



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

پروژه کنترل مدرن

موضوع: سیستم چهار تانک

استاد درس: دکتر شریفی

دانشکده مهندسی برق

اعضای گروه:

فروغ افخمی ۹۸۲۳۰۰۶

ريحانه آهنی ۹۸۲۳۰۰۹

فاطمه رفیعی ۹۸۲۳۰۳۹

سمیرا سلجوqi ۹۸۲۳۰۴۸

پاییز ۱۴۰۱

فهرست مطالب

۴	مقدمه.....
۵	فاز اول پژوهش.....
۵	۱-۱ معرفی سیستم.....
۵	۱-۱-۱ کلیات سیستم و نحوه عملکرد.....
۷	۱-۱-۲ اجزای سیستم و سنسورها.....
۷	۱-۱-۳ کاربرد ها.....
۸	۱-۲ مدل غیرخطی سیستم و پارامتر ها.....
۹	۱-۳ خطی سازی مدل و فضای حالت سیستم.....
۱۴	۱-۴ فرم قطری بلوکی جردن و ماتریس تبدیل سیستم.....
۱۵	۱-۵ توابع تبدیل و نمودار صفر و قطب.....
۱۷	۱-۶ ماتریس انتقال حالت، پاسخ حالت و پاسخ خروجی
۱۹	۱-۷ تحریک نشدن به فرکانس خاص.....
۲۰	۱-۸ محدود معتبر مدل.....
۲۳	۱-۹ کنترل پذیری و رویت پذیری
۲۴	۱-۱۰ تجزیه کالمن.....
۲۵	فاز دوم پژوهش.....
۲۵	۲-۱ زیر سیستم کاهش ناپذیر سیستم خطی شده.....
۲۵	۲-۲ طراحی کنترل فیدبک حالت بر اساس مشخصات مطلوب سیستم.....
۲۷	۲-۳ شبهه سازی و تحلیل سیستم غیر خطی و خطی شده با فیدبک حالت.....
۲۷	۲-۳-۱ شبهه سازی سیستم خطی شده
۲۹	۲-۳-۲ شبهه سازی سیستم غیرخطی
۳۲	۴-۲ ناپایداری سیستم غیر خطی با فیدبک حالت.....
۳۳	۵-۲ شبهه سازی پیش جبرانساز استاتیکی و دینامیکی
۳۳	۵-۲-۱ پیش جبرانساز استاتیکی سیستم خطی شده.....
۳۶	۵-۲-۲ پیش جبرانساز استاتیکی سیستم غیر خطی.....
۳۸	۵-۲-۳ پیش جبرانساز دینامیکی سیستم خطی شده.....

۴۱	۴-۵-۲ پیش جبرانساز دینامیکی سیستم غیر خطی
۴۳	۶-۲ شبیه سازی پیش جبرانساز ردیابی سیگنال مرجع سینوسی
۴۶	فاز سوم پروژه
۴۶	۱-۳ رویتگر مرتبه کامل و مرتبه کاهش یافته
۴۶	۱-۳ رویتگر مرتبه کامل
۴۸	۲-۱-۳ رویتگر مرتبه کاهش یافته
۵۱	۲-۳ باز طراحی قسمت پنجم فاز دوم با رویتگر
۵۱	۱-۲-۳ پیش جبرانساز استاتیکی خطی با رویتگر مرتبه کامل
۵۲	۲-۲-۳ پیش جبرانساز استاتیکی غیر خطی با رویتگر مرتبه کامل
۵۳	۳-۲-۳ پیش جبرانساز دینامیکی خطی با رویتگر مرتبه کامل
۵۴	۴-۲-۳ پیش جبرانساز دینامیکی غیر خطی با رویتگر مرتبه کامل
۵۵	۵-۲-۳ پیش جبرانساز استاتیکی خطی با رویتگر مرتبه کاهش یافته
۵۶	۶-۲-۳ پیش جبرانساز استاتیکی غیر خطی با رویتگر مرتبه کاهش یافته
۵۷	۷-۲-۳ پیش جبرانساز دینامیکی خطی با رویتگر مرتبه کاهش یافته
۵۸	۸-۲-۳ پیش جبرانساز دینامیکی غیر خطی با رویتگر مرتبه کاهش یافته
۵۹	۳-۳ بررسی پایداری
۶۰	۴-۳ طراحی کنترل کننده بهینه
۶۲	۵-۳ پیداوه سازی کنترل کننده بهینه
۶۳	جمع بندی
۶۳	منابع و مراجع

مقدمه

سیستم مورد بررسی ما در این پژوهه سیستم چهار تانک است که مسئله اصلی آن کنترل سطح مایعات و تنظیم جریان بین مخازن است. این سیستم به صورت غیر خطی و با تاخیر زمانی کار میکند. این فرآیند آزمایشگاهی دارای چهار مخزن آب، دو تا در بالا و دو تا در پایین، با دو شیر کنترل و دو پمپ است که در آن هدف کنترل سطح تانکها از طریق پمپها می باشد و دارای دو ورودی و دو خروجی است.

در فاز اول پژوهه به طور کامل تر و با جزئیات بیشتر سیستم را بررسی کرده و در ادامه براساس مقاله انتخاب شده، مدل غیر خطی و پارامتر های آن را به دست می آوریم. سپس با استفاده از ماتریس ژاکوبین معادلات و ماتریس های سیستم خطی شده را محاسبه کرده و ویژگی های مختلف آن نظری توابع تبدیل، ماتریس انتقال حالت، کنترل پذیری و رویت پذیری و تجزیه کالمن را به دست می آوریم.

در فاز دوم در ابتدا یک کنترل فیدبک حالت طراحی کرده و سیستم خطی شده و غیر خطی را با این کنترل کننده شبیه سازی و مقایسه می کنیم. در ادامه نیز پیش جبرانساز های استاتیکی و دینامیکی را در هر دو حالت خطی و غیرخطی اجرا می کنیم و در آخر پیش جبرانسازی برای ردیابی سیگنال مرجع سینوسی به دست می آوریم.

در فاز سوم در ابتدا به طراحی رویت گرهای مرتبه کامل و مرتبه کاهش یافته بر هر دو نوع خطی و غیر خطی پرداخته و در ادامه با استفاده از حالت های تخمین زده شده توسط رویتگر ها، مشابه فاز قبل پیش جبرانساز های استاتیکی و دینامیکی را طراحی نموده و مقایسه می کنیم. همچنین انواع پایداری را در مورد سیستم موردنظر بررسی کرده و در آخر نیز یک کنترل کننده بهینه (LQR) جهت بالا بردن سرعت سیستم طراحی و در سیستم های خطی شده و غیر خطی شبیه سازی و مقایسه می کنیم.

فاز اول پروژه

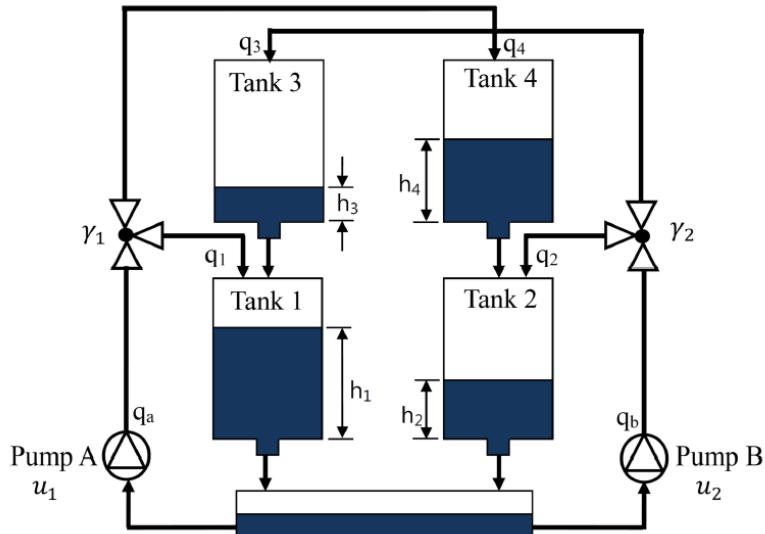
۱-۱ معرفی سیستم

۱-۱-۱ کلیات سیستم و نحوه عملکرد

سیستم چهار تانک (FTS) یک سیستم کنترلی مرسوم است که به صورت غیرخطی عمل می‌کند و دارای تأخیر زمانی نیز می‌باشد. در فرآیندهای صنعتی، کنترل سطح مایع در مخازن و جریان بین آن‌ها یک مسئله‌ی پایه‌ای محسوب می‌شود. در صنایع معمولاً مایعات یا در مخازن ذخیره می‌شوند و یا در حال انتقال به مخازن دیگر هستند. در خیلی از موقع اثواب مایعات در مخازن با مواد شیمیایی هم مخلوط می‌شوند ولی باز هم مسئله‌ی اصلی کنترل سطح مایع و تنظیم جریان بین مخازن است.^[۱]

فرآیند چهارتانک اولین بار توسط Karl Henrik Johansson در سال ۱۹۹۸ پیشنهاد شده است. فرآیند پیشنهادی توسط او یک فرآیند آزمایشگاهی با چهار مخزن آب، دوتا در بالا و دو تا در پایین، با دو شیر کنترل و دو پمپ است که در آن هدف کنترل سطح تانک‌ها از طریق پمپ‌ها می‌باشد. ورودی‌های این فرآیند ولتاژ پمپ‌ها و خروجی آن سطح آب دو تانک پایینی است. مدل خطی شده‌ی این فرآیند دارای صفر قابل تنظیم می‌باشد و امکان بررسی کنترل چندمتغیره را در حالت مینیمم فاز و نامینیمم فاز فراهم می‌کند. مقدار این صفر توسط ضریب شیرهای کنترلی قابل تنظیم است. تنها تغییراتی که از ابتدا تاکنون در مورد این فرآیند اعمال شده این است که به جای دو شیر از چهار شیر در مسیر ورودی هر مخزن استفاده شده است و در یک ساختار دیگر به جای اینکه از پمپ‌ها آب وارد مخزن‌های پایینی شود از هردو مخزن بالایی آب به دو مخزن پایین میریزد که تفاوت در میزان و نحوه تداخل بین مخزن‌ها می‌باشد.

در نمونه‌های قبلی فرآیند چهارتانک، ورودی کنترل کننده ولتاژ پمپ‌ها هستند و فرمان ورودی به پمپ‌ها اعمال شده و کنترل از طریق آن‌ها صورت می‌گیرد. در فرآیند چهارتانک جدید که به صورت وسیع در صنعت کاربرد دارد، کنترل سطح با استفاده از شیر کنترل در مسیر ورودی آب انجام می‌شود. هدف کنترلی در این فرآیند، کنترل سطح هر چهار مخزن و همچنین کنترل فلوی خروجی یکی از مخازن پایینی است.



شکل ۱- شماتیک سیستم چهارタンک

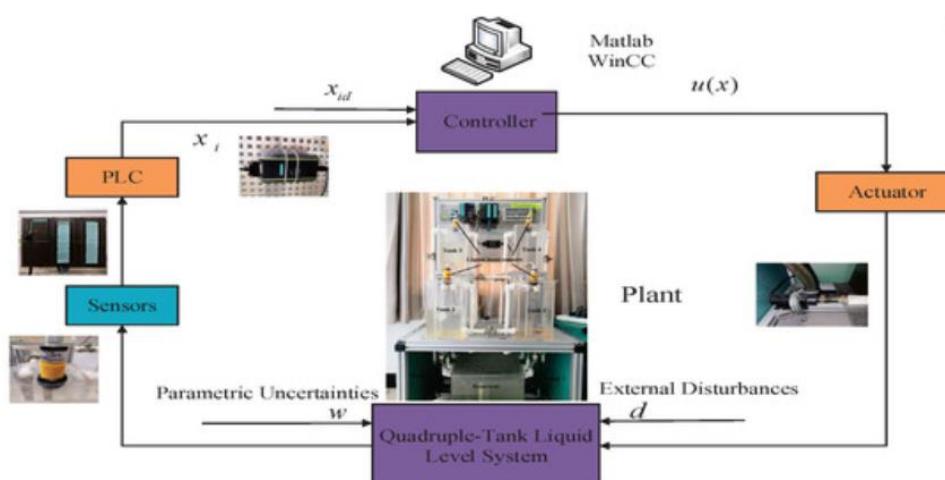
با توجه به اهمیتی که کنترل دبی در کنار کنترل سطح در صنعت به خصوص صنایع نفت و پتروشیمی دارد کنترل فلو به سیستم اضافه میشود. پس برخلاف سیستمهای قبلی که سیستم چندمتغیرهی 2×2 بودند، یک سیستم 5×5 طراحی کردهایم که ورودی‌های کنترلی سیگنال‌های اعمالی به پنج شیر کنترل بوده و خروجی‌ها سطح چهار مخزن و فلوی خروجی از یکی از آن‌ها می‌باشد. فرآیند چهارتانک پیشنهاد شده امکان بررسی کنترلکنندهای چندمتغیره در سیستمهای با ابعاد بزرگ که در کنار پیچیدگی و غیرخطیگری دارای تداخلهای زیادی بین کانالهای مختلف ورودی و خروجی‌شان هستند را فراهم می‌آورد. به این صورت که هر چهار مخزن از طریق لولهایی به هم ارتباط دارند. همچنین نحوه پمپاژ آب به داخل مخزنها از مخزن اصلی به این صورت است که هر پمپ به صورت ضربدری یک مخزن از بالا و یک مخزن از پایین را تغذیه می‌کند. آب مخزن بالایی در مخزن دیگر که مستقیما در زیر آن قرار دارد میریزد. خروجی مخازن پایینی به مخزن اصلی ریخته می‌شود. در مسیر یکی از مخازن پایینی یک شیر کنترل و یک فلومتر قرار دارد که امکان کنترل فلوی خروجی آن را به کاربر می‌دهد. تانک‌های همسطح از طریق لوله‌ایی به هم متصل شده‌اند که باعث انتقال آب بین آنها می‌شود. در تمامی لوله‌های رابط دستگاه از شیرها استفاده شده است که بتوان برای ایجاد شرایط مختلف آزمایشگاهی لوله مربوطه را حذف کرده و یا مقدار مقاومت مسیر را تغییر داد.

این سیستم نیز مانند بسیاری از فرآیندهای صنعتی، چند متغیره است، موضوعی که گاها مطرح می‌شود این است که اغلب فرآیندهای شیمیایی غیرخطی هستند و دینامیک پیچیدهای دارند، مانند داشتن چندین ورودی و خروجی، پاسخ معکوس، رفتار چندگانه و همچنین تغییر در پارامترهای خود فرآیند. به همین دلیل استفاده از

کنترلرهای PID که ثابت هستند به اندازه کافی تأثیرگذار نیست، چون ممکن است به خاطر سوئیچینگ مدام، منجر به نوسانات زیاد و اتلاف منابع گردد. در نتیجه در برخی موارد از کنترلر Model Predictive Control (MPC) بهره می‌گیرند که توانایی جبران ضعفهای PID را دارد؛ چرا که قادر است تأخیر زمانی که از ویژگی‌های ذاتی سیستم است، عدم قطعیت در مدل‌سازی و تداخل متغیرها را مدیریت کند.

۱-۱-۲ اجزای سیستم و سنسورها

این دستگاه چهار مخزن اصلی دارد (البته انواع دیگر آن اعم از دو مخزن و سه مخزن هم وجود دارد)، در نتیجه به چهار عدد سنسور سطح (Level transmitters) نیاز داریم. علاوه بر آن دو پمپ در دو طرف دستگاه قرار می‌گیرند که با دو شیر کنترلی (۷۱، ۷۲) میزان جریان را کنترل و تنظیم می‌نمایند. برای این کار باید یک فلومتر هم وجود داشته باشد که این اندازه‌گیری را انجام دهد.



شکل ۲ - روند کلی انجام کار

۱-۳ کاربردها

نیاز مبرم بسیاری از فرایندها و حلقه‌های کنترلی به کنترل چند متغیره باعث افزایش اهمیت این سیستم در صنعت شده است. این موضوع به پیچیده شدن سیستم‌ها و افزایش تعداد ورودی‌ها و خروجی‌های موثر در آن‌ها بر می‌گردد و مطالعات نشان می‌دهد که یکی از جدی‌ترین چالش‌ها در صنایع، فرایندها و سیستم‌های پیشرفته، مسئله تداخل ناشی از ماهیت چند متغیره بودن آن‌ها است. در فرآیندهای شیمیایی مثل نیروگاه‌ها و پالایشگاه‌ها، سیستم‌های هوافضا، صنایع غذایی، و خودروسازی دستگاه‌هایی وجود دارند که در آن‌ها کنترل همزمان چندین ورودی و چندین خروجی ضروری است.

یکی از نمونه کاربردهای آن در هواپیماهast. از نظر ساختاری و عملکردی، سیستم مدیریت سوخت هواپیما یک دستگاه نسبتاً پیچیده است که وظیفه آن اطمینان از تحويل سوخت از مخازن هواپیما به محفظه احتراق موتور

است. این نوع هواپیما است که وظایفی را که سیستم باید در پرواز انجام دهد، یعنی الزامات ویژگی‌های ساختاری عناصر منفرد و پارامترهای آن‌ها را تعیین می‌کند.

معمولًا این نوع هواپیماها دارای ۴-۶ مخزن داخلی و ۳-۲ موتور هستند. در این مورد خاص، هواپیما دارای دو موتور و چهار مخزن داخلی است که در بال‌ها قرار دارند. مخازن به هم متصل هستند تا در صورت بروز هر گونه خرابی، امکان انتقال سوخت از یک قسمت هواپیما به قسمت دیگر را فراهم کنند و در نتیجه تغییر قابل توجهی در مرکز ثقل و از دست دادن احتمالی کنترل هواپیما از بین برود.

هدف از سیستم چهار تانک، کنترل سطح دو تانک پایین با تغییرات دبی ورودی است. این سیستم به منظور اثبات کنترل سیستم چند متغیره با انتقال صفر مطرح شد. QTP به عنوان یک معیار تجربی برای تست تکنیک‌های کنترلی روی پلنت واقعی، هم به صورت متمرکز و هم به صورت سلسله مراتبی و یا کنترلرهای توزیع شده، طراحی شده است. در اینجا، کنترل کننده، با استفاده از مدل خطی سیستم چهار تانک در نقاط کار مختلف طراحی شده است، بنابراین به دلیل ماهیت غیر خطی سیستم چهار تانک، نمی‌تواند عملکرد خوب سیستم کنترل را روی تمام محدوده کاری تضمین نمایند. این فرآیند برای نشان دادن مفهوم کنترل چند متغیره و همچنین محدودیت‌های عملکرد به دلیل صفحه‌ای نیم صفحه راست چند متغیره ایده آل است.^[۲]

۲-۱ مدل غیرخطی سیستم و پارامترها

معادلات اصلی سیستم به صورت زیر هستند:

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1} \cdot \sqrt{2gh_1} + \frac{a_3}{A_1} \cdot \sqrt{2gh_3} + \frac{\gamma_1}{A_1} \cdot k_1 \vartheta_1$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3} \cdot \sqrt{2gh_3} + \frac{a_1}{A_3} \cdot \sqrt{2gh_1} + \frac{\gamma_3}{A_3} \cdot k_3 \vartheta_3$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2} \cdot \sqrt{2gh_2} + \frac{1-\gamma_2}{A_2} \cdot k_2 \vartheta_2$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4} \cdot \sqrt{2gh_4} + \frac{1-\gamma_1}{A_4} \cdot k_1 \vartheta_1$$

در مواردی نیز می‌توان به صورت گسترده‌تر عمل کرد، به این شکل که دینامیک بین پمپ و سیگنال‌های کنترلی u و سرعت v هم در مدلسازی اعمال گردد. که این کار با تأخیر مرتبه یک، گین واحد و ثابت زمانی مدل می‌شود.

$$\frac{d\vartheta_1}{dt} = -\frac{\vartheta_1}{\tau_1} + \frac{1}{\tau_1} u_1$$

$$\frac{d\vartheta_2}{dt} = -\frac{\vartheta_2}{\tau_2} + \frac{1}{\tau_2} u_2$$

جدول ۱ - پارامترها و مقادیر عددی

نام	پارامتر	مقدار عددی
h	ارتفاع مایع در مخزن‌ها	۱۱,۴, ۱۱,۶, ۵,۳, ۴,۰ cm
ϑ	ورودی‌ها (پمپ‌ها)	۵۰%, ۵۰%
a_i	سطح شعاع داخلی لوله‌ی مخزن‌ها	۲,۱۰, ۲,۱۴, ۲,۲, ۲,۳ cm ^۲
A_i	سطح مقطع مخزن	۷۳۰ cm ^۲
γ_1	ضریب فلوی مخزن ۱	۰,۳
γ_2	ضریب فلوی مخزن ۲	۰,۳۵
k_j	گین پمپ‌ها	۷,۴۵, ۷,۳۰ cm ^۳ /(s %)
g	ثابت گرانش	۹۸۱ cm/s ^۲
τ_i	ثابت زمانی پمپ	۲,۰, ۲,۱ s

۱-۳ خطی سازی مدل و فضای حالت سیستم

مدل غیرخطی سیستم چهارتانک از ابتدا به این صورت حاصل می‌شود:

$$[\text{Rate of accumulation}] = [\text{Rate of in-flow}] - [\text{Rate of out-flow}]$$

در نتیجه مدل غیرخطی هر فرآیند به این شکل است:

$$A_i \frac{dh_i}{dt} = q_{in_i} - q_{out_i}$$

عبارت q_{in_i} تنها به ولتاژ ورودی بستگی دارد و q_{out_i} به شتاب گرانش و آب درون مخزن وابسته است، و با استفاده از رابطه‌ی اصل برنولی به دست می‌آید.

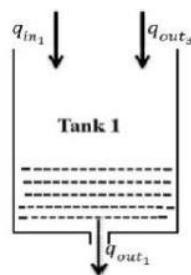
$$q_1 = \gamma_1 q_a , \quad q_2 = \gamma_2 q_a$$

$$q_3 = (1 - \gamma_2) q_b , \quad q_4 = (1 - \gamma_1) q_a$$

می‌دانیم γ_1 و γ_2 ضرایب مربوط به شیرها هستند که فلو توسط آن‌ها کنترل می‌شود. و همانطور که اشاره شد مقدار q خروجی به h و g بستگی دارد، در نتیجه داریم:

$$q_{out_i} = a_i \sqrt{2gh_i}$$

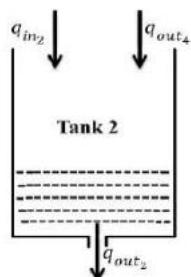
حال اگر بخواهیم این رابطه را برای هر مخزن به طول جداگانه بررسی کنیم:



شکل ۳- تعادل در مخزن ۱

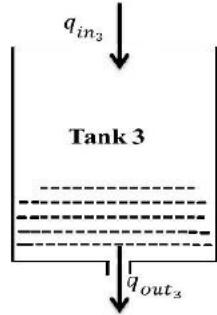
همان رابطه را تنها برای مخزن شماره یک بازنویسی می‌کنیم.

$$A_1 \frac{dh_1}{dt} = q_{in_1} + q_{out_2} - q_{out_1} = \gamma_1 q_a - a_1 \sqrt{2gh_1} + a_2 \sqrt{2gh_2}$$



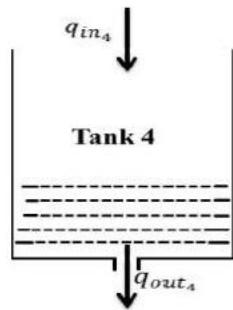
شکل ۴- تعادل در مخزن ۲

$$A_r \frac{dh_r}{dt} = q_{in_r} + q_{out_f} - q_{out_r} = \gamma_r q_b - a_r \sqrt{2gh_r} + a_f \sqrt{2gh_f}$$



شکل ۵ - تعادل در مخزن ۳

$$A_r \frac{dh_r}{dt} = q_{in_r} - q_{out_r} = (1 - \gamma_r)q_b - a_r \sqrt{2gh_r}$$



شکل ۶ - تعادل در مخزن ۴

$$A_f \frac{dh_f}{dt} = q_{in_f} - q_{out_f} = (1 - \gamma_1)q_a - a_f \sqrt{2gh_f}$$

با تقسیم کردن طرفین به A ، به همان روابط می‌رسیم؛ که در آن، $k_1\vartheta_1$ و q_a همان $k_2\vartheta_2$ و همان q_b است.

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1} \cdot \sqrt{2gh_1} + \frac{a_r}{A_1} \cdot \sqrt{2gh_r} + \frac{\gamma_1}{A_1} \cdot k_1\vartheta_1$$

$$\frac{dh_r}{dt} = -\frac{a_r}{A_r} \cdot \sqrt{2gh_r} + \frac{a_f}{A_r} \cdot \sqrt{2gh_f} + \frac{\gamma_r}{A_r} \cdot k_r\vartheta_r$$

$$\frac{dh_r}{dt} = -\frac{a_r}{A_r} \cdot \sqrt{2gh_r} + \frac{1-\gamma_r}{A_r} \cdot k_r \vartheta_r$$

$$\frac{dh_f}{dt} = -\frac{a_f}{A_f} \cdot \sqrt{2gh_f} + \frac{1-\gamma_f}{A_f} \cdot k_f \vartheta_f$$

این چهار معادله به دست آمده روابط اصلی سیستم ما هستند که بر اساس آنها معادلات فضای حالت را به دست می‌آوریم.

