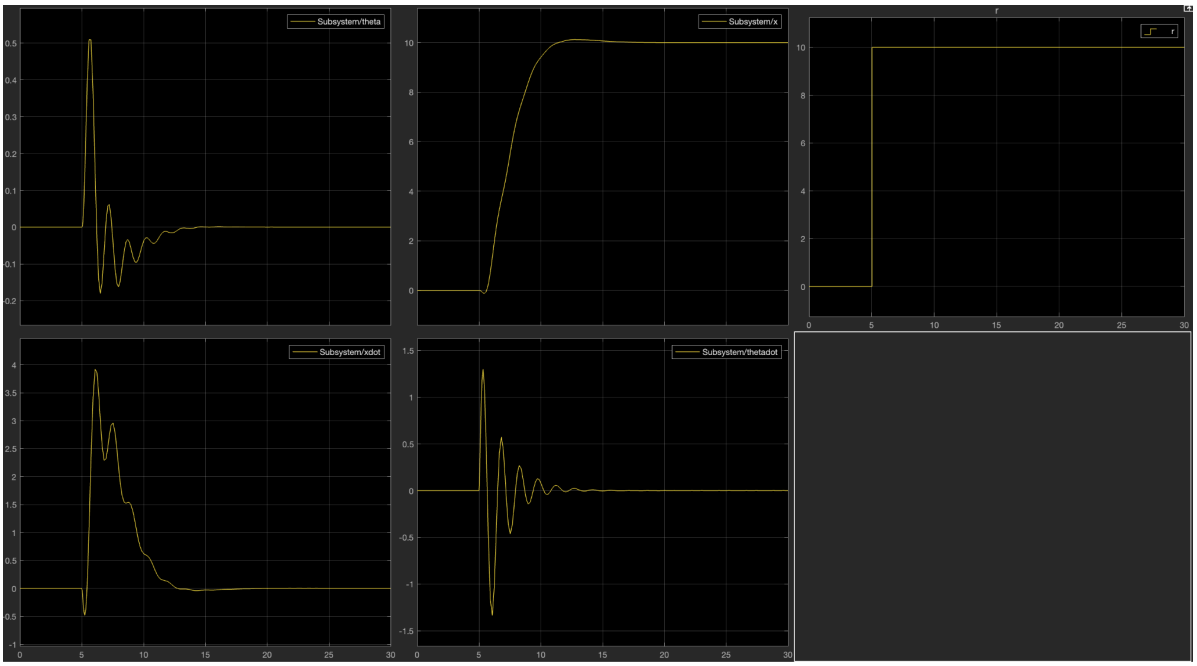
K_1 = -120, K_2 = -50, K_3 = -20, K_4 = -60, K_I=-20



همانطور که مشاهده می‌شود تغییرات زاویه‌ای پایدارتر و نرم‌تر از حالت قبل انجام شده است. اما هنوز به شکل تغییر پله‌ای جابجایی نزدیک نشده‌ایم. در نتیجه در باقی Kها باید تغییراتی ایجاد کرد.

اکنون مقادیر را به این شکل تغییر می‌دهیم:

K_1 = -120, K_2 = -50, K_3 = -50, K_4 = -50, K_I=-20

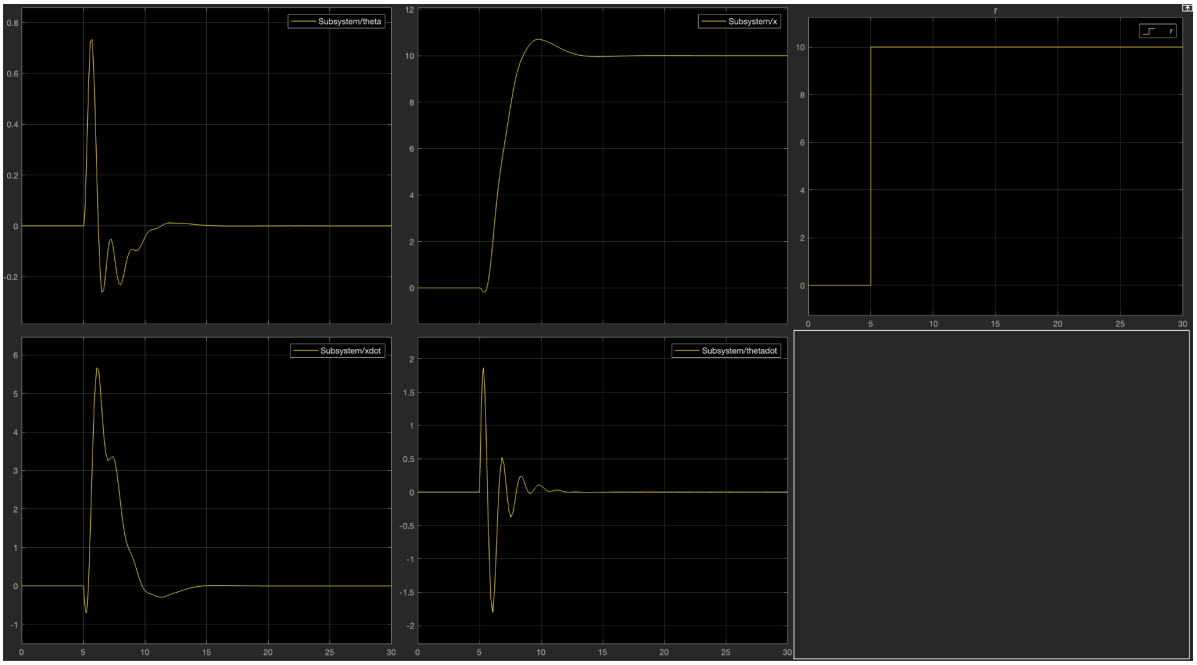


همانطور که مشاهده می‌شود نوسانات تغییرات x هم بهبود یافته و به تبع آن زاویه آونگ نیز در زمان کوتاه‌تری به همگرایی درست یعنی مقدار صفر رسیده است.

حالا مقدار Ki را هم تغییر می‌دهیم و نتیجه به این شکل است:

K_1 = -120, K_2 = -50, K_3 = -50, K_4 = -50, K_I=-30

در اینجا همگرایی جابجایی کمی زودتر انجام شده اما دامنه تغییرات زاویه بیشتر شده. اینجا انتخاب اعمال تغییرات بسته به این است که هدف ما از عملکرد سیستم چیست.



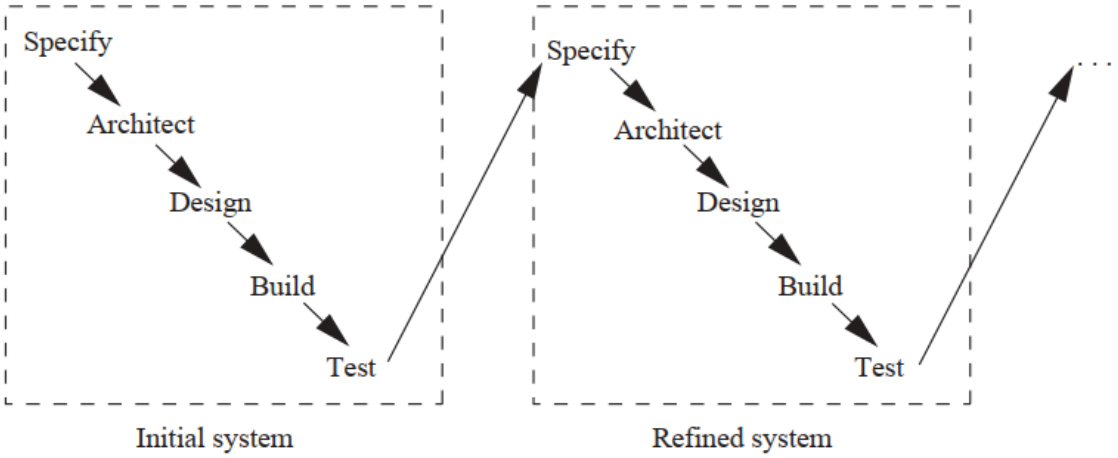
در آخر به تحلیل نمودار فوق می‌پردازیم:

همانطور که میبینیم بعد از ثانیه 5 سیستم شروع به بهبود خود حول هدفی که داریم می‌کند. مقدار ورودی که یک تابع پله است به این معناست که مکان نهایی و دلخواه ما برای قرار گرفتن گاری کجا باشد. با توجه به این موضوعات همانطور که در تصاویر مشخص است گاری به سمت xهای مثبت حرکت می‌کند تا اینکه به حوالی ۱۰ که مکان مطلوب ماست می‌رسد و در نهایت بعد از کمی نوسان در همان‌جا ثابت می‌ماند. طبیعتا مقدار سرعت گاری نیز ابتدا به طور ناگهانی افزایش پیدا می‌کند و سپس بعد از اینکه گاری به محل مورد نظر نزدیک می‌شود به صفر میل میکنند و در نهایت صفر می‌شوند (نمودار x و مشتق x). در مورد پاندول نیز می‌توانیم بگوییم که وقتی گاری در یک لحظه ابتدای نمودار به سمت xهای منفی رفته در نمودار θ مشاهده می‌شود که ابتدا زاویه مثبت می‌شود و به عبارتی نمودار رشد میکند. پس از اینکه نمودار x به سمت مثبت شدن می‌رود θ نیز به سمت منفی شدن می‌رود و همانطور که دیده می‌شود

با هر حرکت مثبت x، θ با کمی تاخیر به سمت منفی نمودار حرکت می‌کند (و برعکس) تا جایی که با نوسانات خیلی کم به سمت صفر شدن همگرا می‌شود.