برای این کار ابتدا لازم است خطی سازی را با استفاده از بسط تیلور و سپس ماتریس ژاکوبین انجام دهیم. می‌دانیم:

$$\frac{dx_n}{dt} = f_1(h_1, h_r, \dots, h_n, u_1, u_r, \dots, u_n)$$

$$H_e = h_e + \Delta h$$

$$U_e = u_e + \Delta u$$

Taylor series, ignoring higher orders:

$$\begin{aligned}\dot{x} &= \frac{dx}{dt} = f(H_e, U_e) = f(h_e + \Delta h, u_e + \Delta u) \\ &= f(h_e, u_e) + \frac{df}{dh}(h_e, u_e)\Delta h + \frac{df}{du}(h_e, u_e)\Delta u\end{aligned}$$

فرم ماتریس ژاکوبین به صورت کلی به شکل زیر است.

$$A = \frac{\delta f}{\delta h}(h_e, u_e) = \begin{bmatrix} \frac{\delta f_1}{\delta h_1} & \dots & \frac{\delta f_1}{\delta h_n} \\ \vdots & \dots & \vdots \\ \frac{\delta f_n}{\delta h_1} & \dots & \frac{\delta f_n}{\delta h_n} \end{bmatrix} \quad B = \frac{\delta f}{\delta u}(h_e, u_e) = \begin{bmatrix} \frac{\delta f_1}{\delta u_1} & \dots & \frac{\delta f_1}{\delta u_n} \\ \vdots & \dots & \vdots \\ \frac{\delta f_n}{\delta u_1} & \dots & \frac{\delta f_n}{\delta u_m} \end{bmatrix}$$

از روابط قبلی می‌دانیم:

$$\frac{dH_e}{dt} = \frac{d}{dt}h_e + \frac{d}{dt}\Delta h$$

Since $\frac{d}{dt}\Delta h$ is constant, $\frac{dH_e}{dt} = \frac{d}{dt}h_e$

As a result, $f(h_e, u_e) = \cdot$ and $\frac{dx}{dt} = \frac{df}{dh}(h_e, u_e)\Delta h + \frac{df}{du}(h_e, u_e)\Delta u$

$$A = \frac{\partial f}{\partial h}(h_e, u_e)$$

$$B = \frac{\partial f}{\partial u}(h_e, u_e)$$

$$\frac{d}{dt}\Delta x = A\Delta x + B\Delta u$$

همین کار را عیناً برای ماتریس‌های C و D هم تکرار می‌کنیم و به $y = C\Delta x + D\Delta u$ رسیم.

ماتریس اصلی خطی‌سازی:

$$A = \begin{bmatrix} \frac{\partial \dot{h}_1}{\partial h_1} \Big|_{h_1=h_1^0} & \frac{\partial \dot{h}_1}{\partial h_3} \Big|_{h_3=h_3^0} & \frac{\partial \dot{h}_1}{\partial v_2} \Big|_{v_2=v_2^0} & \frac{\partial \dot{h}_1}{\partial v_1} \Big|_{v_1=v_1^0} & \frac{\partial \dot{h}_1}{\partial h_2} \Big|_{h_2=h_2^0} & \frac{\partial \dot{h}_1}{\partial h_4} \Big|_{h_4=h_4^0} \\ \frac{\partial \dot{h}_3}{\partial h_1} \Big|_{h_1=h_1^0} & \frac{\partial \dot{h}_3}{\partial h_3} \Big|_{h_3=h_3^0} & \frac{\partial \dot{h}_3}{\partial v_2} \Big|_{v_2=v_2^0} & \frac{\partial \dot{h}_3}{\partial v_1} \Big|_{v_1=v_1^0} & \frac{\partial \dot{h}_3}{\partial h_2} \Big|_{h_2=h_2^0} & \frac{\partial \dot{h}_3}{\partial h_4} \Big|_{h_4=h_4^0} \\ \frac{\partial \dot{v}_2}{\partial h_1} \Big|_{h_1=h_1^0} & \frac{\partial \dot{v}_2}{\partial h_3} \Big|_{h_3=h_3^0} & \frac{\partial \dot{v}_2}{\partial v_2} \Big|_{v_2=v_2^0} & \frac{\partial \dot{v}_2}{\partial v_1} \Big|_{v_1=v_1^0} & \frac{\partial \dot{v}_2}{\partial h_2} \Big|_{h_2=h_2^0} & \frac{\partial \dot{v}_2}{\partial h_4} \Big|_{h_4=h_4^0} \\ \frac{\partial \dot{v}_1}{\partial h_1} \Big|_{h_1=h_1^0} & \frac{\partial \dot{v}_1}{\partial h_3} \Big|_{h_3=h_3^0} & \frac{\partial \dot{v}_1}{\partial v_2} \Big|_{v_2=v_2^0} & \frac{\partial \dot{v}_1}{\partial v_1} \Big|_{v_1=v_1^0} & \frac{\partial \dot{v}_1}{\partial h_2} \Big|_{h_2=h_2^0} & \frac{\partial \dot{v}_1}{\partial h_4} \Big|_{h_4=h_4^0} \\ \frac{\partial \dot{h}_2}{\partial h_1} \Big|_{h_1=h_1^0} & \frac{\partial \dot{h}_2}{\partial h_3} \Big|_{h_3=h_3^0} & \frac{\partial \dot{h}_2}{\partial v_2} \Big|_{v_2=v_2^0} & \frac{\partial \dot{h}_2}{\partial v_1} \Big|_{v_1=v_1^0} & \frac{\partial \dot{h}_2}{\partial h_2} \Big|_{h_2=h_2^0} & \frac{\partial \dot{h}_2}{\partial h_4} \Big|_{h_4=h_4^0} \\ \frac{\partial \dot{h}_4}{\partial h_1} \Big|_{h_1=h_1^0} & \frac{\partial \dot{h}_4}{\partial h_3} \Big|_{h_3=h_3^0} & \frac{\partial \dot{h}_4}{\partial v_2} \Big|_{v_2=v_2^0} & \frac{\partial \dot{h}_4}{\partial v_1} \Big|_{v_1=v_1^0} & \frac{\partial \dot{h}_4}{\partial h_2} \Big|_{h_2=h_2^0} & \frac{\partial \dot{h}_4}{\partial h_4} \Big|_{h_4=h_4^0} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{\partial \dot{h}_1}{\partial u_1} & \frac{\partial \dot{h}_3}{\partial u_1} & \frac{\partial \dot{v}_2}{\partial u_1} & \frac{\partial \dot{v}_1}{\partial u_1} & \frac{\partial \dot{h}_2}{\partial u_1} & \frac{\partial \dot{h}_4}{\partial u_1} \\ \frac{\partial \dot{h}_1}{\partial u_2} & \frac{\partial \dot{h}_3}{\partial u_2} & \frac{\partial \dot{v}_2}{\partial u_2} & \frac{\partial \dot{v}_1}{\partial u_2} & \frac{\partial \dot{h}_2}{\partial u_2} & \frac{\partial \dot{h}_4}{\partial u_2} \end{bmatrix}^T, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

حالا با جایگذاری مقادیر مربوط به این سیستم، ماتریس‌های A, B, C, D را می‌یابیم.

$$A = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} \quad B = \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_1 k_1}{A_1} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix}$$

به همین ترتیب در کد متلب نیز پس از تعریف پارامترها و معادلات غیر خطی سیستم، با دستور jacobian ماتریس‌های خطی‌سازی A و B را مناسب با متغیرهای حالت سیستم تشکیل داده با دستور subs مقادیر

نقطه تعادل را در ماتریس جای گذاری می کنیم و در آخر با دستور ss(A,B,C,D) مدل steady-state را تعریف می کنیم که مقادیر آن به صورت زیر است:

$$A = \begin{bmatrix} -0.01886 & 0.0289922 & 0 & 0.00306 & 0 & 0 \\ 0 & -0.02899 & 0.0065 & 0 & 0 & 0 \\ 0 & 0 & -0.4762 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0 & 0 \\ 0 & 0 & 0.0035 & 0 & -0.019063 & 0.0349 \\ 0 & 0 & 0 & 0.007144 & 0 & -0.0349 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0.5 & 0 & 0 \end{bmatrix}^T, C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

۱-۴ فرم قطری بلوکی جردن و ماتریس تبدیل سیستم

با استفاده از دستور Jordan(A) در متلب ماتریس فرم قطری بلوکی جردن (J)، معکوس ماتریس تبدیل (V) و ماتریس تبدیل (P) به صورت زیر خواهد شد:

V =

$$\begin{bmatrix} -0.1203 & 0 & -2.8641 & 0.4143 & 0 & 1.0000 \\ 1.8984 & 0 & 1.0000 & 0 & 0 & 0 \\ -130.6080 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -65.1066 & 0 & 0 \\ 1.0000 & -2.2044 & 0 & -0.0725 & 1.0000 & 0 \\ 0 & 1.0000 & 0 & 1.0000 & 0 & 0 \end{bmatrix}$$

J =

$$\begin{bmatrix} -0.4762 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0349 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0290 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5000 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0191 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.0189 \end{bmatrix}$$

P =

$$\begin{bmatrix} 0 & 0 & -0.0077 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0154 & 0 & 1.0000 \\ 0 & 1.0000 & 0.0145 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0154 & 0 & 0 \\ 0 & 0 & 0.0077 & 0.0327 & 1.0000 & 2.2044 \\ 1.0000 & 2.8641 & 0.0407 & 0.0064 & 0 & 0 \end{bmatrix}$$

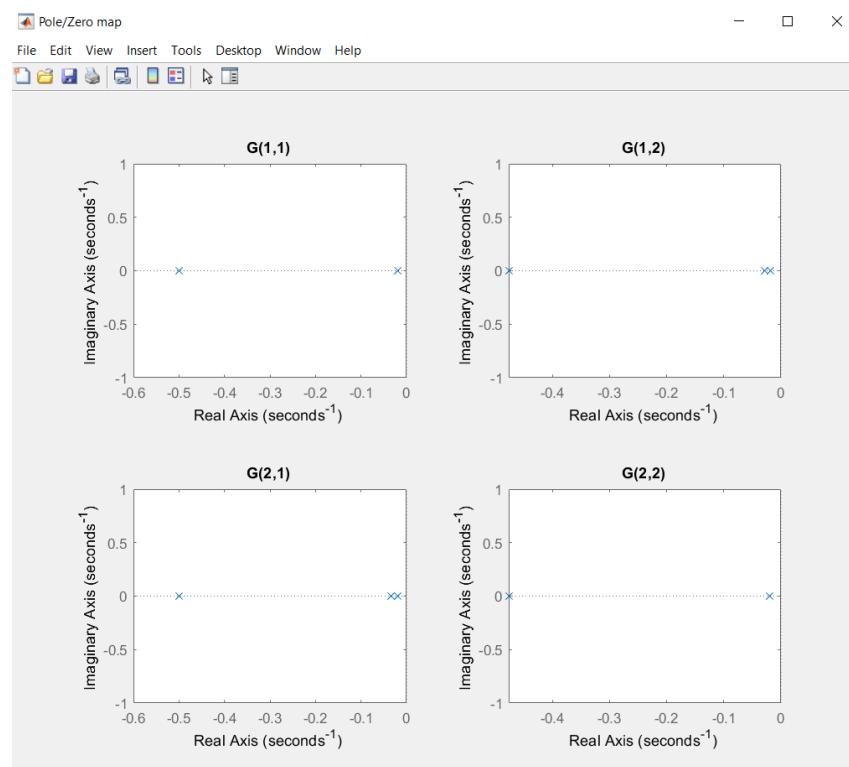
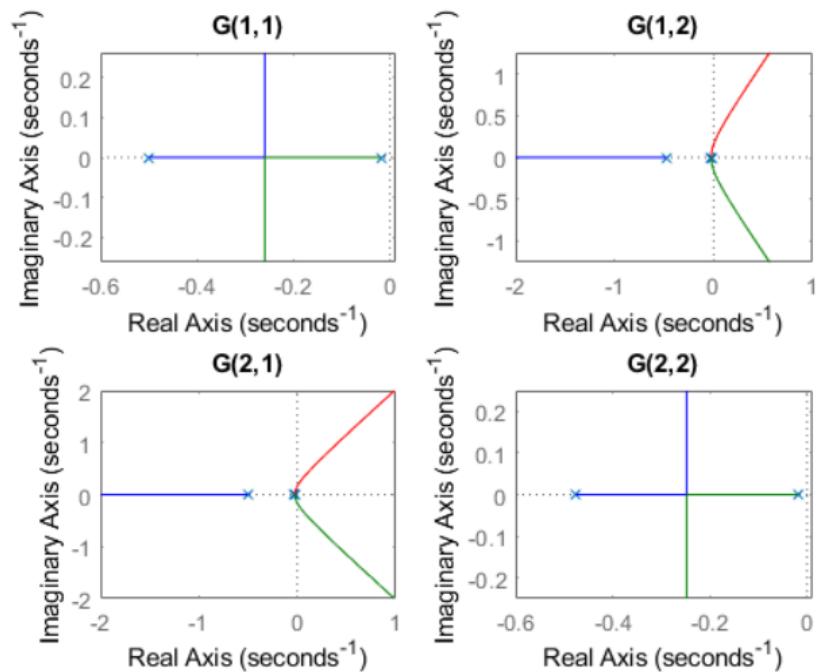
۱-۵ توابع تبدیل و نمودار صفر و قطب

با استفاده از دستور `tf(sys)` می‌توان تابع تبدیل سیستم را پیدا کرد که چون دو ورودی و دو خروجی داریم، ماتریس تابع تبدیل به صورت و مقادیر زیر است:

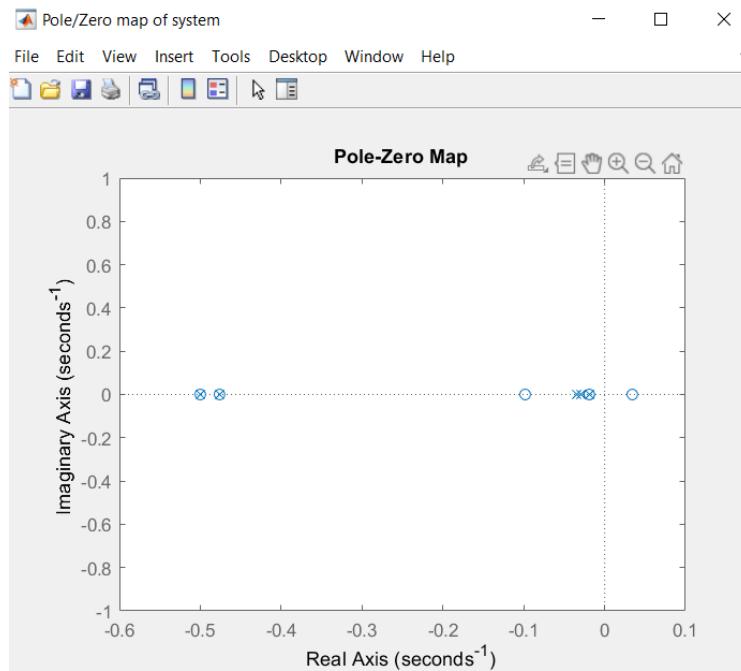
$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

```
G =  
  
From input 1 to output...  
    0.001531  
1:  -----  
    s^2 + 0.5189 s + 0.009435  
  
                           0.0001246  
2:  -----  
    s^3 + 0.554 s^2 + 0.02764 s + 0.0003325  
  
From input 2 to output...  
    8.974e-05  
1:  -----  
    s^3 + 0.5241 s^2 + 0.02334 s + 0.0002605  
  
                           0.001667  
2:  -----  
    s^2 + 0.4953 s + 0.009077  
  
Continuous-time transfer function.
```

حال با استفاده از دستور `rlocus` مکان هندسی ریشه ها و با دستور `pzmap` نمودار صفر و قطب هر تابع تبدیل را مشاهده می کنیم:



تمام صفر و قطب های سیستم نیز به صورت زیر هستند:



۱-۶ ماتریس انتقال حالت، پاسخ حالت و پاسخ خروجی
با استفاده رابطه زیر و دستور expm ماتریس انتقال حالت را به دست می آوریم:

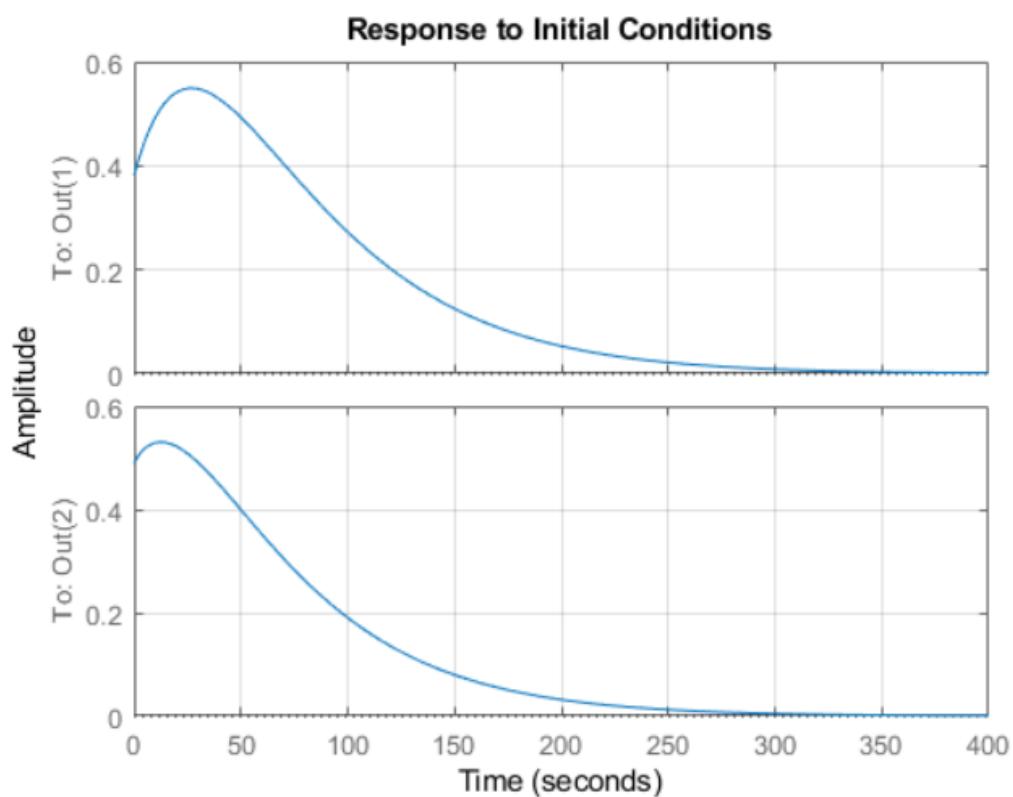
$$\varphi(t) = L^{-1}\{(SI - A)^{-1}\}$$

$$\begin{aligned}
 \text{phi_s} = & \left(\begin{array}{cccccc}
 \frac{1.44e+17}{(1.44e+17s+2.72e+15)} & \frac{1.2e+33}{(1.44e+17s+2.72e+15)(2.88e+17s+8.36e+15)} & \frac{1.64e+32}{(1.44e+17s+2.72e+15)(21.0s+10.0)(2.88e+17s+8.36e+15)} & \frac{8.82e+14}{(2.0s+1.0)(1.44e+17s+2.72e+15)} & 0 & 0 \\
 0 & \frac{2.88e+17}{2.88e+17s+8.36e+15} & \frac{3.93e+16}{(21.0s+10.0)(2.88e+17s+8.36e+15)} & 0 & 0 & 0 \\
 0 & 0 & \frac{21.0}{21.0s+10.0} & 0 & 0 & 0 \\
 0 & 0 & 0 & \frac{2.0}{2.0s+1.0} & 0 & 0 \\
 0 & 0 & \frac{1.06e+16}{(1.44e+17s+2.75e+15)(21.0s+10.0)} & \frac{1.29e+30}{(2.0s+1.0)(1.8e+16s+6.29e+14)(1.44e+17s+2.75e+15)} & \frac{1.44e+17}{1.44e+17s+2.75e+15} & \frac{9.06e+31}{(1.8e+16s+6.29e+14)(1.44e+17s+2.75e+15)} \\
 0 & 0 & 0 & \frac{2.57e+14}{(2.0s+1.0)(1.8e+16s+6.29e+14)} & 0 & \frac{1.8e+16}{1.8e+16s+6.29e+14}
 \end{array} \right) \\
 \text{phi_t} = & \left(\begin{array}{cccccc}
 e^{-0.0189t} & 2.86e^{-0.0189t} - 2.86e^{-0.029t} & 0.0407e^{-0.0189t} - 0.0416e^{-0.029t} + 9.21e^{-4}e^{-0.476t} & 0.00636e^{-0.0189t} - 0.00636e^{-0.5t} & 0 & 0 \\
 0 & e^{-0.029t} & 0.0145e^{-0.029t} - 0.0145e^{-0.476t} & 0 & 0 & 0 \\
 0 & 0 & e^{-0.476t} & 0 & 0 & 0 \\
 0 & 0 & 0 & e^{-0.5t} & 0 & 0 \\
 0 & 0 & 0.00766e^{-0.0191t} - 0.00766e^{-0.476t} & 0.00111e^{-0.5t} - 0.0339e^{-0.0349t} + 0.0327e^{-0.0191t} & e^{-0.0191t} & 2.2e^{-0.0191t} - 2.2e^{-0.0349t} \\
 0 & 0 & 0 & 0.0154e^{-0.0349t} - 0.0154e^{-0.5t} & 0 & e^{-0.0349t}
 \end{array} \right)
 \end{aligned}$$

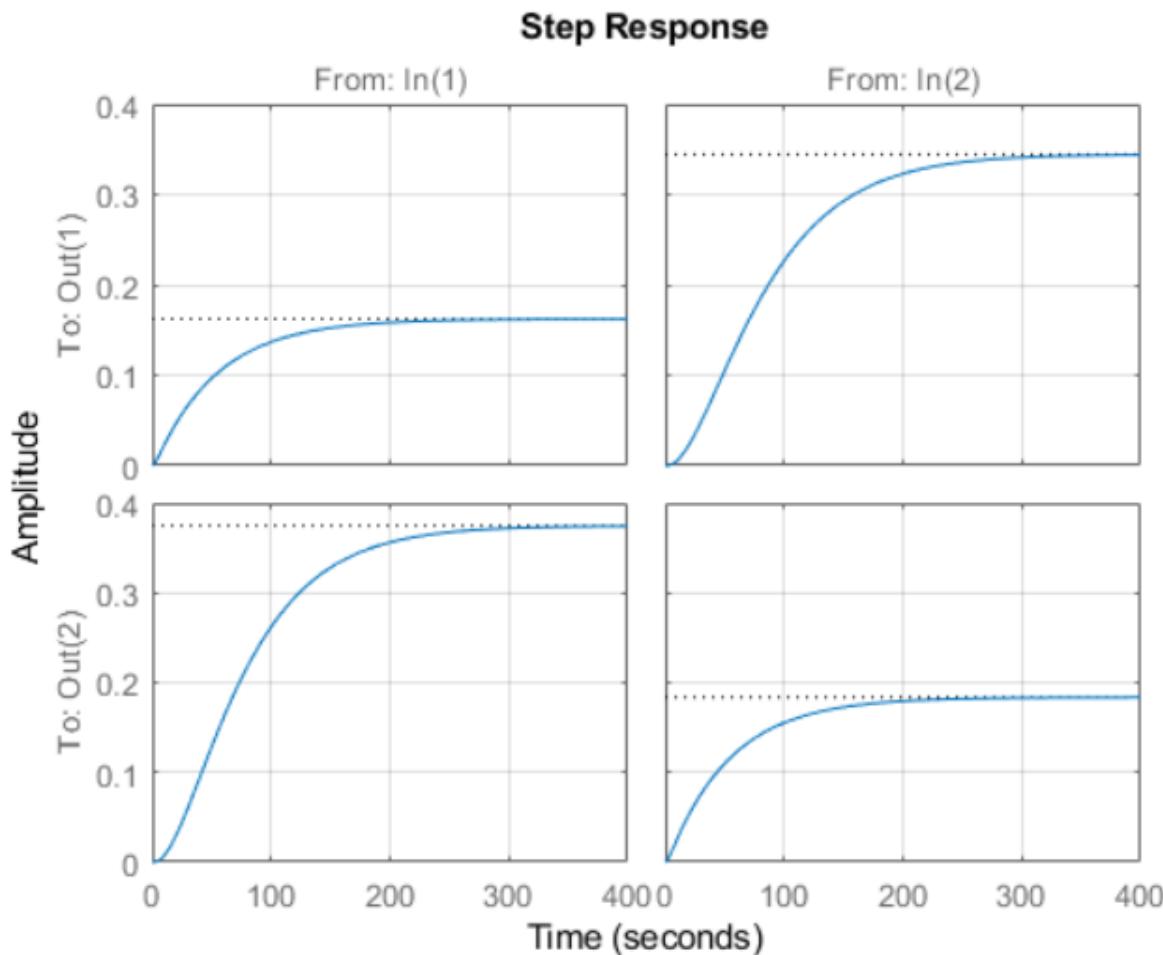
همچنین برای شرایط اولیه دلخواه رندوم پاسخ خروجی سیستم به صورت زیر می شود:

$$x_0 =$$

0.3816
0.7655
0.7952
0.1869
0.4898
0.4456



برای پاسخ پله سیستم نیز با دستور step داریم:



۷-۱ تحریک نشدن به فرکانس خاص

اگر بخواهیم شرایط اولیه را به گونه ای تعیین کنیم که فرکانس خاصی از سیستم تحریک نشود ابتدا باید ماتریس Cc را به صورت زیر تعریف کنیم :

$$Cc = P \times X..$$

که در آن ماتریس P شامل تمام بردار ویژه ها و بردار $X..$ شامل مقادیر اولیه ی حالت ها است.

در این بخش مقدار ویژه های ما این اعداد هستند :

$$\lambda_1 = -0.4762, \lambda_2 = -0.0349, \lambda_3 = -0.029$$

$$\lambda_4 = -0.5, \lambda_5 = -0.0191, \lambda_6 = -0.0189$$

عناصر بردار \mathbf{C}_C به صورت زیر است :

$\mathbf{c}_C =$

$$\begin{aligned} & -0.00766 \cdot x_3 \\ & 0.0154 \cdot x_4 + x_6 \\ & x_2 + 0.0145 \cdot x_3 \\ & -0.0154 \cdot x_4 \\ & 0.00766 \cdot x_3 + 0.0327 \cdot x_4 + x_5 + 2.2 \cdot x_6 \\ & x_1 + 2.86 \cdot x_2 + 0.0407 \cdot x_3 + 0.00636 \cdot x_4 \end{aligned}$$

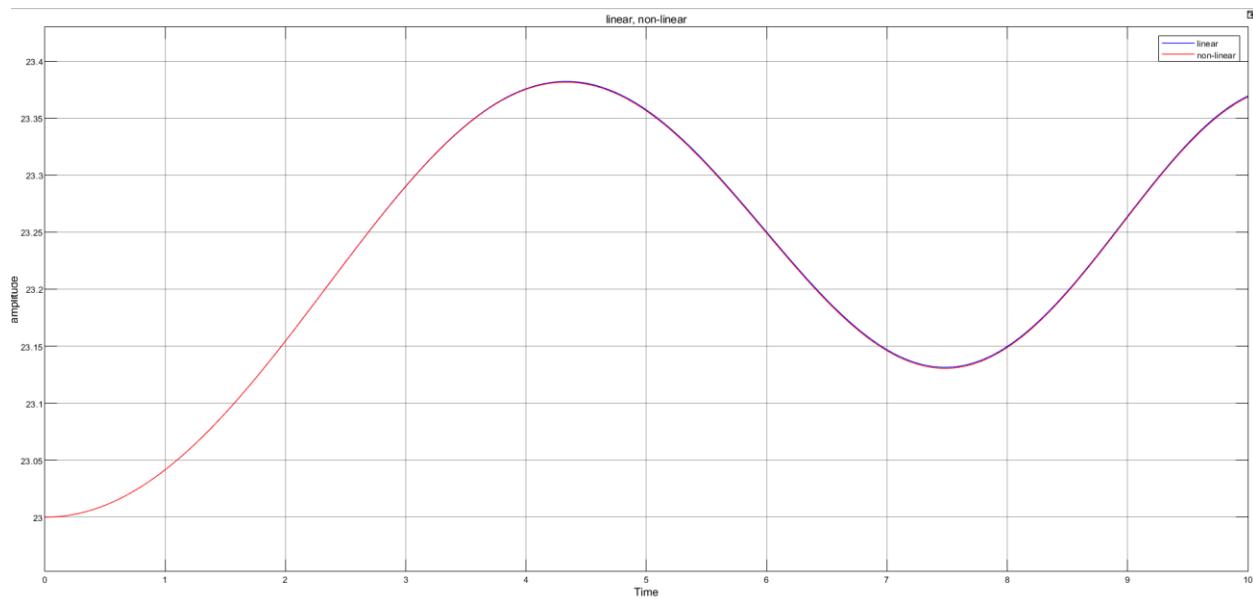
حالا اگر بخواهیم مثلاً مد اول ($\lambda_1 = -0.4762$) توسط ورودی تحریک نشود باید x_3 را برابر با صفر در نظر بگیریم و باقی شرایط اولیه را میتوان به صورت دلخواه انتخاب کرد.

۱-۱ محدود معابر مدل

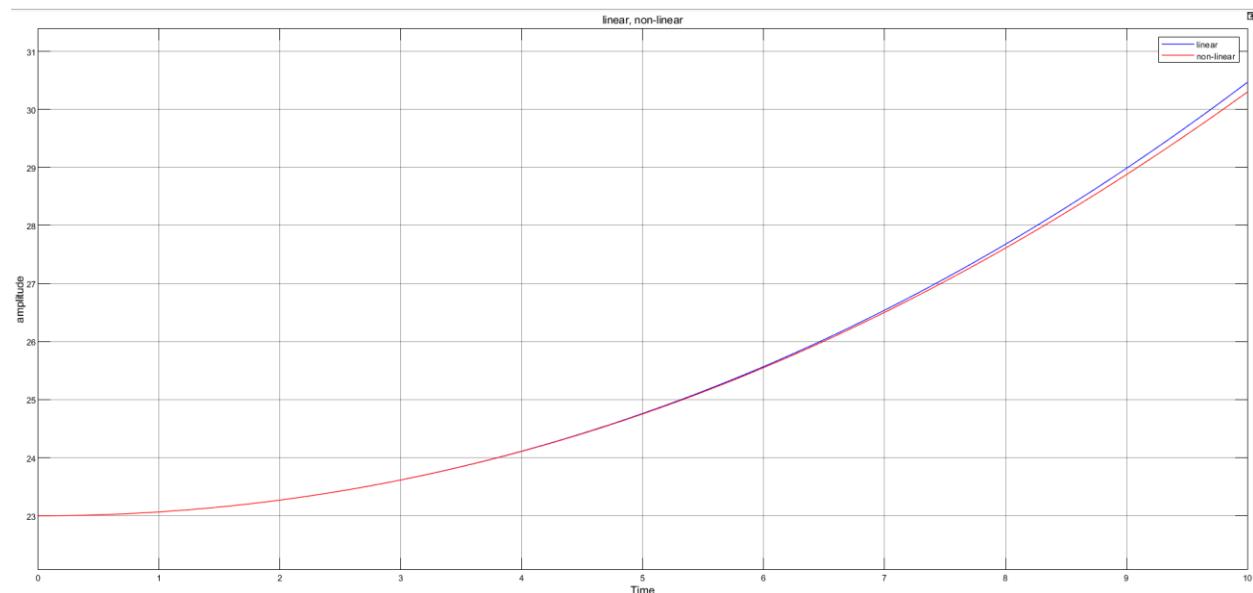
در هنگام بررسی محدوده‌ی معابر خطی سیستم در این قسمت باید ببینیم به ازای چه انحرافی از ورودی، خروجی سیستم خطی سازی شده و غیر خطی از هم فاصله‌ی زیادی می‌گیرند. با توجه به اینکه دو معادله اخر مربوط به سرعت پومپ‌ها و ... است از این دو معادله در این قسمت برای ساده‌تر شدن محاسبات صرف نظر میکنیم. سپس با m معادله موجود دوباره نقاط تعادل را به دست می‌اوریم. در m فایل متلب این کارها را انجام میدهیم و قبل از ران کردن شبیه سازی باید m فایل را ران کنیم تا مقدار دهی‌ها انجام شود.

شایان ذکر است که معادلات غیر خطی سیستم چهارتانک، رادیکالی است و حوالی نقطه‌ی تعادل، تقریب خطی بسیار نزدیک به تابع اصلی غیرخطی است. پس انتظار انحراف زیادی در پاسخ‌ها نداریم. ابتدا ورودی سینوسی را امتحان می‌کنیم با زیاد شدن دامنه ورودی‌ها خروجی‌ها خروجی‌ها خالت خطی و غیرخطی تفاوت خاصی با یکدیگر پیدا نمیکنند که نشان از دقیقت مناسب در محاسبات خطی سازی است.

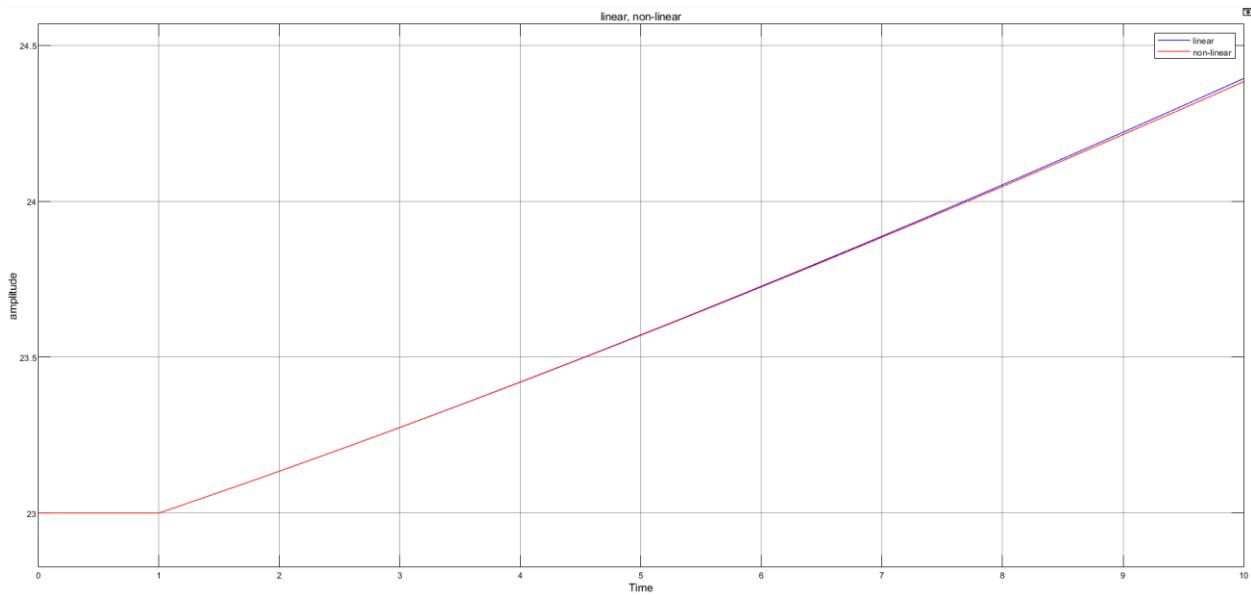
با ورودی signal chirp



با ورودی ramp

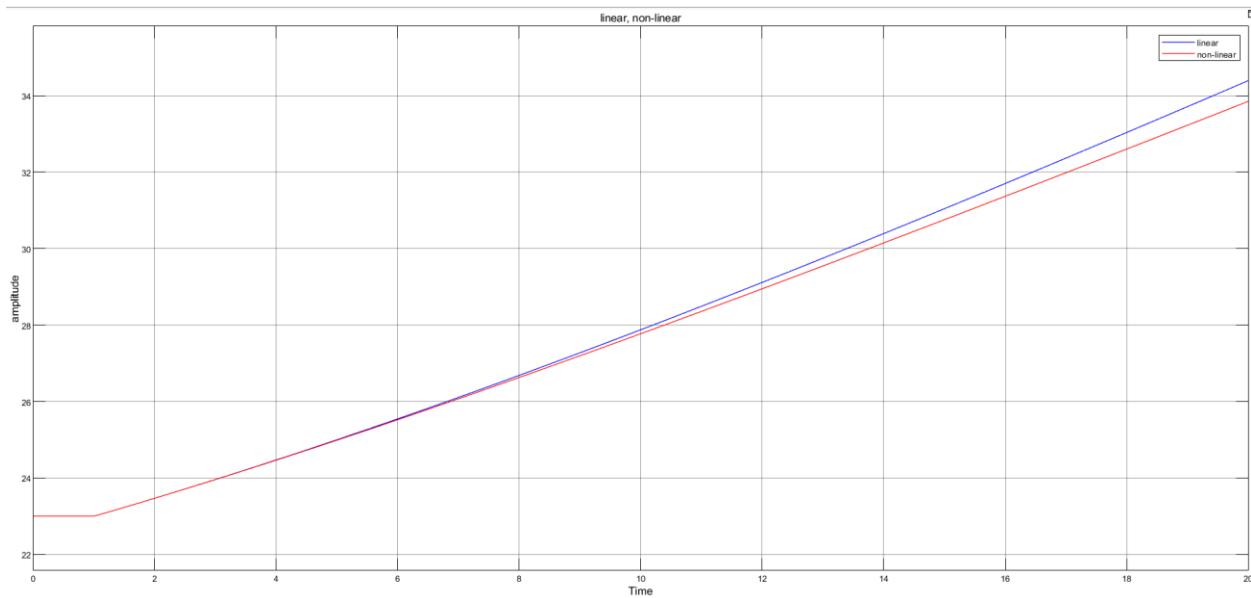


با ورودی پله واحد:



همانطور که میبینیم در ورودی signal chirp و ورودی پله و شیب دو پاسخ به خوبی روی هم افتاده اند.

حال مقدار ورودی پله را زیاد میکنیم که با ورودی پله با اندازه ۳,۵ پاسخ های پله در حالت خطی و غیر خطی از هم فاصله میگیرند. که در زیر مشاهده میکنیم.



۹-۱ کنترل پذیری و رویت پذیری

برای بررسی کنترل پذیری و رویت پذیری به ترتیب رنک ماتریس های $\text{ctrb}(A,B)$ و $\text{obsv}(A,C)$ را محاسبه می کنیم که هر دو دارای رنک کامل هستند. بنابراین تمام مود های سیستم کنترل پذیر و رویت پذیر هستند و سیستم مینیمال است.

ماتریس کنترل پذیری:

$C_o =$

Columns 1 through 7

0	0	0.0015	0	-0.0008	0.0001	0.0004
0	0	0	0.0031	0	-0.0016	0
0	0.4762	0	-0.2268	0	0.1080	0
0.5000	0	-0.2500	0	0.1250	0	-0.0625
0	0	0	0.0017	0.0001	-0.0008	-0.0001
0	0	0.0036	0	-0.0019	0	0.0010

Columns 8 through 12

-0.0000	-0.0002	0.0000	0.0001	-0.0000	
0.0007	0	-0.0004	0	0.0002	
-0.0514	0	0.0245	0	-0.0117	
0	0.0313	0	-0.0156	0	
0.0004	0.0000	-0.0002	-0.0000	0.0001	
0	-0.0005	0	0.0002	0	

$R_c =$

6

ماتریس رویت پذیری:

$O_b =$

1.0000	0	0	0	0	0
0	0	0	0	1.0000	0
-0.0189	0.0290	0	0.0031	0	0
0	0	0.0035	0	-0.0191	0.0349
0.0004	-0.0014	0.0002	-0.0016	0	0
0	0	-0.0017	0.0002	0.0004	-0.0019
-0.0000	0.0001	-0.0001	0.0008	0	0
0	0	0.0008	-0.0001	-0.0000	0.0001
0.0000	-0.0000	0.0000	-0.0004	0	0
0	0	-0.0004	0.0001	0.0000	-0.0000
-0.0000	0.0000	-0.0000	0.0002	0	0
0	0	0.0002	-0.0000	-0.0000	0.0000

$R_o =$

6

۱۰-۱ تجزیه کالمن

با استفاده از دستور mineral(sys) تجزیه کالمن را به صورت زیر به دست می آوریم:

```
A_minreal =  
  
-0.0189  0.0290      0  0.0031      0      0  
  0  -0.0290  0.0065      0      0      0  
  0      0  -0.4762      0      0      0  
  0      0      0  -0.5000      0      0  
  0      0  0.0035      0  -0.0191  0.0349  
  0      0      0  0.0071      0  -0.0349  
  
  
B_minreal =  
  
  0      0  
  0      0  
  0  0.4762  
0.5000      0  
  0      0  
  0      0  
  
  
C_minreal =  
  
  1      0      0      0      0      0  
  0      0      0      0      1      0
```

فاز دوم پروژه

۱-۲ زیر سیستم کاهش ناپذیر سیستم خطی شده

سیستم کاهش ناپذیر دارای تحقق مینیمال است و سیستم مورد نظر ما یعنی سیستم چهار تانک مطابق آنچه در فاز قبل بررسی شد، هیچ مود رویت ناپذیر یا کنترل ناپذیر ندارد. بنابراین ماتریس های تحقق زیر سیستم کاهش ناپذیر، همان ماتریس های خطی شده سیستم اصلی می باشند که به صورت زیر هستند:

$$A = \begin{bmatrix} -0.01886 & 0.0289922 & 0 & 0.00306 & 0 & 0 \\ 0 & -0.02899 & 0.0065 & 0 & 0 & 0 \\ 0 & 0 & -0.4762 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0 & 0 \\ 0 & 0 & 0.0035 & 0 & -0.019063 & 0.0349 \\ 0 & 0 & 0 & 0.007144 & 0 & -0.0349 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.4762 & 0 & 0 & 0 \end{bmatrix}^T, C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

۲-۲ طراحی کنترل کننده فیدبک حالت بر اساس مشخصات مطلوب سیستم

همانطور که در بخش های قبل نیز اشاره شد، این سیستم دارای دو ورودی و دو خروجی و در نتیجه چهار تابع تبدیل به صورت زیر است.

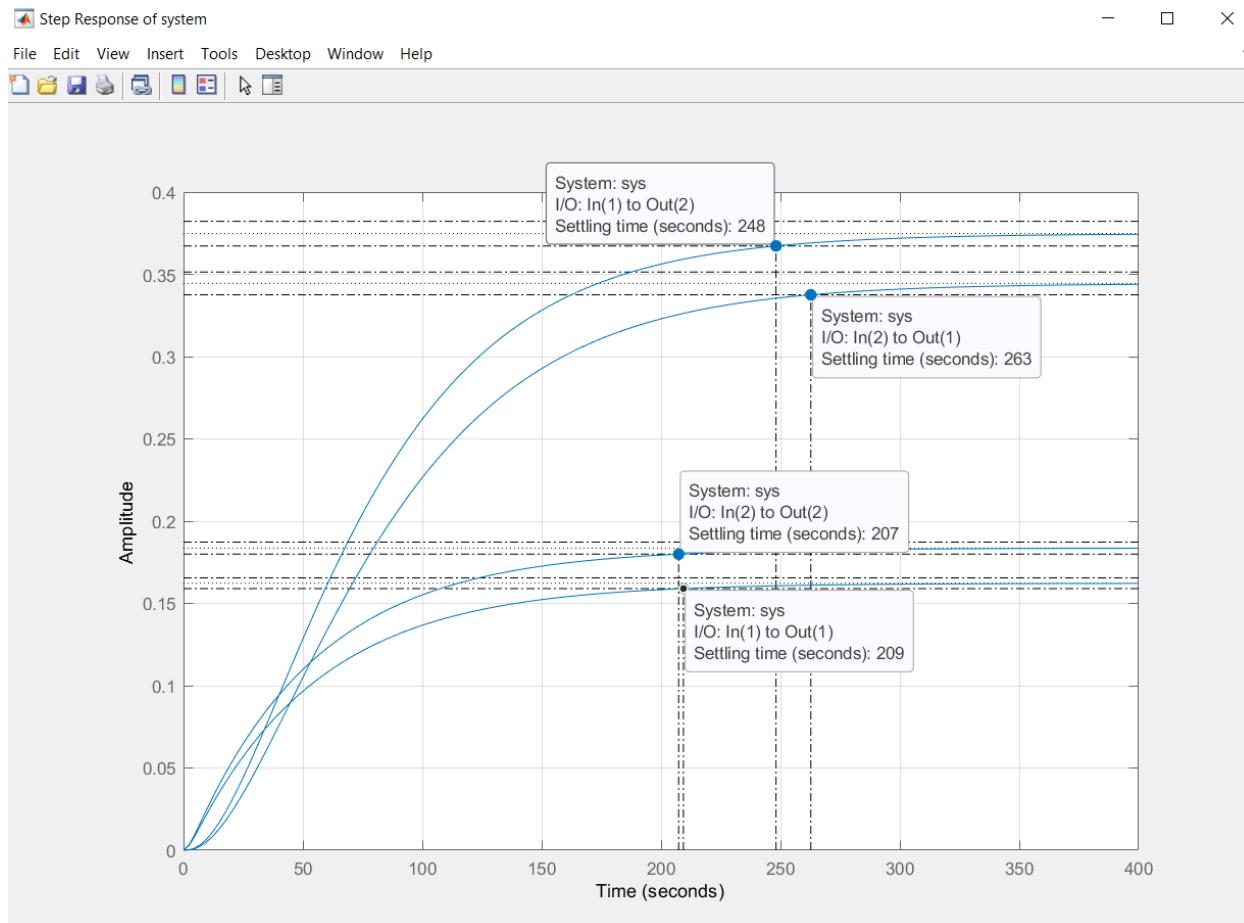
$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

```

G =
From input 2 to output...
8.974e-05
1: -----
s^3 + 0.5241 s^2 + 0.02334 s + 0.0002605
From input 1 to output...
0.001531
1: -----
s^2 + 0.5189 s + 0.009435
0.0001246
2: -----
s^3 + 0.554 s^2 + 0.02764 s + 0.0003325
Continuous-time transfer function.

```

پاسخ پله و مشخصات آن ها به صورت زیر مشخص شده اند که حدودا زمان نشست بین ۲۰۷ تا ۲۶۳ دارد و اورشوت و فراجهشی نیز ندارد.



به همین منظور برای به دست آوردن قطب های مطلوب سیستم معیار زمان نشست را کاهش داده و اورشوت را نیز همان صفر در نظر می گیریم که به این صورت ξ و ω_n به صورت زیر خواهند شد.

$$\left\{ \begin{array}{l} T_{s,new} = \frac{4}{\xi\omega_n} = 6 \\ OverShoot = e^{\frac{-\xi\pi}{\sqrt{1-\xi}}} = . \\ \xi \approx 1, \omega_n \approx 0.66 \end{array} \right.$$

در نتیجه قطب های غالب ما دو قطب اول خواهند و برای ۴ قطب دیگر مقادیر زیر را در نظر می گیریم:

```
new_poles =
-0.08362
-0.05301
- 0.5679 - 0.4555i
- 0.5679 + 0.4555i
- 0.4707 + 0.3203i
- 0.4707 - 0.3203i
```

که در نهایت K فیدبک حالت به صورت زیر به دست می آید:

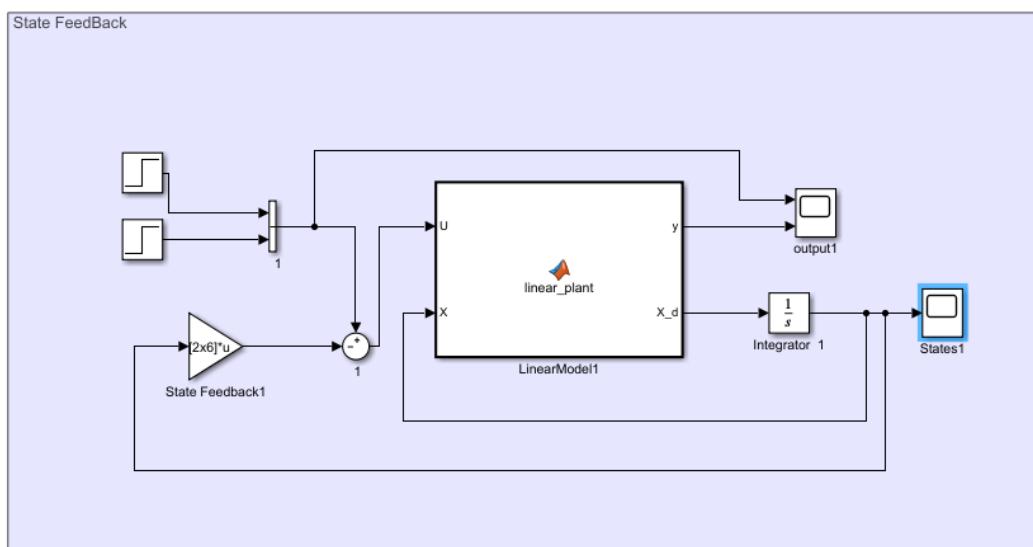
```
%% State Feedback
% new_poles = [-0.08362, -0.05301, - 0.5679 - 0.4555i, - 0.5679 + 0.4555i,- 0.4707 + 0.3203i,- 0.4707 - 0.3203i ];
% K = place(A, B, new_poles);
% vpa(K, 3)

K_stat = [106.15 -61.299 0.111 0.986 160.72 57.96; 169.55 189.92 1.35 0.106 -43.79 -62.97]
```

۳-۲ شبیه سازی و تحلیل سیستم غیر خطی و خطی شده با فیدبک حالت

۱-۳-۲ شبیه سازی سیستم خطی شده

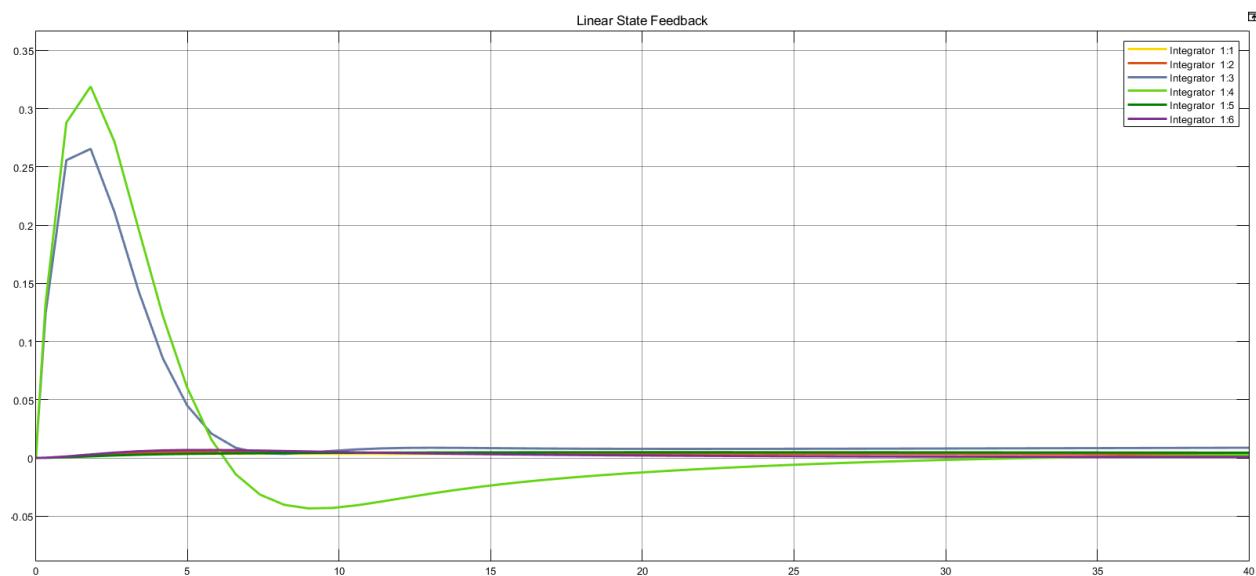
سیستم خطی شده با فیدبک حالت به صورت زیر است که در آن ورودی پله، شرایط اولیه صفر و مقادیر ماتریس های سیستم نیز مطابق قسمت های اول در **matlab function** تعریف شده اند.



کد داخل **LinearModel** به صورت زیر است:

```
function [y,X_d] = linear_plant(U,X)
A = [ -0.0189, 0.029, 0, 0.00306, 0, 0;
      0, -0.029, 0.0065, 0, 0, 0;
      0, 0, -0.476, 0, 0, 0;
      0, 0, 0, -0.5, 0, 0;
      0, 0, 0.0035, 0, -0.0191, 0.0349;
      0, 0, 0, 0.00714, 0, -0.0349];
B =[0, 0;
     0, 0;
     0, 0.476;
     0.5, 0;
     0, 0;
     0, 0];
C = [1.0, 0, 0, 0, 0, 0;
      0, 0, 0, 0, 1.0, 0];
X_d = A * X + B * U;
y = C*X;
```

در نتیجه آن نمودار حالت های سیستم با ورودی پله و شرایط اولیه صفر به صورت زیر به سمت صفر همگرا خواهد شد که مطلوب است و سیستم علاوه بر حفظ پایداری خود، نتیجه مطلوب را ارائه می دهد.