سوال 1)

الف) روش successive refinement یک متدولوژی توسعه نرم افزار است که در آن نرم افزار در چندین نسخه متوالی ساخته و به تدریج بهبود داده می‌شود. این روش معمولاً زمانی استفاده می‌شود که تیم توسعه آشنایی کمی با دامنه کاربردی سیستم (application domain) دارند و نیاز به آزمایش تدریجی معماری و طراحی دارند.



هر iteration شامل مراحل اجرایی زیر است:

- مشخص کردن نیازمندی (Specify): در این مرحله نیازمندی‌های سیستم شناسایی، معرفی و ثبت می‌شوند. در تکرارهای بعدی این نیازمندی‌ها بهبود و اصلاح می‌یابند.
- معماری (Architecture): ساختار کلی انتخاب شده و تصمیماتی قبیل تکنولوژی‌های اصلی، معماری نرم افزار و ... مشخص می‌شوند.
- طراحی (Design): برنامه ریزی پیاده سازی در این بخش اتفاق می‌افتد. کارهایی نظیر طراحی پایگاه داده و انتخاب الگوریتم‌ها.
- پیاده سازی (Build): سیستم در این قسمت ساخته می‌شود. به عبارتی بسته به iteration نسخه‌ای ساخته یا اصلاح می‌شود.
- آزمون (Test): سیستم مورد آزمون قرار می‌گیرد و اشکالات به iteration بعدی انتقال می‌یابد.

به طور کلی ابتدا یک سیستم اولیه ساخته می‌شود، اشکالات آن یافته می‌شود و با در نظر گیری اصلاحات نسخه جدیدی ساخته می‌شود. این کار آنقدر ادامه می‌یابد تا سیستم مورد نظر ساخته شود.
اگر بخواهیم این متدولوژی را با متدولوژی‌های دیگر توسعه مقایسه کنیم می‌توانیم بگوییم: این روش عیب مدل آبشاری که سختی بازگشت به مراحل قبل است را می‌گیرد چون توسعه مرحله به مرحله دارد. در مقایسه با مدل V که تمرکز آن روی تست بود این مدل انعطاف پذیرتر است همچنان که تست و نگاه به اصلاحات را در خود جای داده است. نسبت به مدل اجایل تکرارها کندتر و ساختاریافته‌تر هستند اما مشابه آن نگاه تدریجی را دارد. به طور خلاصه می‌توان گفت این روش برای پروژه‌هایی خوب است که نیاز به کشف تدریجی راه‌حل داریم.

ب) تیم بازبینی طراحی وظیفه بررسی و ارزیابی یک مؤلفه از سیستم را بر عهده دارد تا مشکلات طراحی، باگ‌ها و ناهماهنگی‌ها را در مراحل اولیه شناسایی و اصلاح کند. این فرآیند باعث کاهش زمان توسعه، بهبود کیفیت پیاده‌سازی و جلوگیری از مشکلات پرهزینه در مراحل بعدی می‌شود.

تیم بازبینی طراحی و وظایف آنها به صورت زیر دسته‌بندی می‌شوند:
۱- طراحان (Designers):
ارائه و توضیح طراحی به سایر اعضای تیم.
پاسخ به سوالات و دریافت بازخورد برای بهبود طراحی.

۲- رهبر بازبینی (Review Leader):
هماهنگی جلسه بازبینی و توزیع اسناد لازم.
مدیریت بحث و پیگیری اصلاحات موردنیاز پس از جلسه.

۳- نویسنده بازبینی (Review Scribe):
ثبت نکات و مشکلات مطرح‌شده در جلسه.
تهیه گزارش جلسه برای اطمینان از اعمال تغییرات لازم.

۴- مخاطبان بازبینی (Review Audience):
بررسی طراحی و ارائه بازخورد.
شناسایی مشکلات احتمالی که ممکن است از دید طراحان پنهان مانده باشد.