۲-۳-۲ شبیه سازی سیستم غیرخطی

قبل از اجرای شبیه سازی قسمت غیر خطی باید فایل متلب مربوطه را اجرا کرد. در این فایل ابتدا معادلات و مقادیر سیستم مشابه فاز قبل تعریف شده اند که از آوردن کدشان صرف نظر می کنیم.

در بخش بعد باید برای دو ورودی ای که داریم و همچنین حالت های سیستم نقطه تعادل را پیدا کنیم و اطمینان داشته باشیم که هیچ یک از حالت ها منفی نمی شوند. در نتیجه ۴ معادله تشکیل می دهیم که در آن ها مقادیر تعادل خروجی های اصلی ما (همان h_1 و h_2) معلوم هستند و باید نقاط تعادل h_3 , h_4 و ورودی های v_1 , v_2 را بیابیم. با حل این معادلات بر حسب ۴ متغیر ذکر شده، به مقدار تعادل سایر حالت ها و ورودی ها نیز می رسمیم و برای درست اجرا شدن شبیه سازی نیاز داریم که این مقادیر را به حلقه اضافه کنیم.

```
% Linearization and Steady-State model
F = [h1d; h2d; h3d; h4d];

E1 = subs(h1d, [h1,h2], [h1_eq, h2_eq]);
E2 = subs(h2d, [h1,h2], [h1_eq, h2_eq]);
E3 = subs(h3d, [h1,h2], [h1_eq, h2_eq]);
E4 = subs(h4d, [h1,h2], [h1_eq, h2_eq]);

eq_values = zeros(4,1);
tmp = vpasolve([E1==0; E2==0; E3==0; E4==0], [h3 h4 v1 v2]);
total_ans = fieldnames(tmp);
for i=1:4
    eq_values(i,1) = real(double(getfield(tmp, total_ans{i})));
end

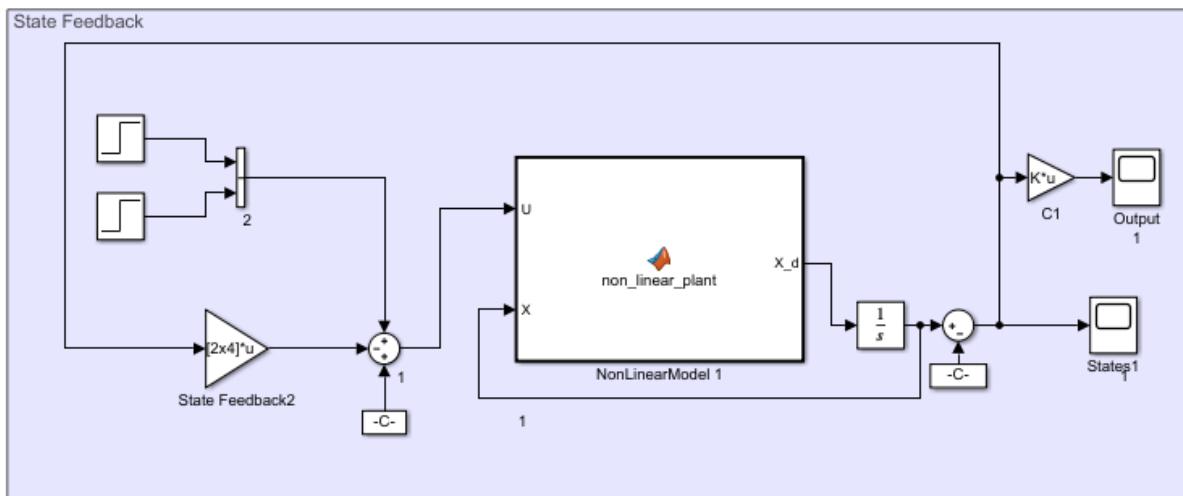
Amat = jacobian(F,[h1,h2,h3,h4]);
Amat = subs(Amat,[h1,h2,h3,h4,v1,v2],[h1_eq,h2_eq,h3_eq,h4_eq,v1_eq,v2_eq]);
A = double(Amat);

Bmat = jacobian(F,[v1,v2]);
Bmat = subs(Bmat,[h1,h2,h3,h4,v1,v2],[h1_eq,h2_eq,h3_eq,h4_eq,v1_eq,v2_eq]);
B = double(Bmat);

% outputs are h1 & h2 (level of tank 1 and tank 2)
C=[1 0 0 0;
   0 1 0 0];

D = [0 0 ;
      0 0];
```

سیستم غیرخطی با فیدبک حالت به صورت زیر است که در آن ورودی پله و شرایط اولیه نقاط تعادل سیستم هستند.



کد داخل non_linear_plant به صورت زیر است که شامل معادلات و مقادیر سیستم اصلی می باشد:

```

function X_d = non_linear_plant(U,X)
Area = [730 730 730 730]; % area of the tanks in cm^2
a = [2.10 2.14 2.2 2.3]; % area of the drain in cm^2
y = [0.3 0.35]; % y(1) = Ratio of flow in tank1 to flow in tank4
% y(2) = Ratio of flow in tank2 to flow in tank3
k = [7.45 7.30]; % pump proportionality constants in cm^2/s
g = 981; % gravitational acceleration in cm/s^2

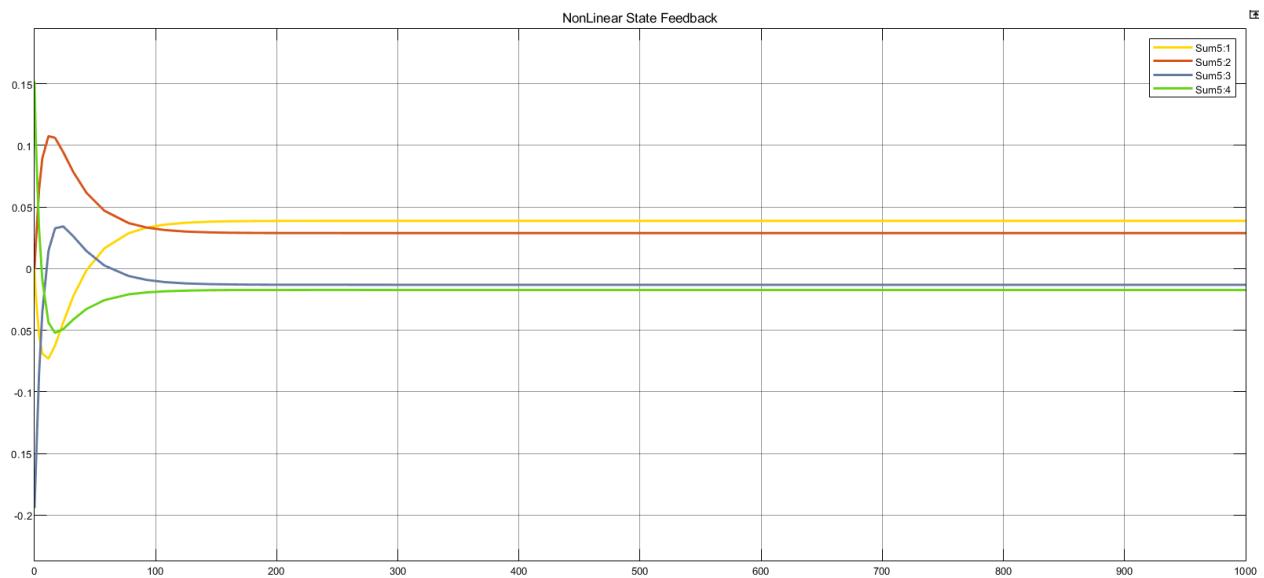
h1 = X(1);
h2 = X(2);
h3 = X(3);
h4 = X(4);
|
v1 = U(1);
v2 = U(2);

h1_d = -a(1)/Area(1)*sqrt(2*g*h1)+a(3)/Area(1)*sqrt(2*g*h3)+y(1)*k(1)/Area(1)*v1+0.00137;
h2_d = -a(2)/Area(2)*sqrt(2*g*h2)+a(4)/Area(2)*sqrt(2*g*h4)+y(2)*k(2)/Area(2)*v2+0.00137;
h3_d = -a(3)/Area(3)*sqrt(2*g*h3)+(1 - y(2))*k(2)/Area(3)*v2;
h4_d = -a(4)/Area(4)*sqrt(2*g*h4)+(1 - y(1))*k(1)/Area(4)*v1;

X_d = [h1_d;h2_d;h3_d;h4_d];

```

در نتیجه آن نمودار حالت های سیستم با ورودی پله و شرایط اولیه ذکر شده به صورت زیر پس از گذشت زمان کوتاهی پایدار خواهند شد که نتیجه مطلوب است.

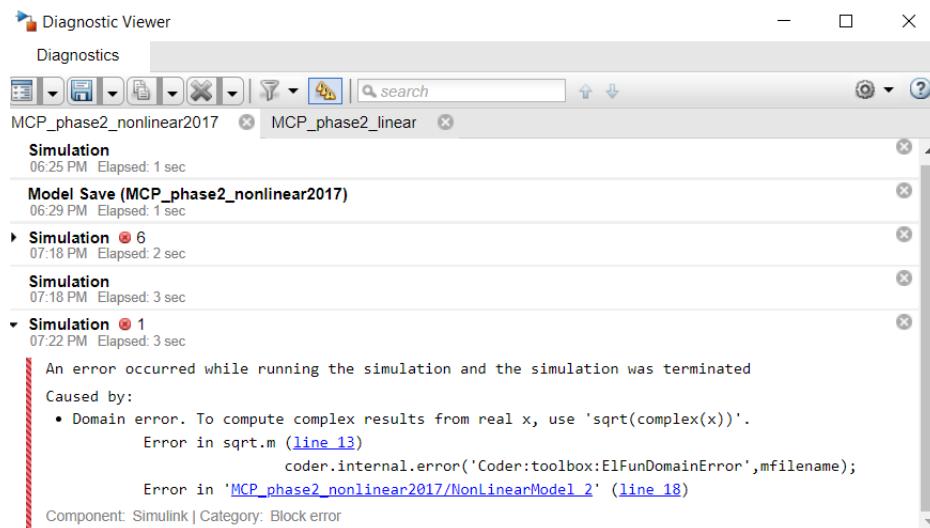


۴-۲ ناپایداری سیستم غیر خطی با فیدبک حالت

برای بررسی سیستم غیرخطی با فیدبک حالت خطی باید سیستم را در حدود بازه نقطه تعادل و با شرایط اول برابر با نقاط تعادل در نظر بگیریم که حوزه همگرایی سیستم غیر خطی است.

از آنجایی که شرایط اولیه سیستم را حول نقاط تعادل در نظر گرفتیم، سیستم غیر خطی در حوزه همگرایی خود قرار گرفته و با فیدبک حالت پایدار خواهد ماند. اما اگر سیستم غیر خطی را خارج از حوزه همگرایی اش بررسی کنیم ممکن است سیستم با این فیدبک حالت ناپایدار شود.

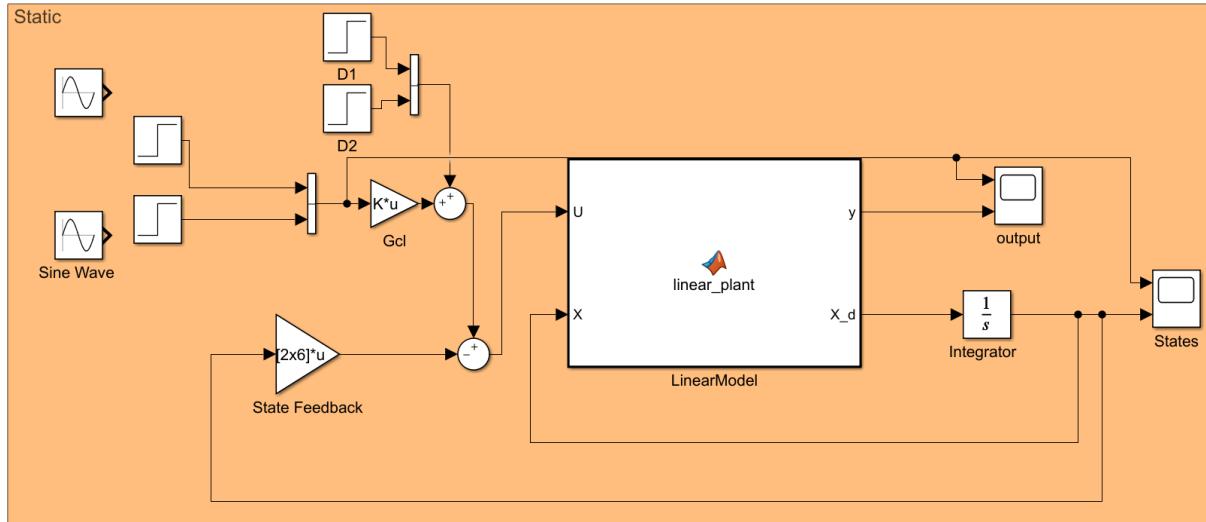
به عنوان نمونه اگر شرایط اولیه را برابر صفر در نظر بگیریم به دلیل منفی شدن مقادیر خروجی در شبیه سازی به خطأ برخورد می کنیم.



۵-۲ شبیه سازی پیش جبرانساز استاتیکی و دینامیکی

۱-۵-۲ پیش جبرانساز استاتیکی سیستم خطی شده

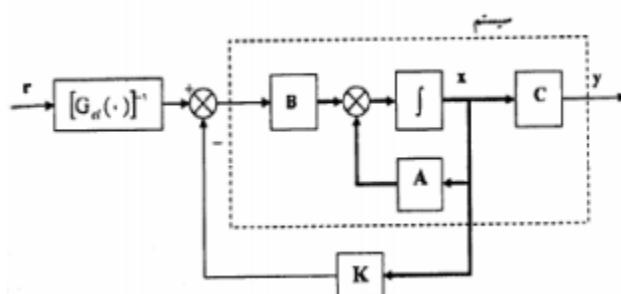
سیستم خطی شده با پیش جبرانساز استاتیکی به صورت زیر می باشد:



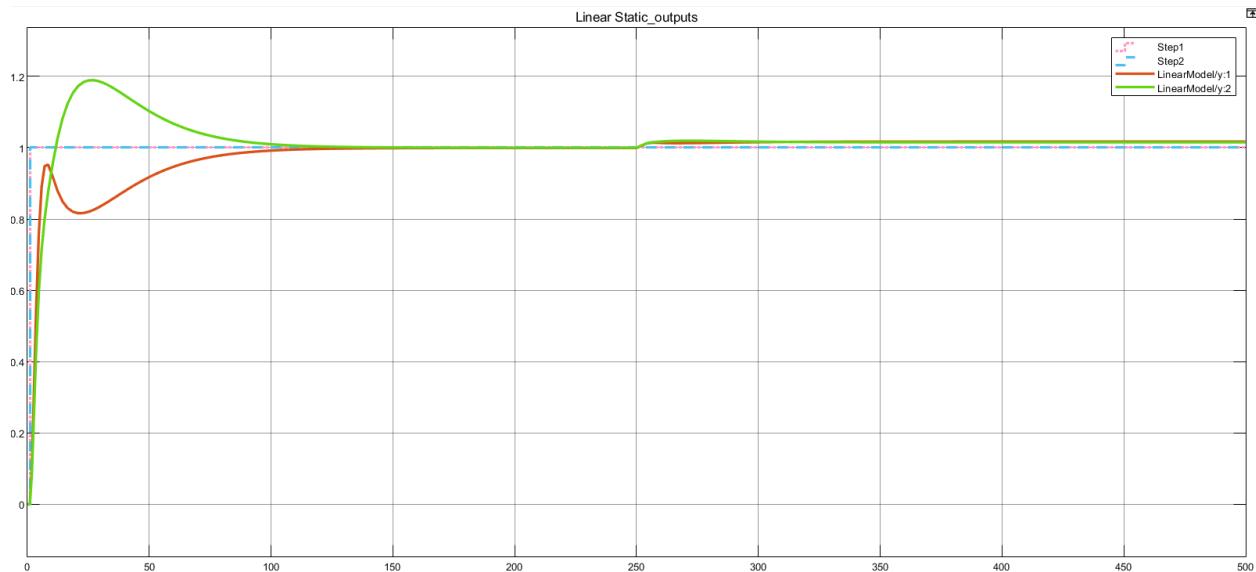
ماتریس K همان ماتریس فیدبک حالت است و $G_{CL}(s)$ در ادامه با استفاده ازتابع تبدیل سیستم محاسبه می شود:

```
%% Static Compensator
G_c10 = inv(C*inv(-(A-B*K_stat))*B)
```

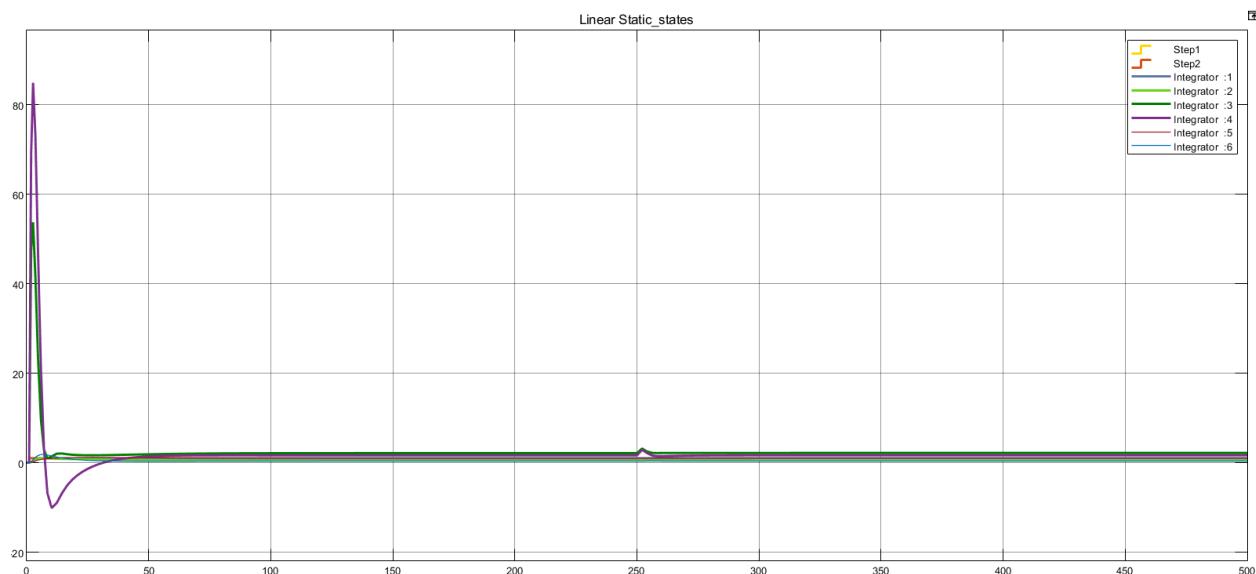
```
G_c10 =
29.0883 231.0511
362.7551 -161.5609
```



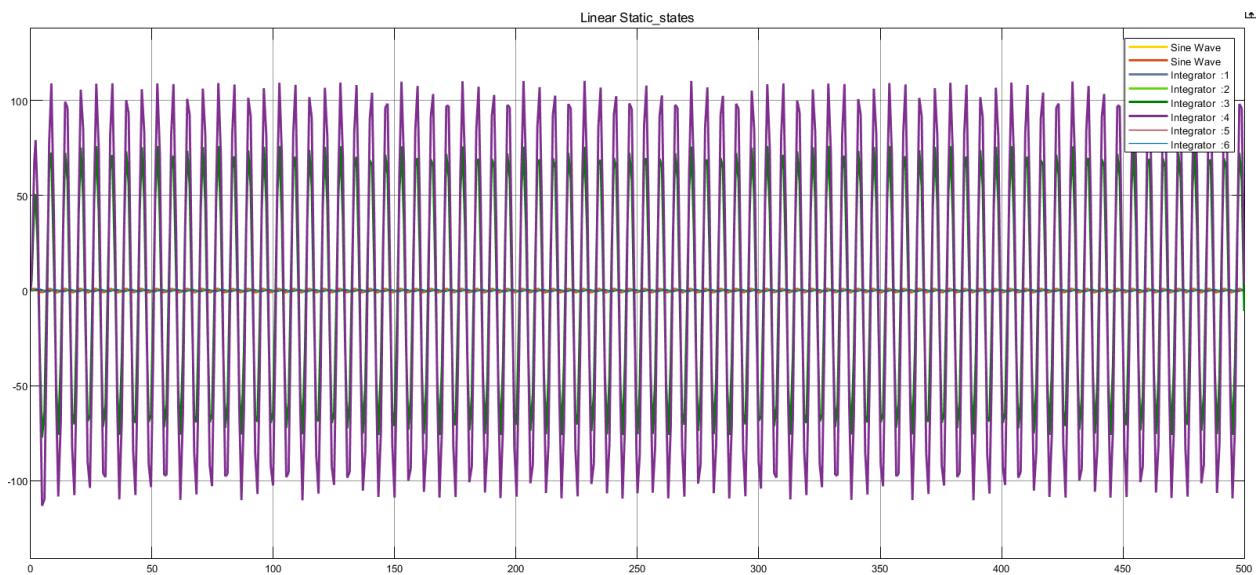
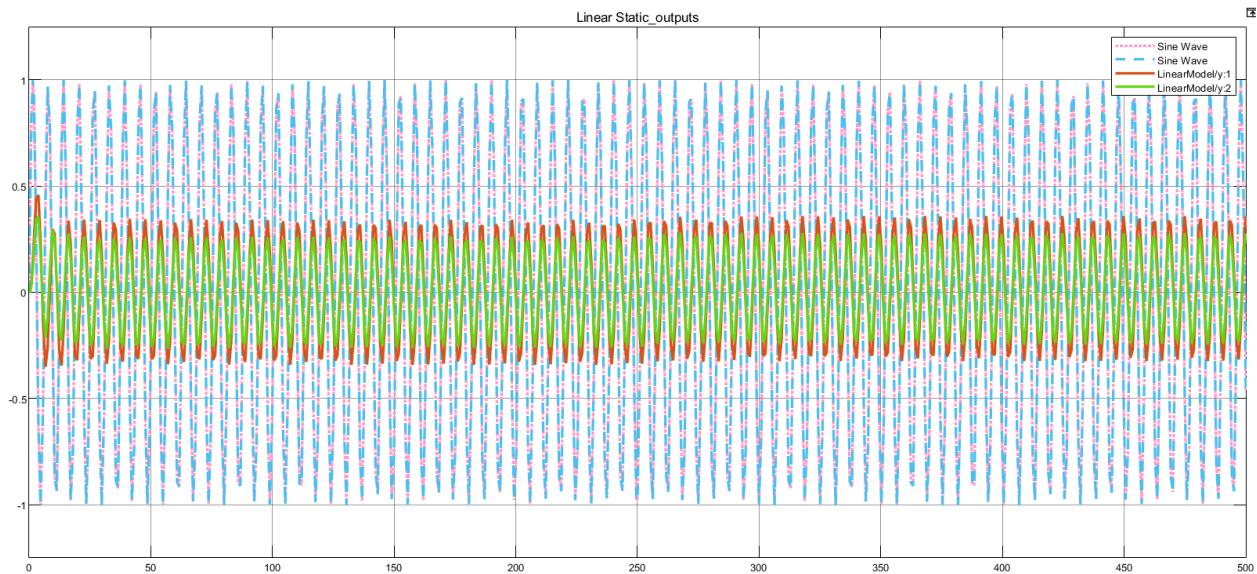
در نهایت خروجی ها به ورودی مرجع پله و اغتشاش ثابت (با مقدار 4° و در زمان 500) به صورت زیر خواهند شد که مشاهده می شود قبل از اعمال اغتشاش ثابت سیستم پس از گذشت زمان نشست می تواند ورودی پله را ردیابی کند اما پس از اعمال اغتشاش دیگر نتوانسته و خطای ماندگار ثابتی با ورودی پله خواهد داشت.



نمودار متغیر های حالت نیز به صورت زیر خواهد شد که همگی به میرایی مطلوب می رسد.

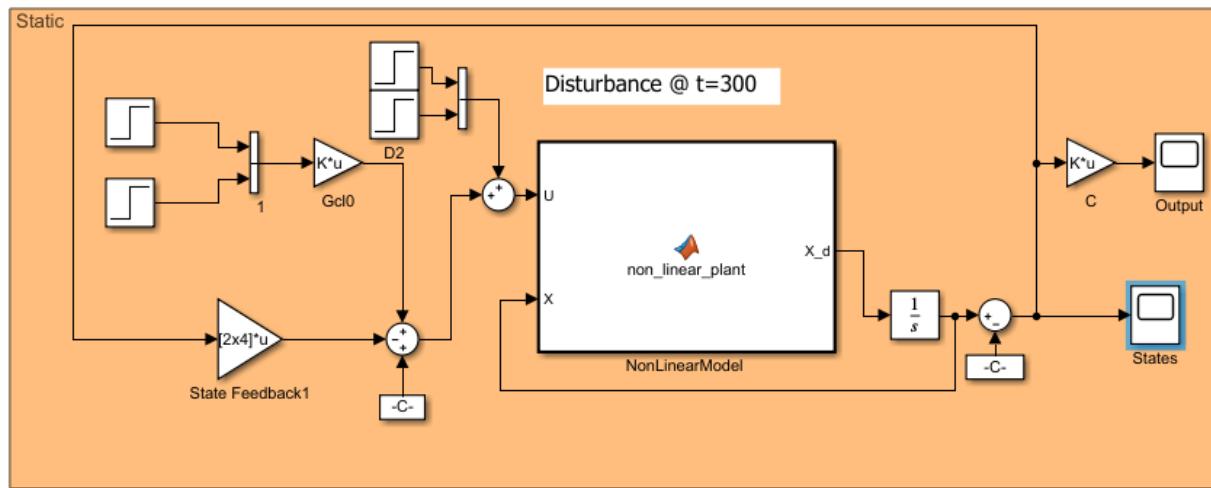


خروجی ها و حالت ها با سیگنال مرجع سینوسی نیز به صورت زیر خواهند شد که خروجی ورودی مرجع را ردیابی نمی کند و فاز و دامنه خروجی با ورودی متفاوت است:



۲-۵-۲ پیش جبرانساز استاتیکی سیستم غیر خطی

سیستم غیرخطی با پیش جبرانساز استاتیکی به صورت زیر می باشد:



ماتریس K (ماتریس فیدبک حالت) و $G_{CL}(\cdot)$ در ادامه با استفاده از تابع تبدیل سیستم محاسبه می شود و شرایط اولیه مشابه قبل حول نقطه تعادل سیستم است:

```
%> Non-linear static compensator
P_non_stat = [-0.2, -0.15, -0.04, -0.08];
K_non_stat = place(A, B, P_non_stat);
K_non_stat = vpa(K_non_stat, 3)
G_c10_non = vpa(inv(C*inv(-(A-B*K_non_stat))*B), 3)
```

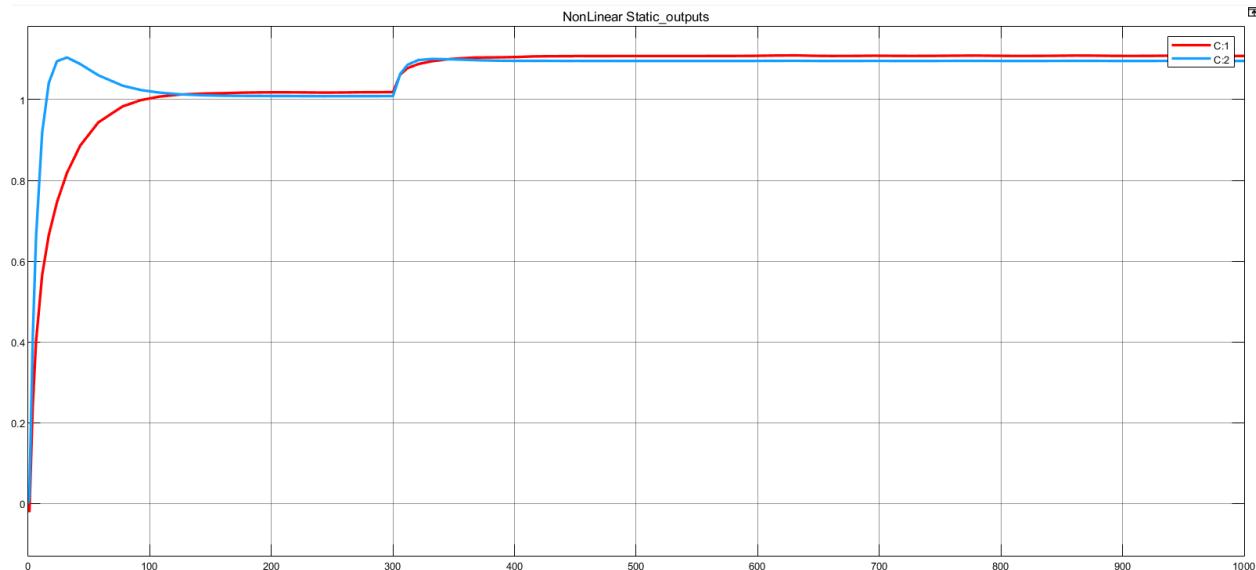
```
K_non_stat =
```

```
[ 10.1, 33.9, -13.2, 24.4]
[ 23.5, 12.2, 18.5, -4.03]
```

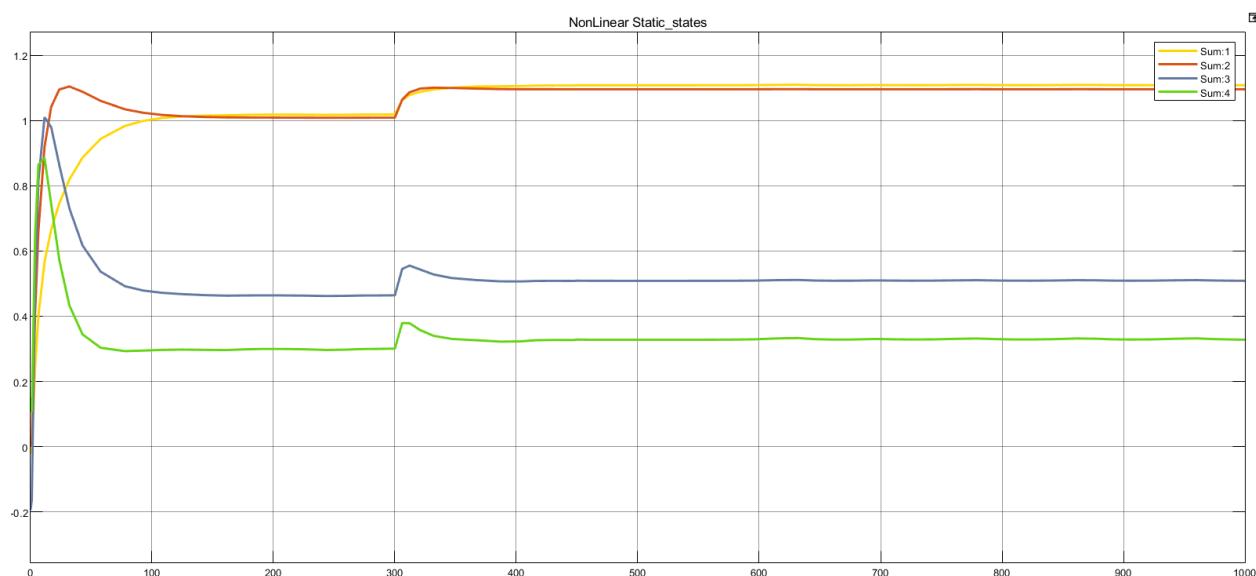
```
G_c10_non =
```

```
[ -12.2, 59.5]
[ 44.5, 0.91]
```

در نهایت خروجی ها به ورودی مرجع پله و اغتشاش ثابت (با مقدار 4° و در زمان ۳۰۰) به صورت زیر خواهند شد که مشاهده می شود قبل از اعمال اغتشاش ثابت سیستم پس از گذشت زمان نشست می تواند ورودی پله را ردیابی کند اما پس از اعمال اغتشاش دیگر نتوانسته و خطای ماندگار ثابتی با ورودی پله خواهد داشت.

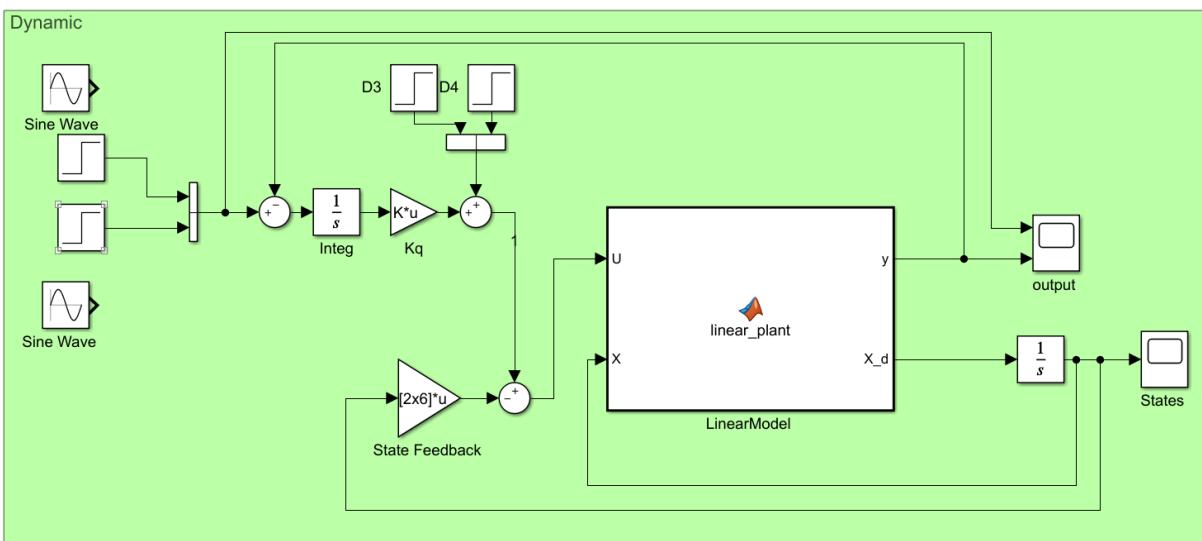


نمودار متغیر های حالت نیز به صورت زیر خواهد شد که همگی پس از گذشت زمان کوتاهی پایدار خواهند شد که مطلوب است:



۳-۵-۲ پیش جبرانساز دینامیکی سیستم خطی شده

سیستم خطی شده با پیش جبرانساز دینامیکی به صورت زیر می باشد:



که در آن مقدار K فیدبک حالت و Kq پیش جبرانساز به صورت زیر به دست آمده اند که پس از بررسی شرط لازم (فول رنک بودن ماتریس کنترل پذیری)، با تعیین قطب های مطلوب و ایجاد فضای حالت جدید براساس روابط مقادیر K فیدبک حالت و Kq پیش جبرانساز محاسبه می شوند:

```

%% Dynamic Compensator
% check for the necessary condition
if rank([B, A; zeros(2,2), -C]) == 8
    P = [-0.12, -0.05, -0.025, -0.2, -0.08, -0.14, -0.22, -0.1];
    Abar=[A, zeros(size(A,1),size(C,1));-C ,zeros(size(C,1),size(C,1))];
    Bbar=[B; zeros(size(C,1),2)];
    K = place(Abar, Bbar, P);
    K = vpa(K, 3);
    K_dyn = K(:, 1:6);
    Kq = K(:, 7:8);
end

```

$$u(t) = \begin{bmatrix} -K_1 & K_2 \end{bmatrix} \begin{bmatrix} x(t) \\ q(t) \end{bmatrix}$$

$$\dot{x}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A - BK_1 & BK_2 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} r + \begin{bmatrix} d \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \bar{B} & \bar{AB} & \dots & (\bar{A})^{n-1}\bar{B} \end{bmatrix} = \begin{bmatrix} B & AB & A^2B & \dots & A^nB \\ 0 & -CB & -CAB & \dots & -CA^{n-1}B \end{bmatrix}$$

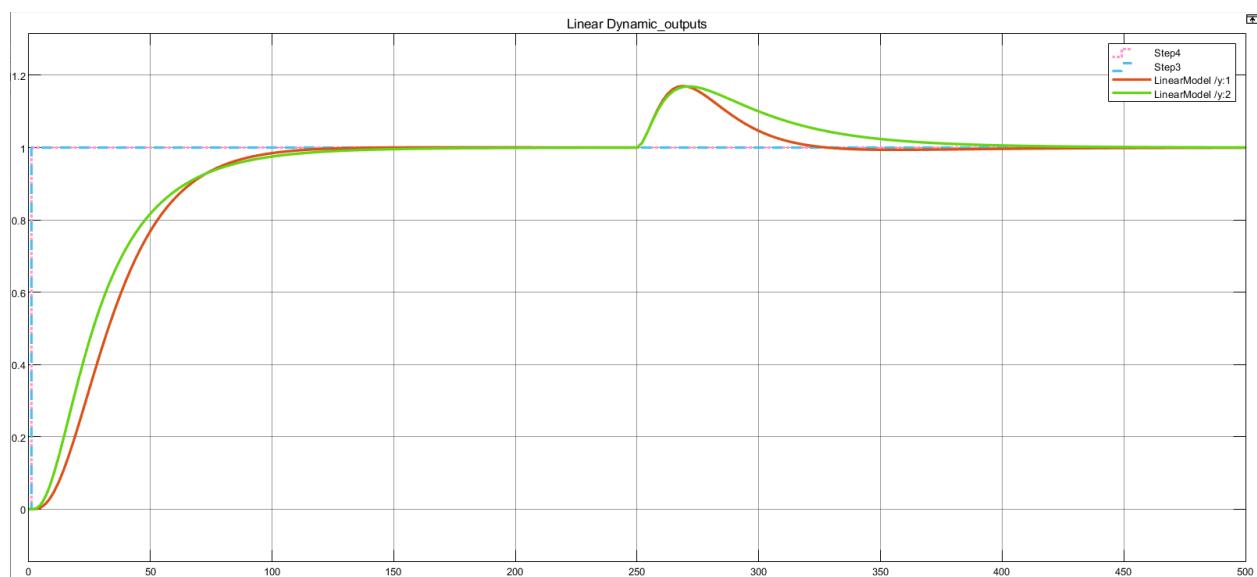
$$= \begin{bmatrix} B & A\Phi_c \\ 0 & -C\Phi_c \end{bmatrix}$$

$$= \begin{bmatrix} B & A \\ 0 & -C \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Phi_c \end{bmatrix}$$

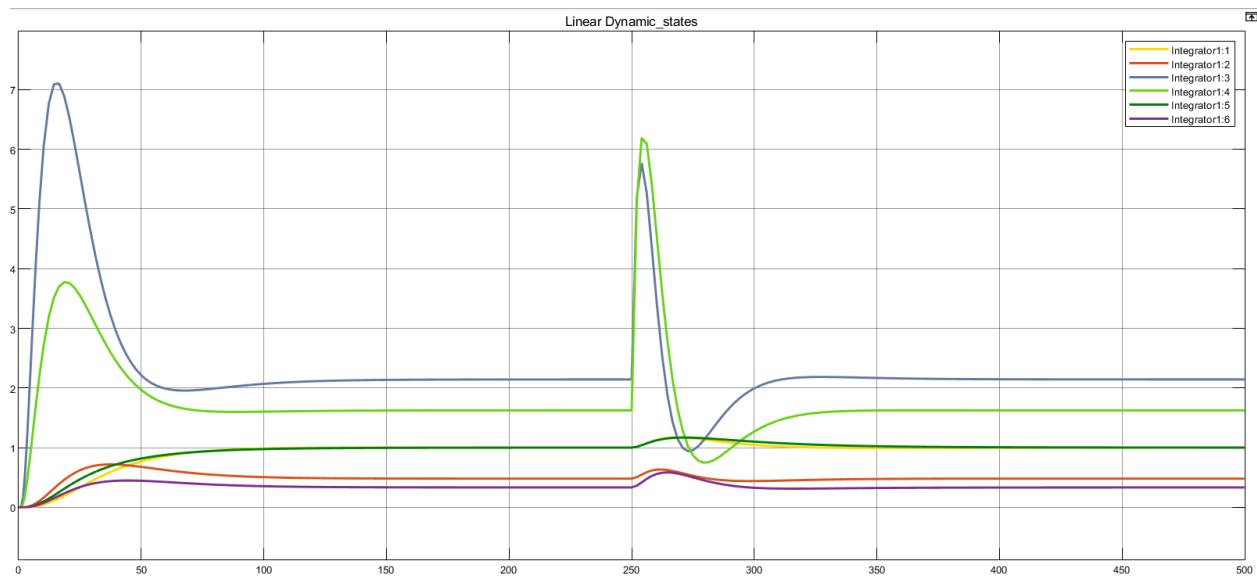
روتیه کامل است

شرط لازم و کافی بودن سیستم افزوده
شده روتیه کامل بودن این ماتریس است

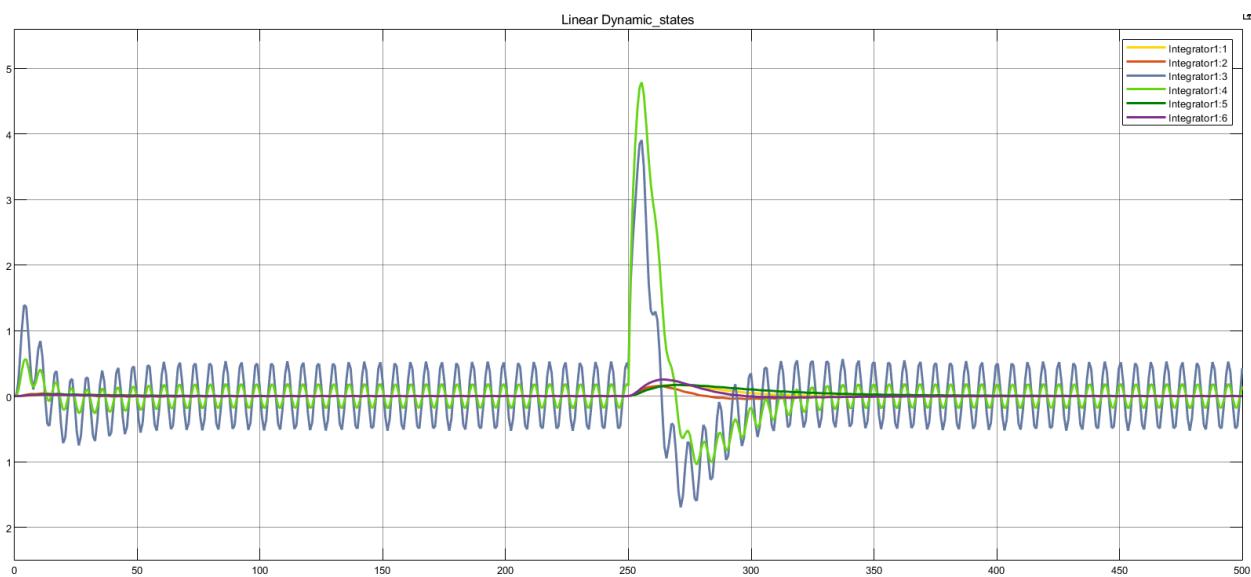
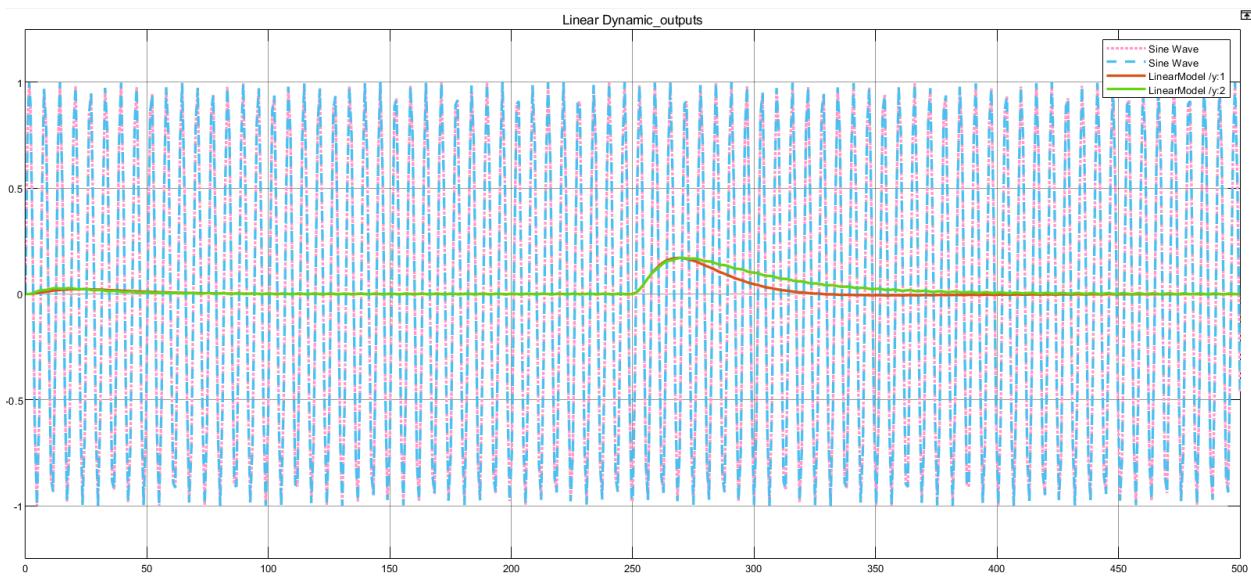
در نهایت خروجی ها به ورودی مرجع پله و اغتشاش ثابت (با مقدار ۵ و در زمان ۲۵۰) به صورت زیر خواهند شد که مشاهده می شود قبل از اعمال اغتشاش ثابت سیستم پس از گذشت زمان نشست می تواند ورودی پله را ردیابی کند. همچنین پس از اعمال اغتشاش نیز مطابق انتظار توانسته اثر اغتشاش ثابت را حذف کرده و ورودی را به طور کامل ردیابی کند.



نمودار متغیر های حالت نیز به صورت زیر خواهد شد که همگی پس از گذشت زمان کوتاهی پایدار خواهند شد که مطلوب است:

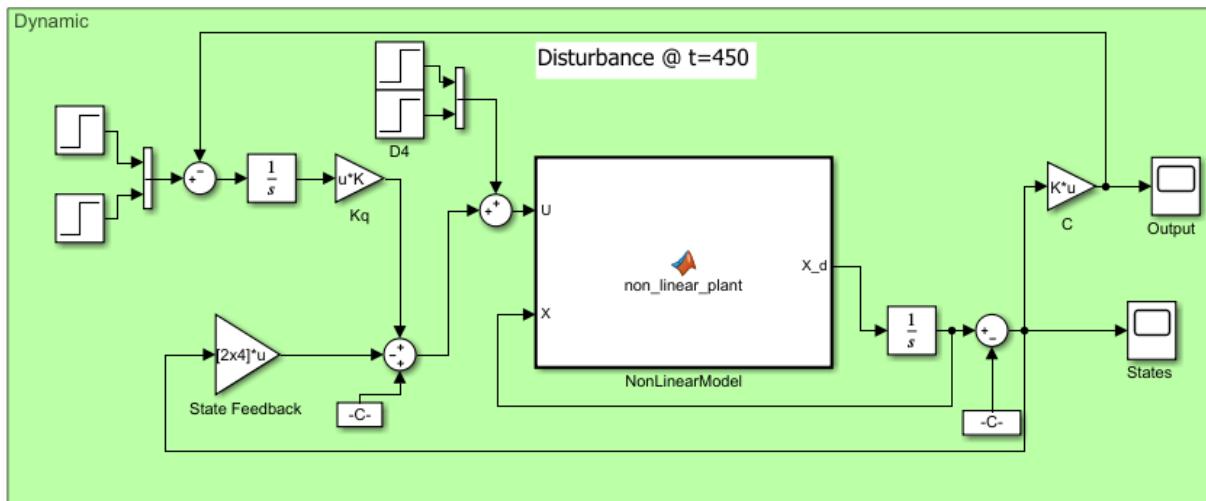


خروجی ها و حالت ها با سیگنال مرجع سینوسی نیز به صورت زیر خواهند شد که خروجی ورودی مرجع را ردیابی نمی کند. (در حالت دینامیکی سیستم مقدار dc سینوس را تعقیب می کند)



۴-۵-۲ پیش جبرانساز دینامیکی سیستم غیر خطی

سیستم غیرخطی با پیش جبرانساز دینامیکی به صورت زیر می باشد:



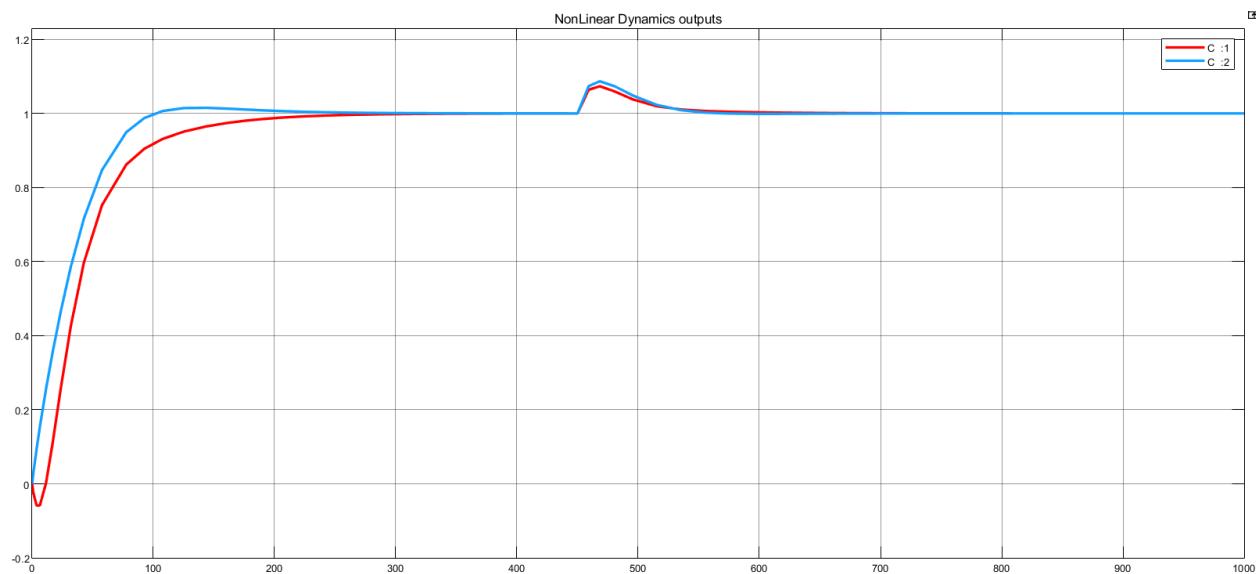
که در آن مقدار K فیدبک حالت و Kq پیش جبرانساز مشابه حالت خطی به دست آمده اند و شرایط اولیه مشابه قبل حول نقطه تعادل سیستم است:

```

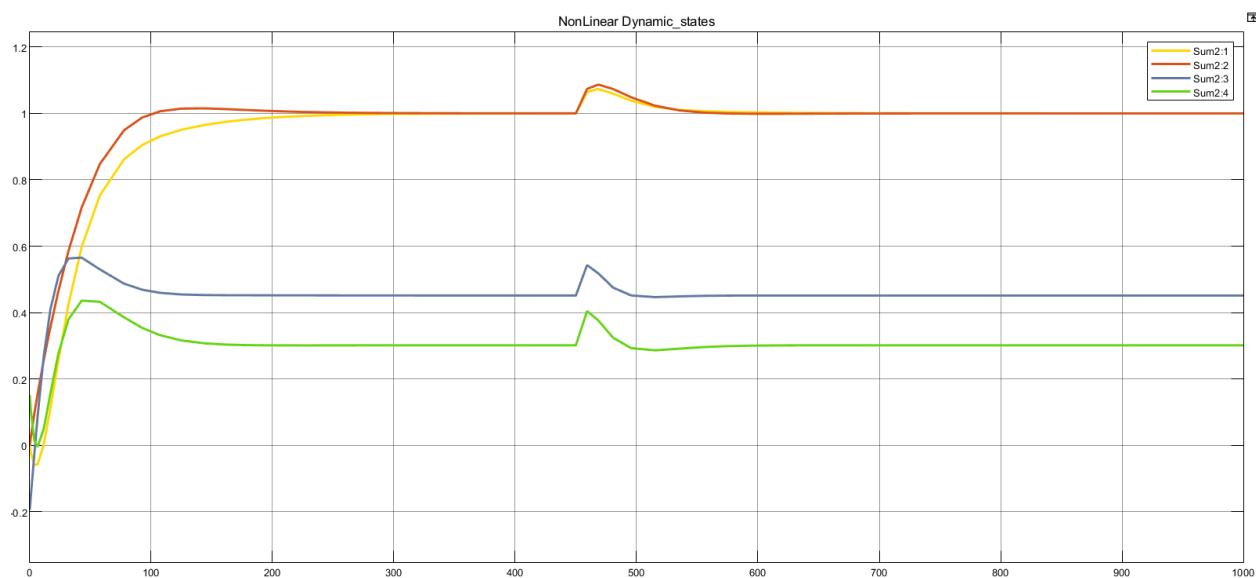
%% Dynamic Compensator
% check for the necessary condition
if rank([B, A; zeros(2,2), -C]) == 6
    P = [-0.05, -0.1, -0.08, -0.1, -0.02, -0.04];
    Abar=[A, zeros(size(A,1),size(C,1));-C ,zeros(size(C,1),size(C,1))];
    Bbar=[B; zeros(size(C,1),2)];
    K = place(Abar, Bbar, P);
    K = vpa(K, 3);
    K_dyn = K(:,1:4);
    Kq = K(:, 5:6);
end

```

در نهایت خروجی ها به ورودی مرجع پله و اغتشاش ثابت (با مقدار 4° و در زمان ۴۵۰) به صورت زیر خواهند شد که مشاهده می شود قبل از اعمال اغتشاش ثابت سیستم پس از گذشت زمان نشست می تواند ورودی پله را ردیابی کند. همچنین پس از اعمال اغتشاش نیز مطابق انتظار توانسته اثر اغتشاش ثابت را حذف کرده و ورودی را به طور کامل ردیابی کند.



نمودار متغیر های حالت نیز به صورت زیر خواهد شد که همگی پس از گذشت زمان کوتاهی پایدار خواهند شد که مطلوب است:



۶-۲ شبیه سازی پیش جبرانساز ردیابی سیگنال مرجع سینوسی

قبل از اجرای شبیه سازی لازم است تا کد متلب زیر یک دور ران شود. در این قسمت ازتابع تبدیل دوم یعنی از ورودی اول به خروجی دوم استفاده می کنیم و در قسمت ابتدایی کد، تابع تبدیل را به فضای حالت تبدیل می کنیم و سپس فاکتور ها و قطب های تابع تبدیل را محاسبه می کنیم. در ادامه در دو حالت بدون اغتشاش و با اغتشاش مقدار بهره فیدبک حالت و رویتگر را با دستور place و قطب های مورد نظر به دست می آوریم که باید قطب های رویتگر چند برابر دورتر از قطب های فیدبک حالت باشند و در نهایت مطابق روابط زیر کنترلر خواسته شده را به دست می آوریم.

```
%% Sine tracking
% input 1 to output 2
num = 0.0001246;
den = [1 0.554 0.02764 0.0003325];
[A, B, C, D] = tf2ss(num, den);

factors = factor(poly2sym(den));
OL_Poles = roots(den);



---


%% without disturbance
num_1 = num;
den_1 = conv(den, [1 0 1]);
[A1, B1, C1, D1] = tf2ss(num_1, den_1);

K1 = place(A1,B1, [-0.025, -0.05, -0.25, -0.9, -1]);
L1 = place(A1', C1', [-5, -3, -4, -6, -4.5])';

% K1 controller
[b1, a1] = ss2tf(A1-B1*K1-L1*C1,L1,K1,0);

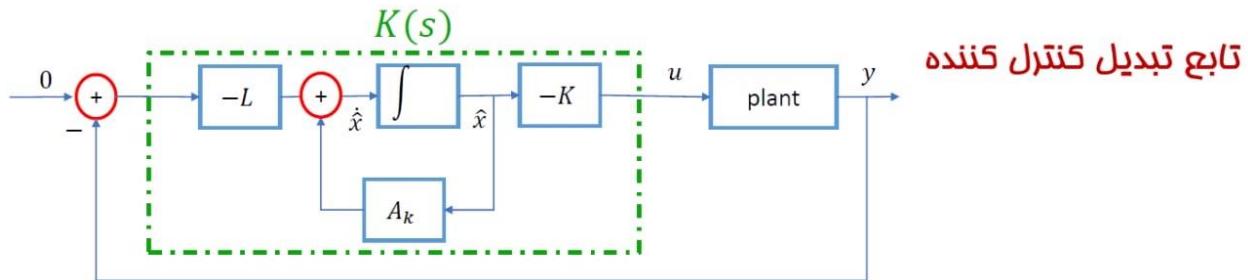


---


%% with disturbance
num_2 = num;
% Considering r_d & w dynamics (1/(s(s^2+1)))
den_2 = conv(den, [1 0 1 0]);
[A2, B2, C2, D2] = tf2ss(num_2, den_2);

K2 = place(A2, B2, [-0.025, -0.05, -0.25, -0.9, -1, -1.1]);
L2 = place(A2', C2', [-3,-4,-5,-6,-7,-8])';
[b2, a2] = ss2tf(A2-B2*K2-L2*C2, L2, K2, 0);
```

روابط ردیابی سیگنال سینوسی:

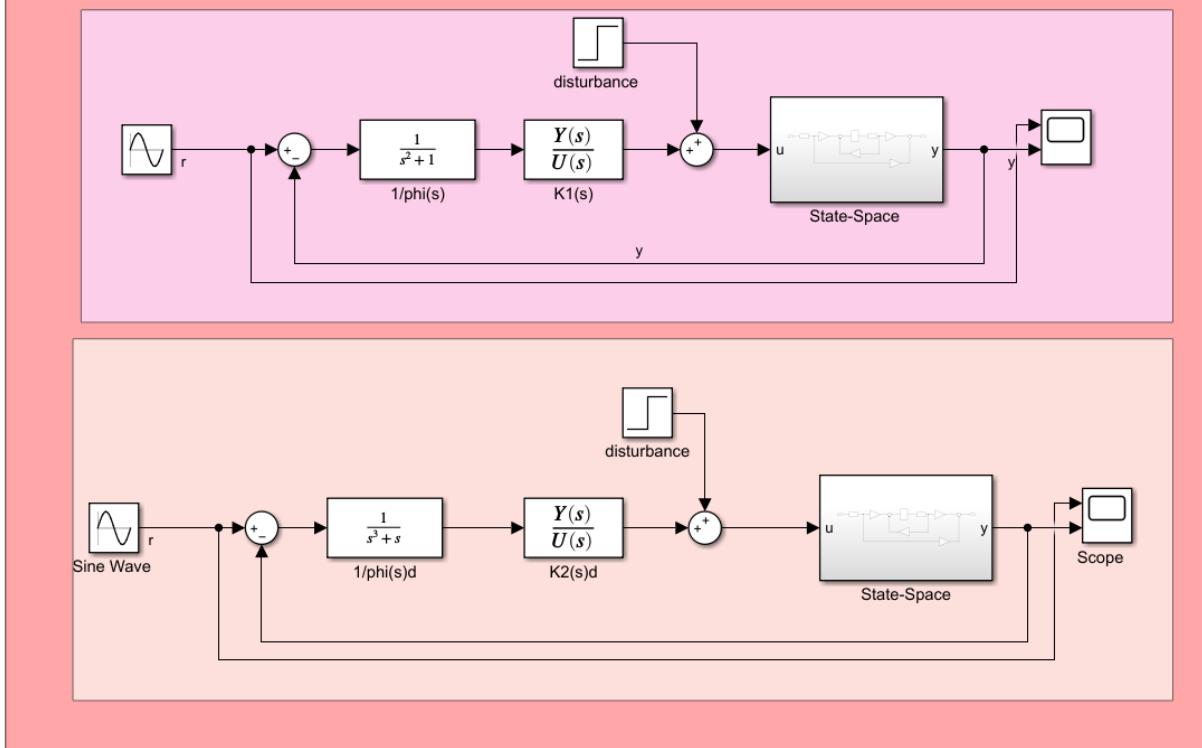


$$\underbrace{\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}}_{\text{system}} \oplus \underbrace{\begin{cases} \dot{x}(t) = (A - BK - LC)\hat{x} + Ly \\ u(t) = -K\hat{x} \end{cases}}_{\text{controller}}$$

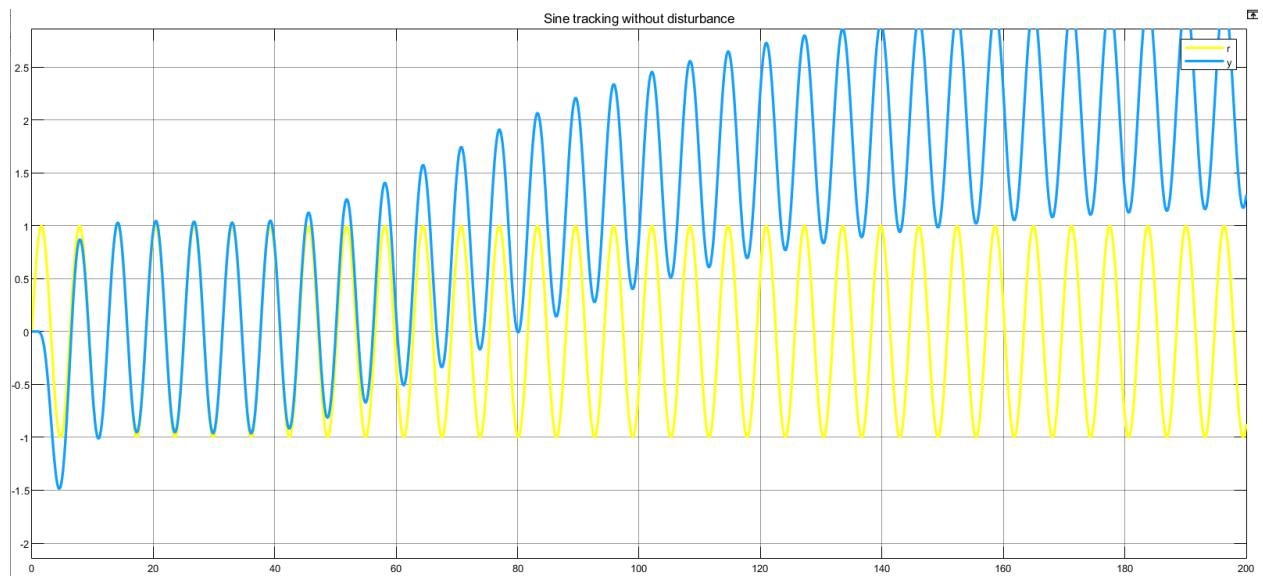
$$K(s) = K(sI - A + BK + LC)^{-1}L$$

شبیه سازی سیستم به صورت زیر خواهد شد:

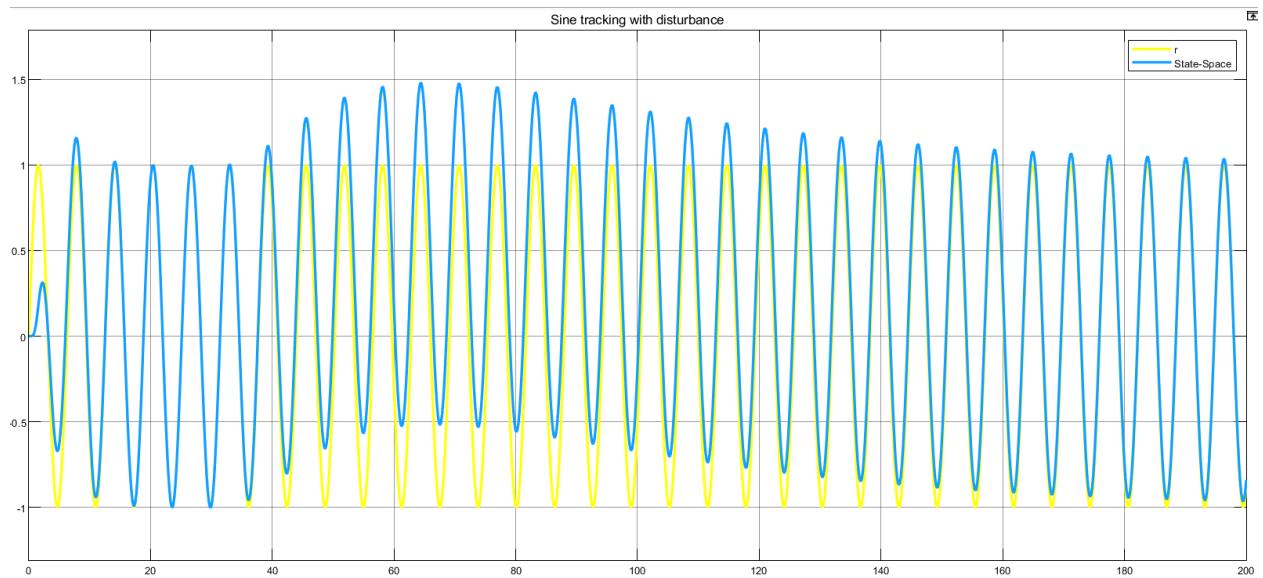
Sine Wave Tracker



به سیستم اول که سیگنال اغتشاش وارد کنیم، به دلیل آنکه در طراحی اش اثر اغتشاش لحاظ نشده، نمی‌تواند سیگنال مرجع سینوسی را ردیابی کند.



اما سیستم دوم به دلیل داشتن ترم اضافه انتگرالگیر در کنترلر توانایی ردیابی سیگنال سینوسی را پس از گذشت زمان گذر دارد.



فاز سوم پروژه

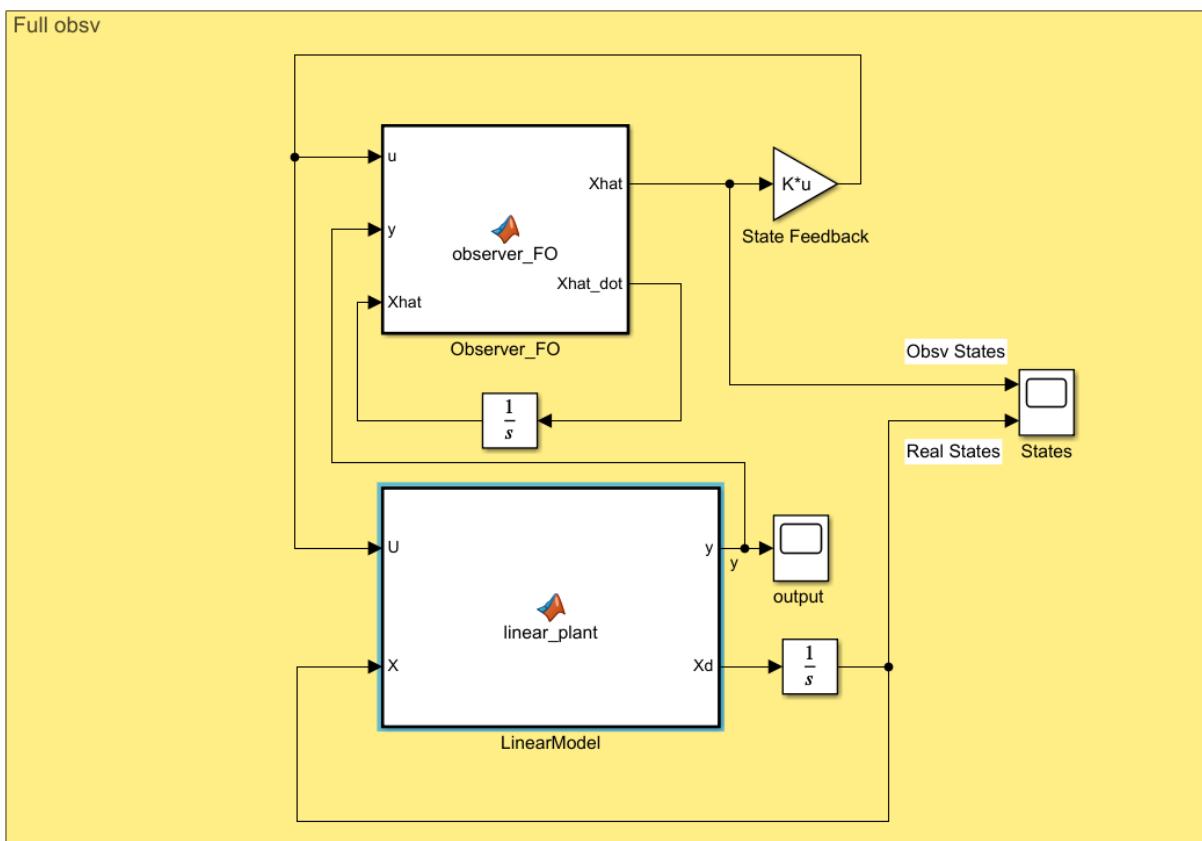
۱-۳ رویتگر مرتبه کامل و مرتبه کاهش یافته

۱-۱-۳ رویتگر مرتبه کامل

برای طراحی رویتگر کامل قطب های رویتگر را در نقاط $(-0.1, -0.2, -0.3, -0.4, -0.5, -0.6)$ قرار داده و با دستور place به صورت زیر ماتریس L را محاسبه می کنیم.

```
%% observer (full order)
FObsv_Poles = [-0.1, -0.2, -0.3, -0.4, -0.5, -0.6];
L = place(A', C', FObsv_Poles)';
```

بلوک دیاگرام رویتگر مرتبه کامل به صورت زیر خواهد شد:



که در آن از سیستم خطی شده استفاده می کنیم و روابط بلوک رویتگر کامل به صورت زیر است:

```

function [Xhat, Xhat_dot]= observer_FO(u, y, Xhat)

A = [ -0.0189, 0.029, 0, 0.00306, 0, 0;
       0, -0.029, 0.0065, 0, 0, 0;
       0, 0, -0.476, 0, 0, 0;
       0, 0, 0, -0.5, 0, 0;
       0, 0, 0.0035, 0, -0.0191, 0.0349;
       0, 0, 0, 0.00714, 0, -0.0349];

B =[0, 0;
     0, 0;
     0, 0.476;
     0.5, 0;
     0, 0;
     0, 0];

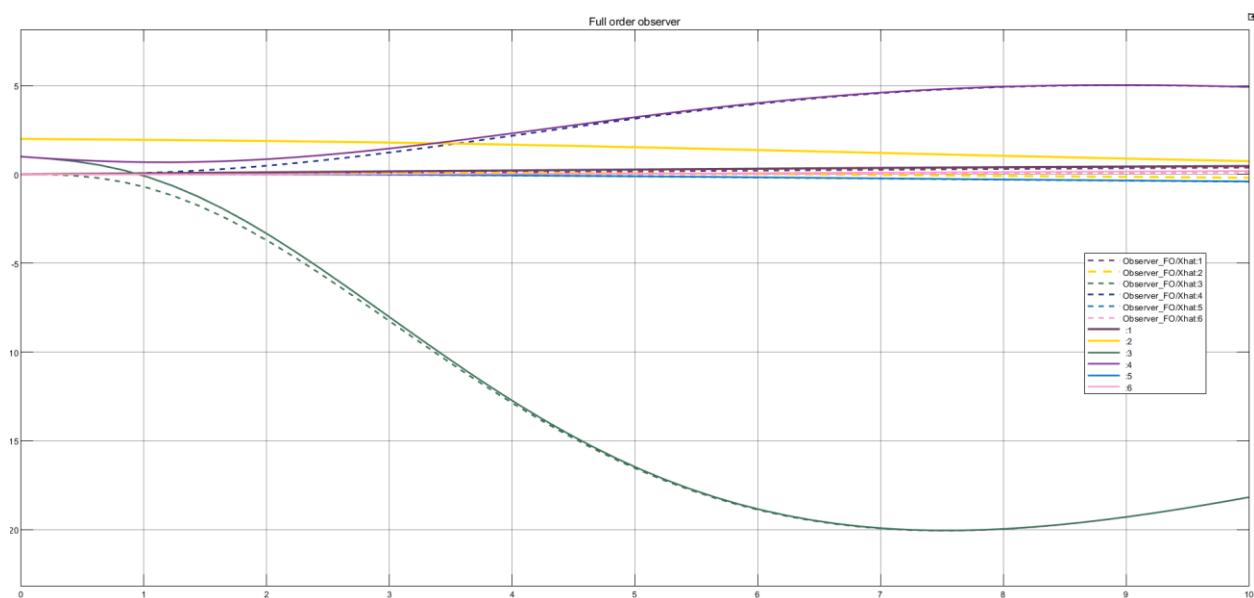
C = [1.0, 0, 0, 0, 0, 0;
      0, 0, 0, 0, 1.0, 0];

L = [0.367, -0.0766;
      0.77, -0.525;
      0.257, -1.66;
      -0.00453, 0.0113;
      -0.0573, 0.655;
      -0.323, 2.12];

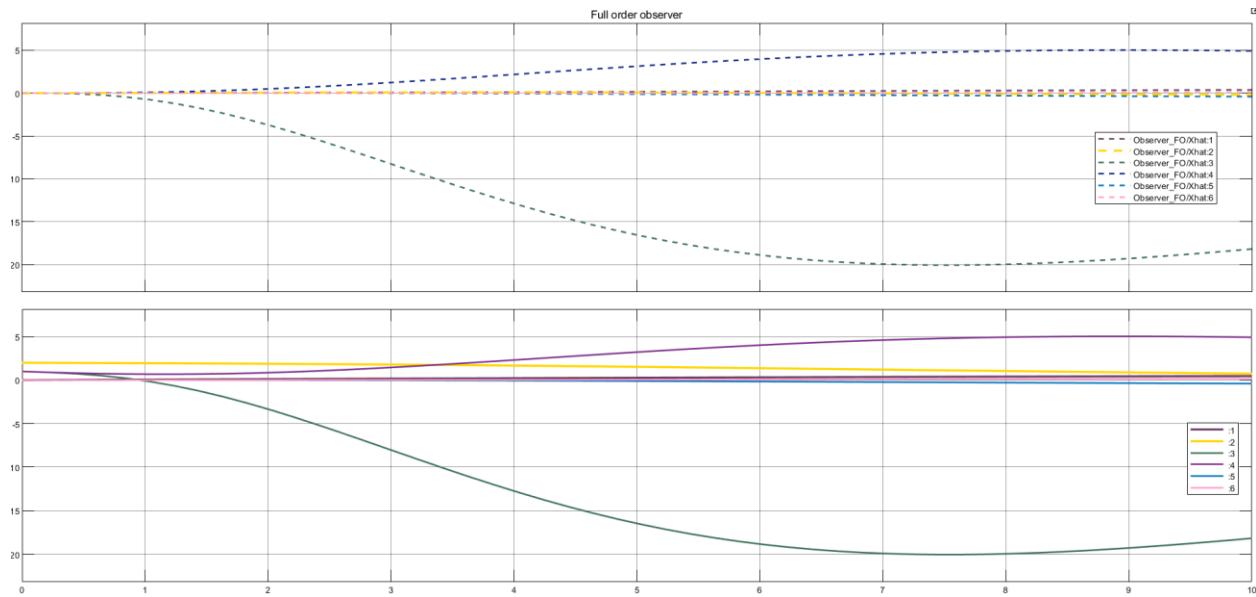
Xhat_dot = A*Xhat + B*u + L*(y - C*Xhat);

```

حالت های سیستم و رویتگر به صورت زیر خواهد شد که نشان می دهد رویتگر مرتبه کامل توانسته تا حدود خوبی حالت های اصلی را دنبال کند.



نمودار حالت ها به صورت جدا گانه نیز به شکل زیر می باشد:



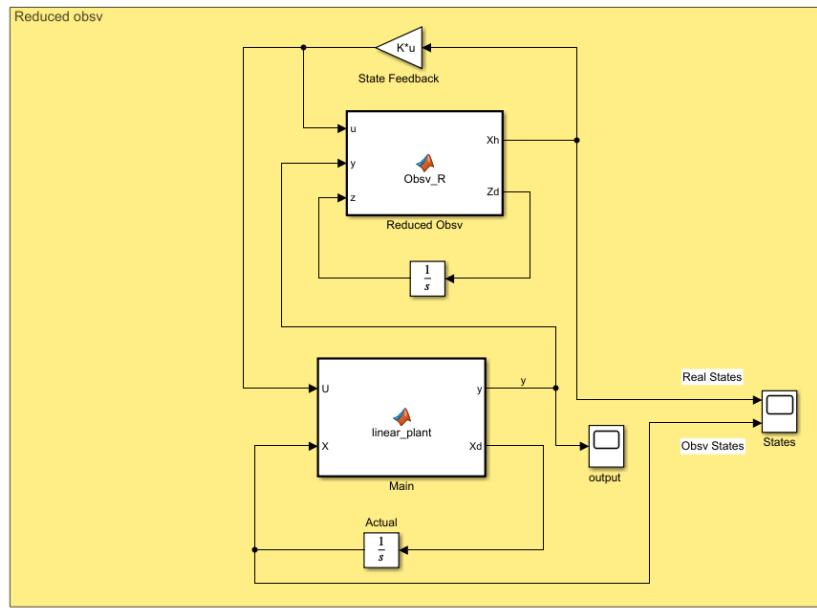
۲-۱-۳ رویتگر مرتبه کاهش یافته

برای طراحی رویتگر مرتبه کاهش یافته از دستور زیر و با استفاده رابطه لیاپانوف ماتریس های مورد نظر را به دست می آوریم و قطب های رویتگر را در $(-0.45, -0.4)$ قرار می دهیم.

```
%% observer (reduced order)

RObsv_Poles = [-0.4, -.45]; % We can't choose -0.5 cause it's exactly the eigenvalue for A matrix
F_mat = diag(RObsv_Poles);
L_mat = [1, 0; 0, -1];
T_mat = (lyap(A', -F_mat', -C'*L_mat'))';
```

بلوک دیاگرام رویتگر مرتبه کاهش یافته به صورت زیر خواهد شد:



که در آن از سیستم خطی شده استفاده می‌کنیم و روابط بلوک رویتگر مرتبه کاهش یافته به صورت زیر است:

```

function [Xh, Zd] = Obsv_R(u, y, z)
    % System main Matrices
    B = [ 0.00306,      0;
          0, 0.0035;
          0, 0.0065;
          0.00714,      0];

    C = [ 1.0,      0, 0, 0;
          0, 1.0,      0, 0];

    % Observer Matrices
    F = [ -0.4,      0;
          0, -0.45];

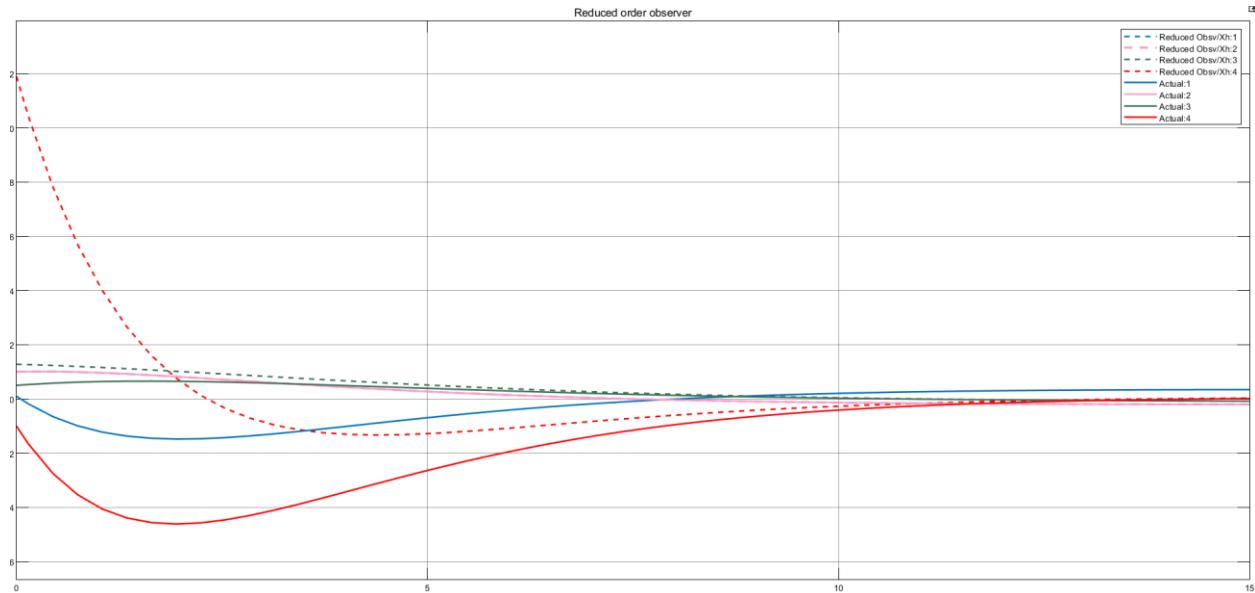
    T = [ 2.62,      0, -0.205,      0;
          0, -2.32,      0, 0.195];

    L = [ 1.0,      0;
          0, -1.0];

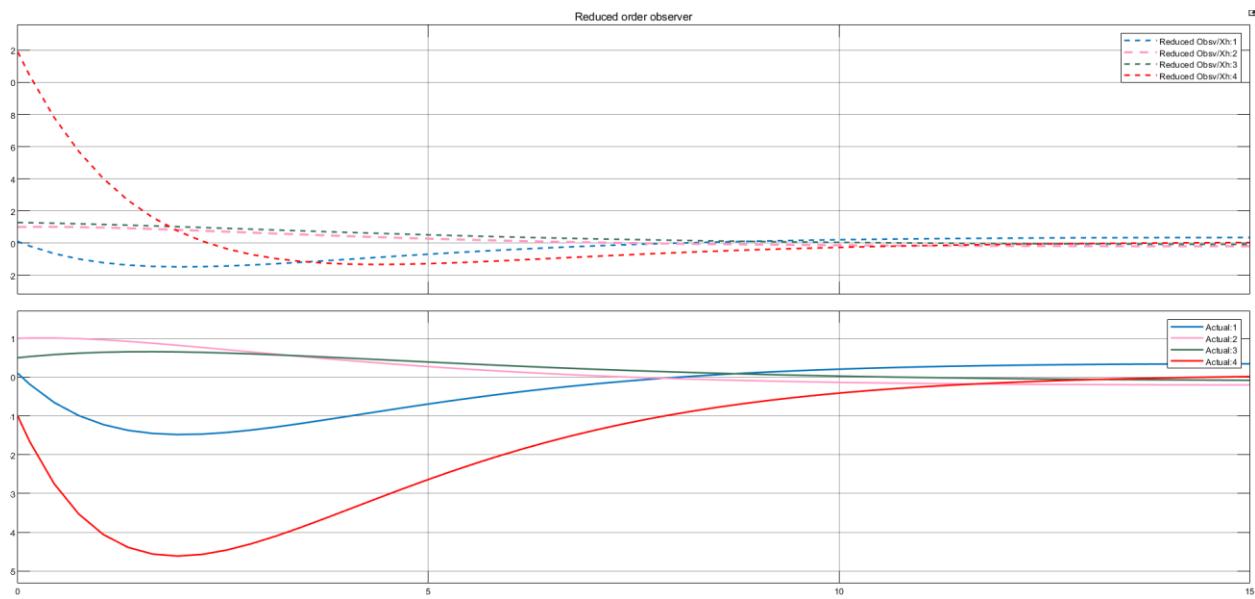
    Zd = F*z + T*B*u + L*y;
    Xh = ([C;T]^-1)*[y;z];
end

```

حالاتی سیستم و رویتگر به صورت زیر خواهد شد که نشان می دهد رویتگر توانسته به غیر از حالت چهارم (رنگ قرمز) به طور خوبی حالت های اصلی را دنبال کند. در حالت چهارم زیر پس از گذشت زمان موفق شده آن را تعقیب کند.



نمودار حالت ها به صورت جدا گانه نیز به شکل زیر می باشد:



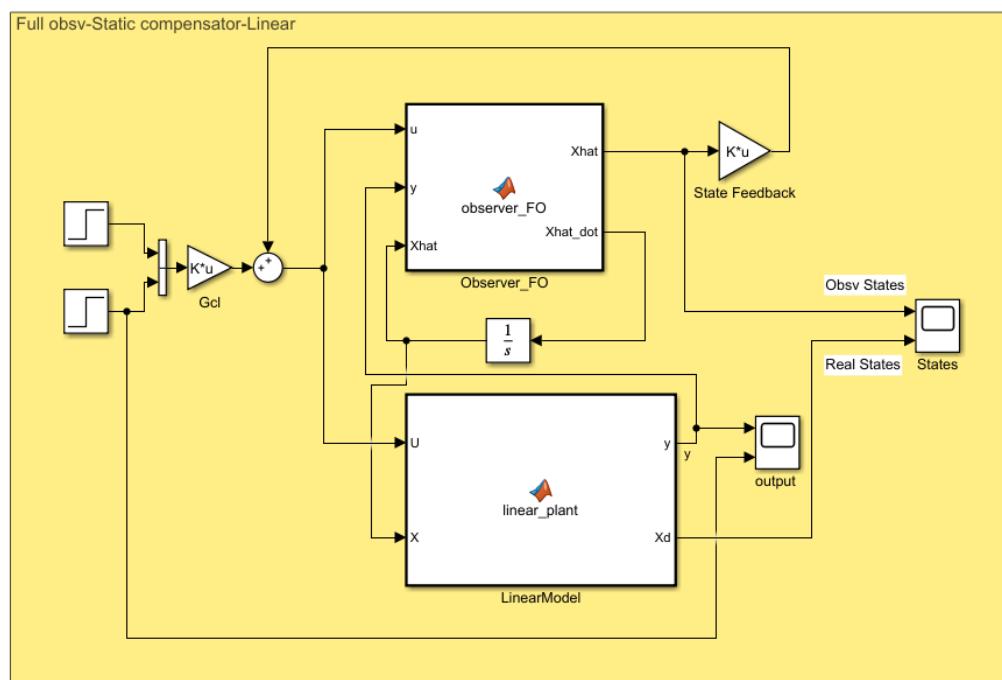
این نتایج نشان می دهد که رویتگر مرتبه کامل خطای کمتری به نسبت رویتگر مرتبه کاهش یافته دارد.

۲-۳ باز طراحی قسمت پنجم فاز دوم با رویتگر

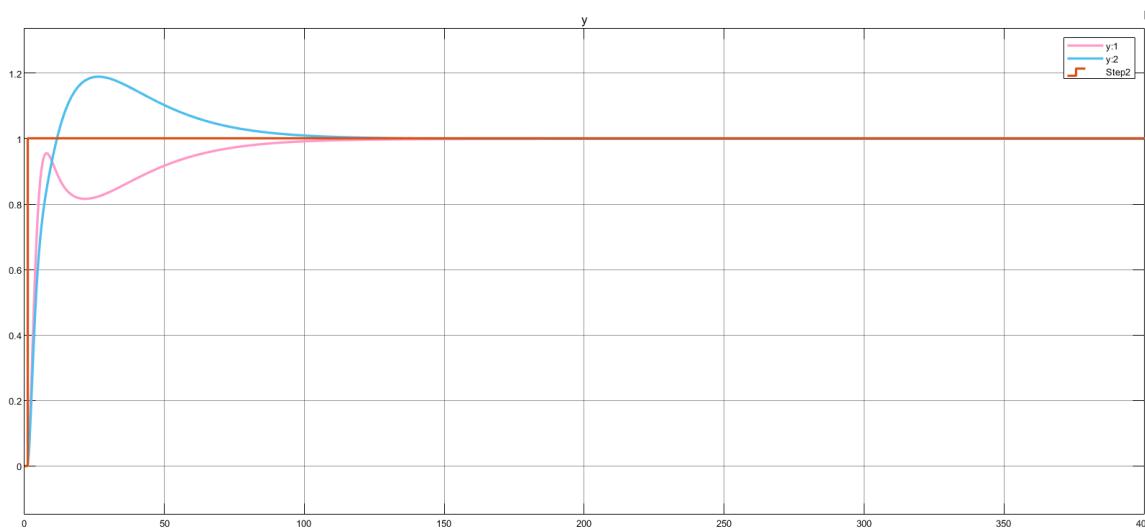
رونند کار مشابه فاز دوم است و بلوک رویتگرها و ماتریس K نیز مشابه قسمت قبل است. در هر بخش بلوک دیاگرام و نتایج شبیه سازی را مشاهده خواهیم کرد که خروجی می باشد ورودی پله را تعقیب کند.

۱-۲-۳ پیش جبرانساز استاتیکی خطی با رویتگر مرتبه کامل

بلوک دیاگرام:

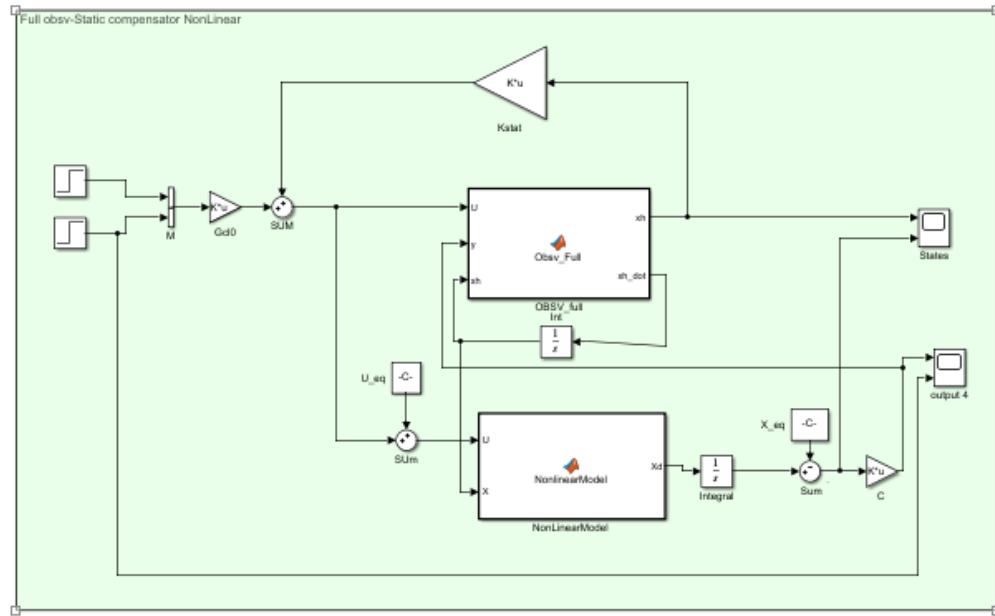


نمودار خروجی ها:

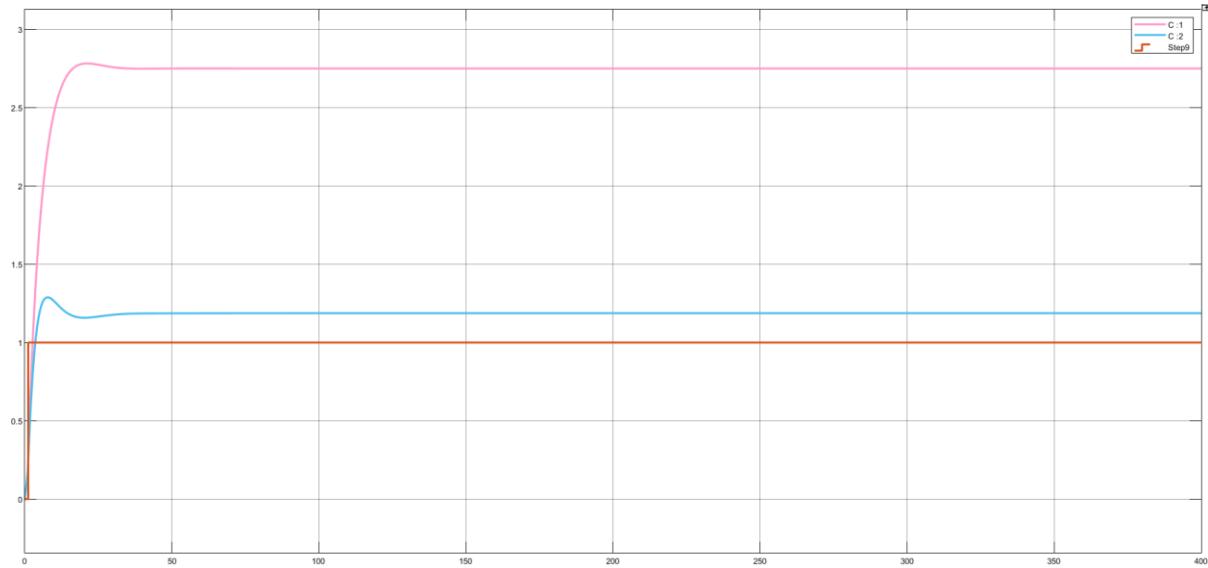


۲-۲-۳ پیش جبرانساز استاتیکی غیر خطی با رویتگر مرتبه کامل

بلوک دیاگرام:



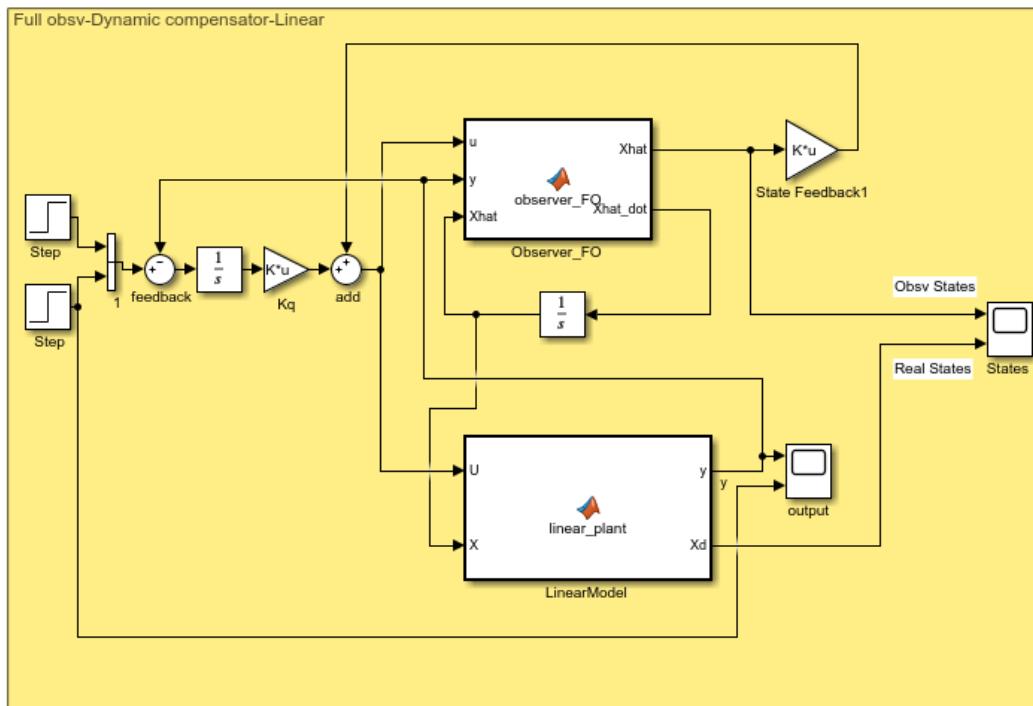
نمودار خروجی ها:



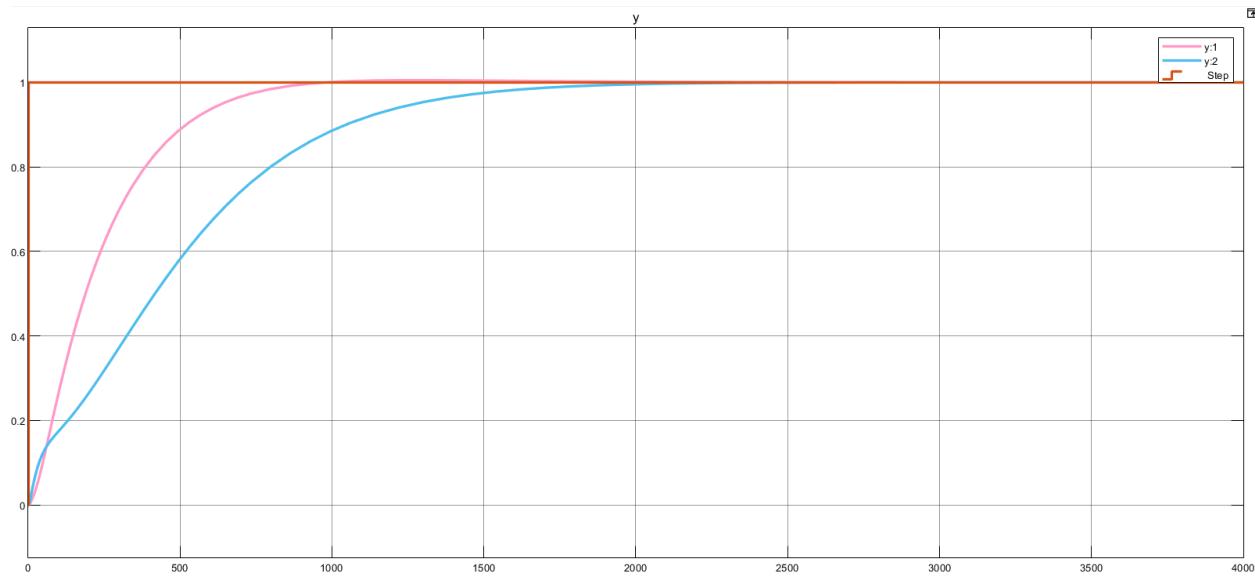
در این قسمت مشاهده می شود که به خاطر تاخیر و شرایط اولیه سیستم در لحظات ابتدایی نمی تواند به درستی تعقیب کند که باعث خطای حالت دائم در پیش جبرانساز استاتیکی شده و پس از پایدار شدن نمیتواند پله را تعقیب کند.

۳-۲-۳ پیش جبرانساز دینامیکی خطی با رویتگر مرتبه کامل

بلوک دیاگرام:

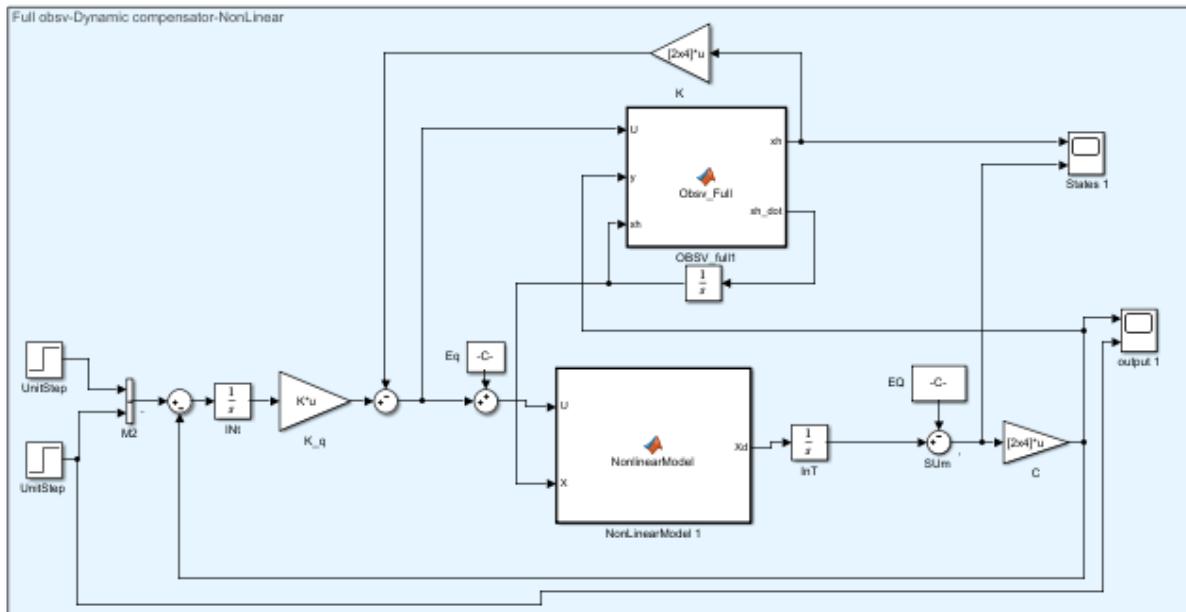


نمودار خروجی ها:

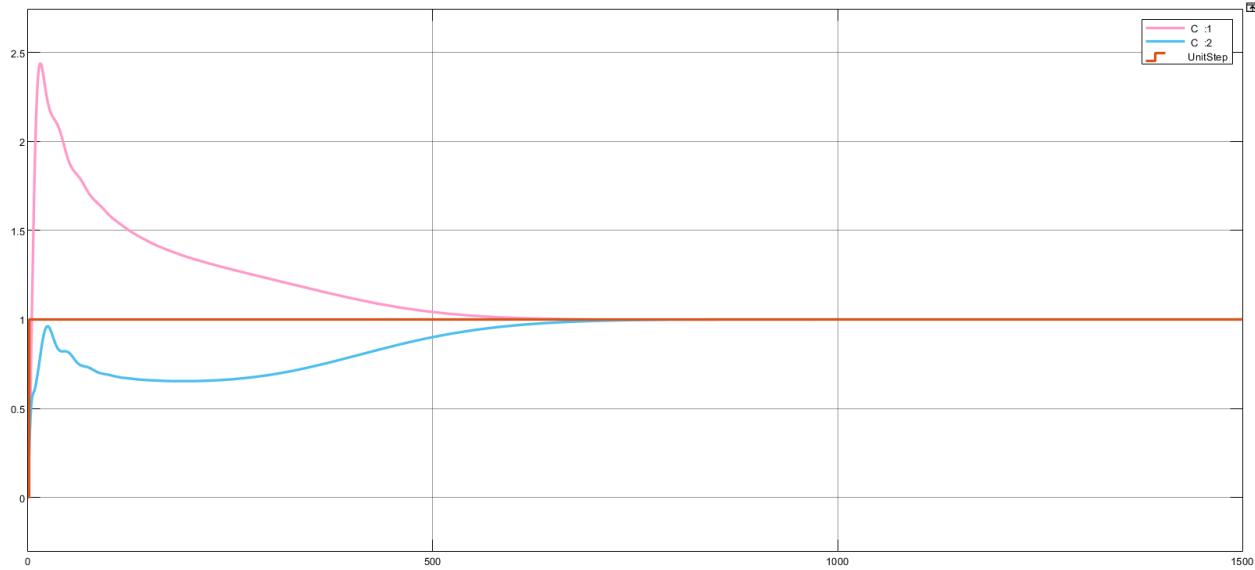


۴-۲-۳ پیش جبرانساز دینامیکی غیر خطی با رویتگر مرتبه کامل

بلوک دیاگرام:

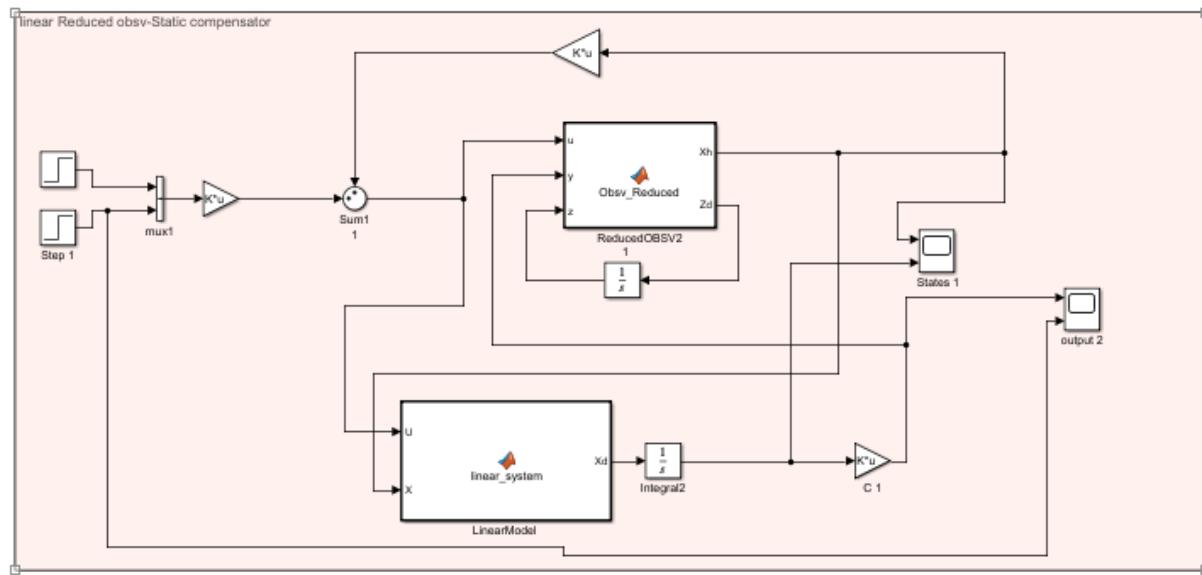


نمودار خروجی ها:

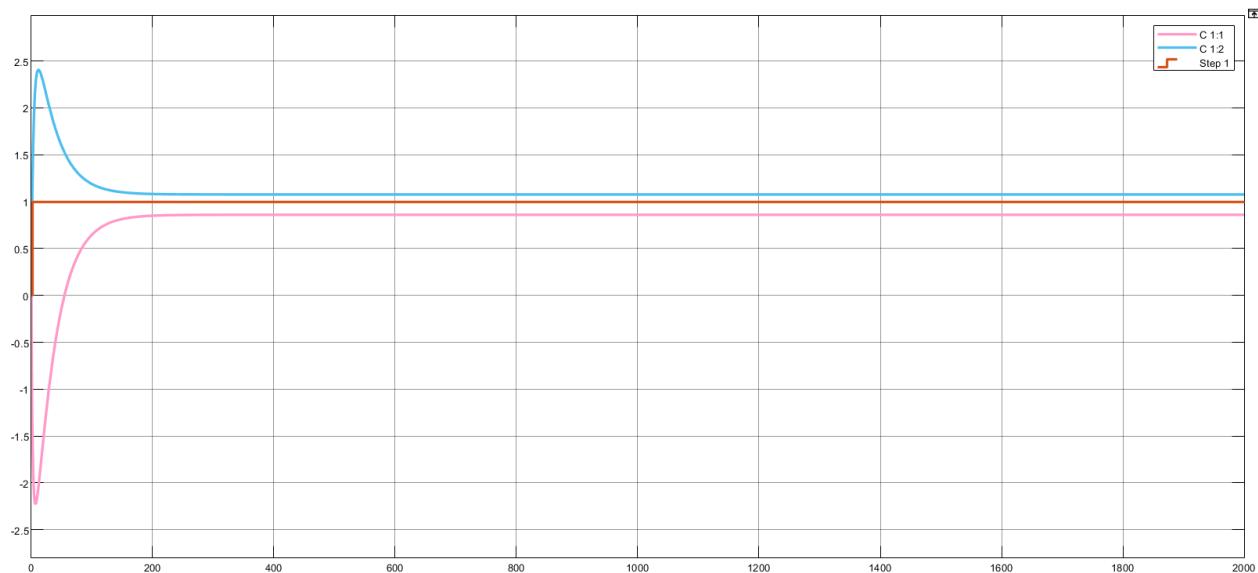


۵-۲-۳ پیش جبرانساز استاتیکی خطی با رویتگر مرتبه کاهش یافته

بلوک دیاگرام:

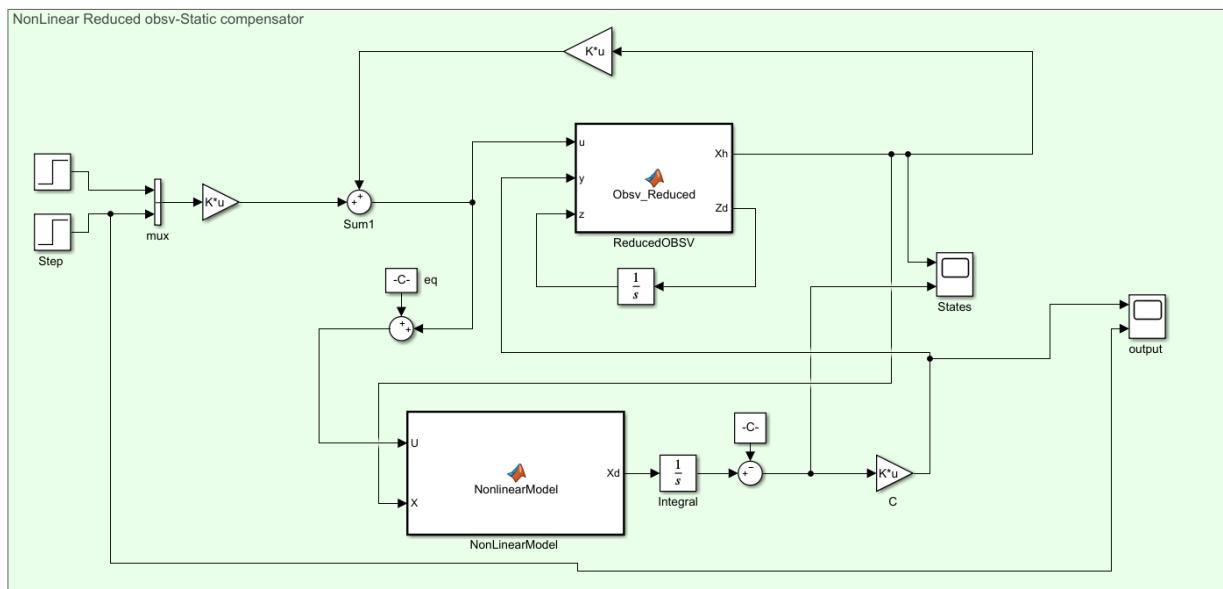


نمودار خروجی ها:

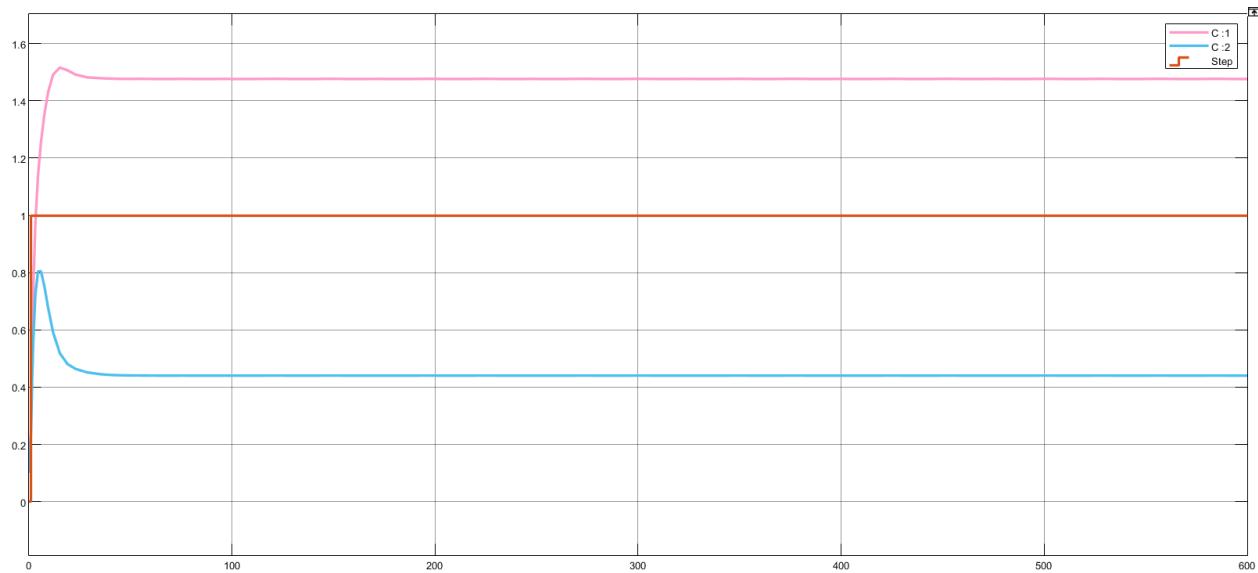


۳-۲-۶ پیش جبرانساز استاتیکی غیر خطی با رویتگر مرتبه کاهش یافته

بلوک دیاگرام:

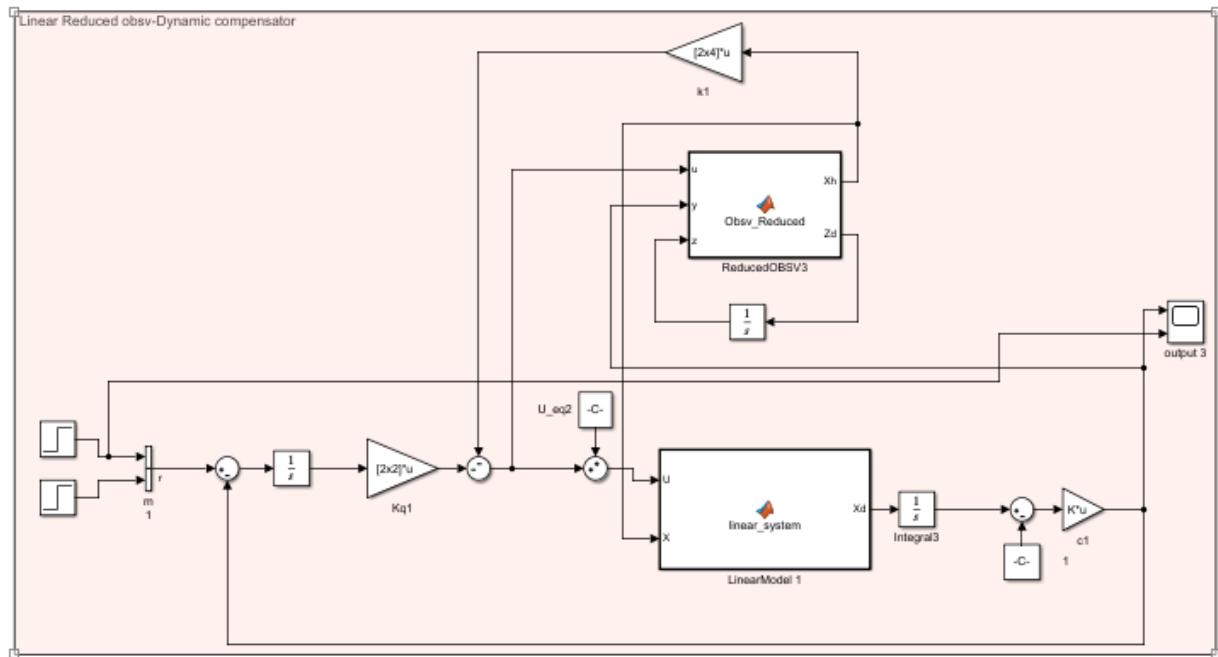


نمودار خروجی ها:

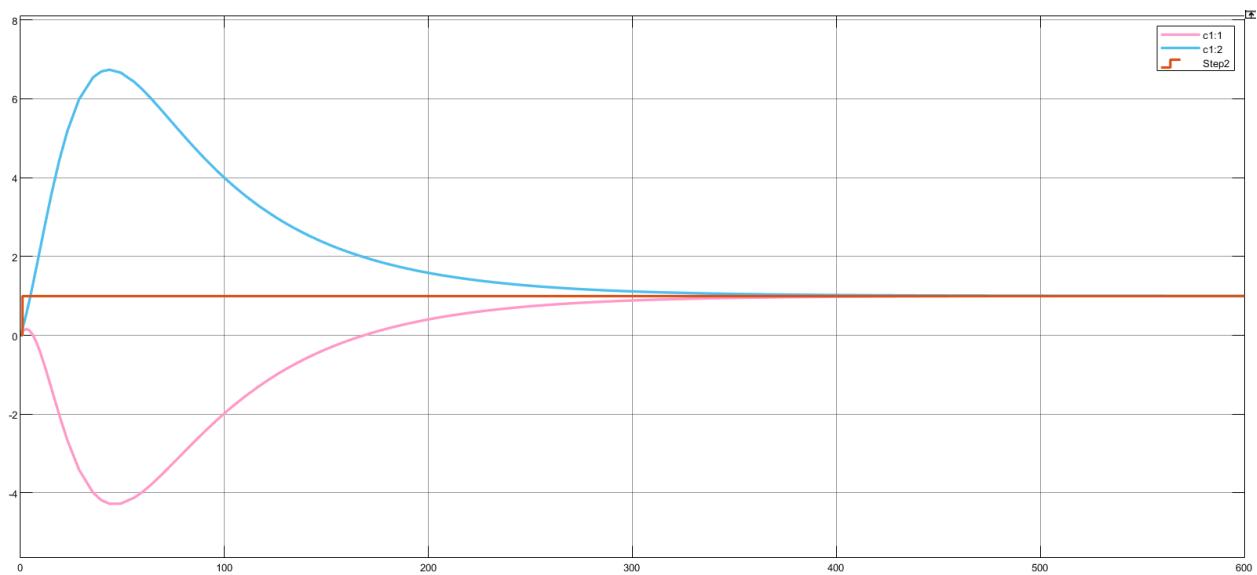


۷-۲-۳ پیش جبرانساز دینامیکی خطی با رویتگر مرتبه کاهش یافته

بلوک دیاگرام:

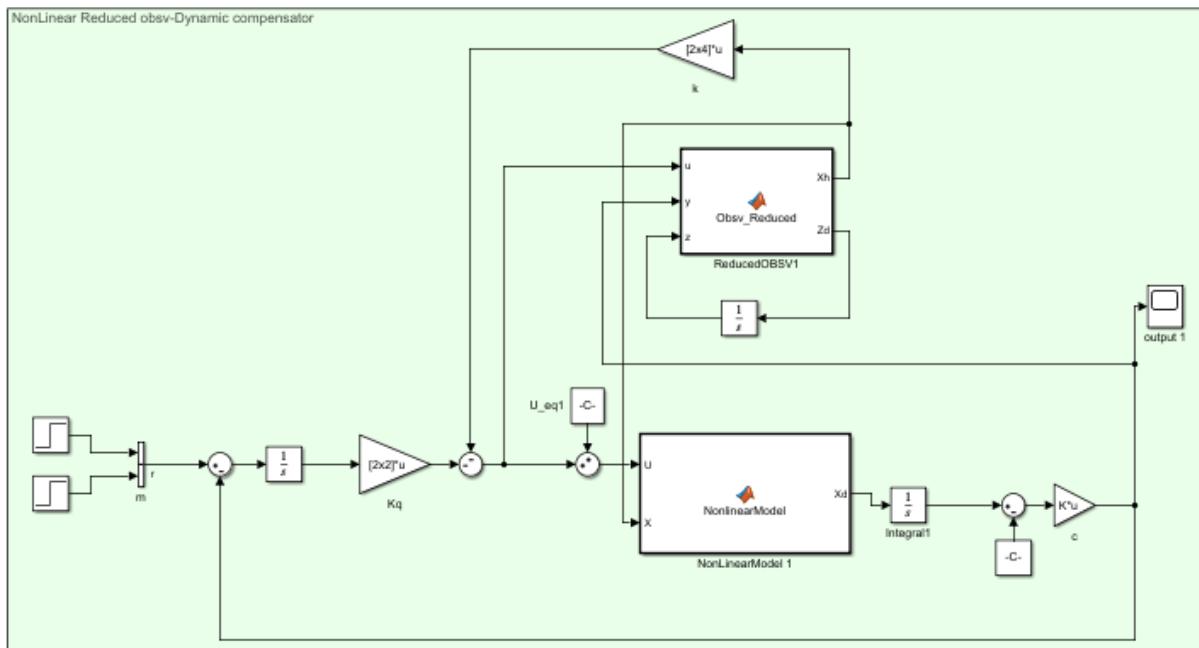


نمودار خروجی ها:

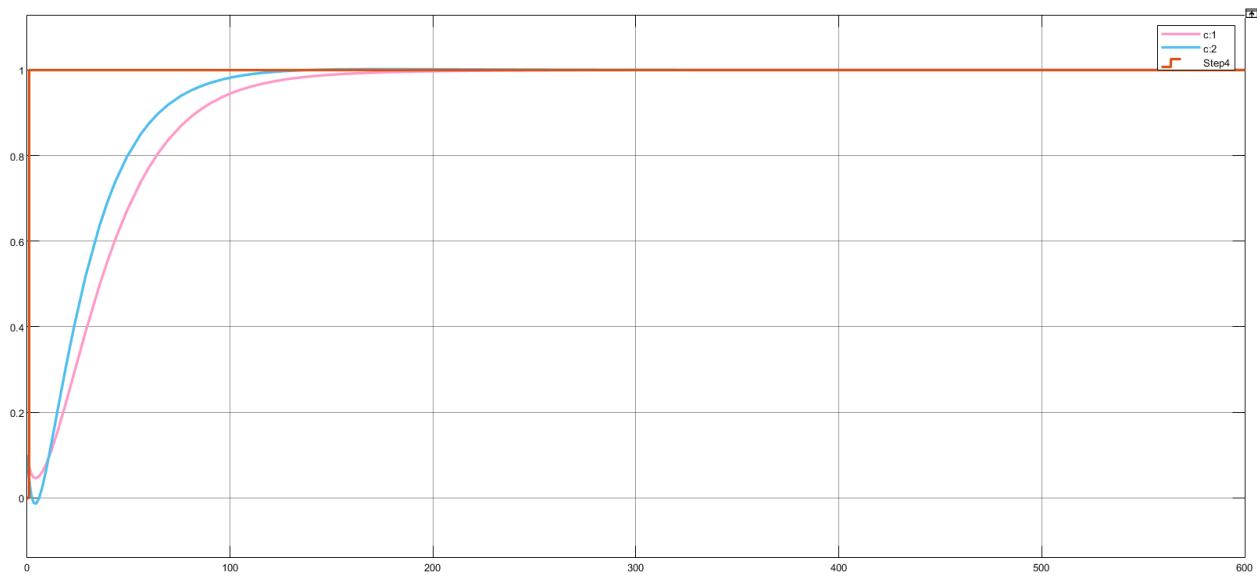


۸-۲-۳ پیش جبرانساز دینامیکی غیر خطی با رویتگر مرتبه کاهش یافته

بلوک دیاگرام:



نمودار خروجی ها:



۳-۳ بررسی پایداری

می دانیم تمام مقادیر ویژه ماتریس A سیستم اکیدا منفی هستند، بنابراین سیستم پایدار داخلی (پایدار مجانبی) است و چون مقدار ویژه صفر ندارد پس پایدار مرزی نیست. همچنین طبق قضیه چون سیستم پایدار داخلی است پس پایدار BIBO نیز می باشد.

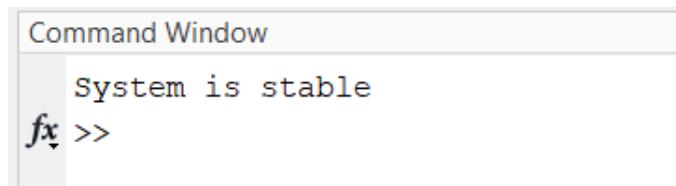
حال با استفاده از روش دوم لیاپانوف و با در نظر گرفتن ماتریس مثبت معین Q (ماتریس همانی) و با دستور lyap مطلب، ماتریس P را به دست آورده و در ادامه با محاسبه علامت کهاد های اصلی مقدم ماتریس P ، آن را تعیین علامت می کنیم که در صورتی که همه کهاد های اصلی مقدم مثبت باشند، سیستم پایدار خواهد بود. برای سیستم ما نیز نتیجه مطابق انتظار است و سیستم پایدار است.

```
% Stability of four tank system
clc;
clear;
close all;

Q = eye(4); % positive definite matrix
A = [ -0.01887, 0, 0.02899, 0;
       0, -0.01906, 0, 0.03489;
       0, 0, -0.02899, 0;
       0, 0, 0, -0.03489];

P = vpa(lyap(A, Q), 4);
D1 = det(P(1,1));
D2 = det(P(1:2,1:2));
D3 = det(P(1:3,1:3));
D4 = det(P);
if D1>0 && D2>0 && D3>0 && D4>0
    disp('System is stable')
elseif D1<0 && D2>0 && D3<0 && D4>0
    disp('System is not stable')
end
```

نتیجه:



Command Window

```
System is stable
fx >>
```

۴-۳ طراحی کنترل کننده بهینه

در سیستم‌های خطی، می‌توانیم با بیان کمک‌حالات به عنوان تابعی خطی از حالات، یک نمایش مستقل از حالت برای آن بنویسیم. همچنین، در سیستم‌های خطی، تابع هزینه یک تابع درجه دوم از حالت و کنترل است. این فرم به عنوان کنترل یا رگولاتور مرتبه دوم خطی (Linear Quadratic Regulator) یا LQR نامیده می‌شود. در LQR، قانون کنترل را به گونه‌ای محاسبه می‌کنیم که تابع هزینه درجه دوم زیر کمینه شود:

$$J(u) = \frac{1}{2} X_f^T S X_f + \frac{1}{2} \int_{t=0}^{t_f} (X^T Q X + u^T R u) dt.$$

که در آن، Q یک ماتریس معین نیمه‌مثبت است و هزینه یا ارزش انحراف متغیرها از صفر را نشان می‌دهد، R یک ماتریس مثبت معین است و هزینه کنترل را نشان می‌دهد، و S یک ماتریس معین مثبت است که خطا از مقدار مطلوب نهایی (در اینجا، صفر) را جرمیه می‌کند. ماتریس‌های R و Q به صورت یک ماتریس قطری انتخاب می‌شوند. مقادیر بزرگ Q خطای مسیر و مقادیر بزرگ R کنترل را جرمیه می‌کنند. بنابراین، استفاده از این فرمول‌بندی، کار طراح کنترل را برای تشخیص قطب‌های سیستم آسان کرده و تمرکز او را به تنظیم وزن‌های مربوط به هزینه کنترل و خطای ردیابی محدود می‌کند. یک فرایند طراحی کنترل معمولی با تغییر وزن‌های خطای ردیابی و هزینه یا تلاش کنترلی سر و کار دارد تا پاسخ مناسب به دست آید.

۱. برای $\text{norm}(Q) \gg \text{norm}(R)$ ، تلاش کنترلی با جرمیه سنگینی رو به رو خواهد بود، و کنترل کننده حاصل از این انتخاب دیرتر خطا را صفر خواهد کرد.

۲. برای $\text{norm}(Q) \gg \text{norm}(R)$ ، هزینه خطای ردیابی بالا بوده و خطای سریعاً به صفر خواهد رسید.

۳. می‌توان R را به گونه‌ای انتخاب کرد که درایه‌های قطری آن مقادیر متفاوتی داشته باشند. در این حالت، کنترل متناظر با مقدار قطری بزرگ‌تر، کمتر استفاده خواهد شد.

۴. می‌توان Q را به گونه‌ای انتخاب کرد که درایه‌های قطری آن مقادیر متفاوتی داشته باشند. در این حالت، حالت‌های متناظر با مقادیر قطری بزرگ‌تر، سریع‌تر به صفر میل می‌کنند.

طبق نکات بالا برای اینکه حالت‌های خروجی سرعت بالاتری داشته باشند ماتریس Q را به صورت زیر تعریف می‌کنیم و ماتریس R نیز اندازه کمتری نسبت به Q دارد تا خطای سرعت بیشتری به صفر برسد.

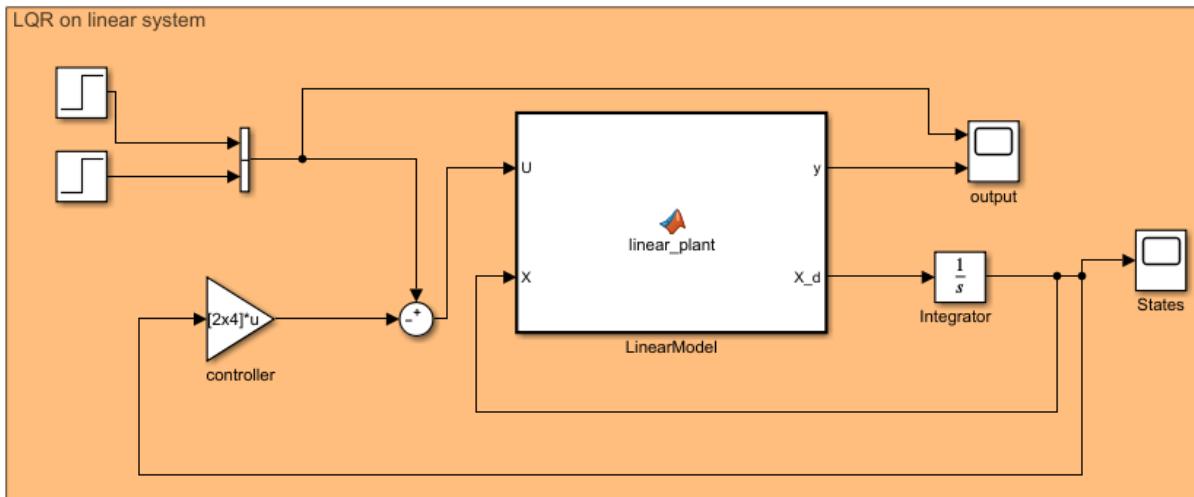
مقادیر عددی این ماتریس‌ها بر مبنای اعداد مقاله مرجع تنظیم شده‌اند که در آنجا با مقایسه انواع مختلف ماتریس‌های Q و R بهینه ترین آن‌ها را انتخاب کرده است.

```

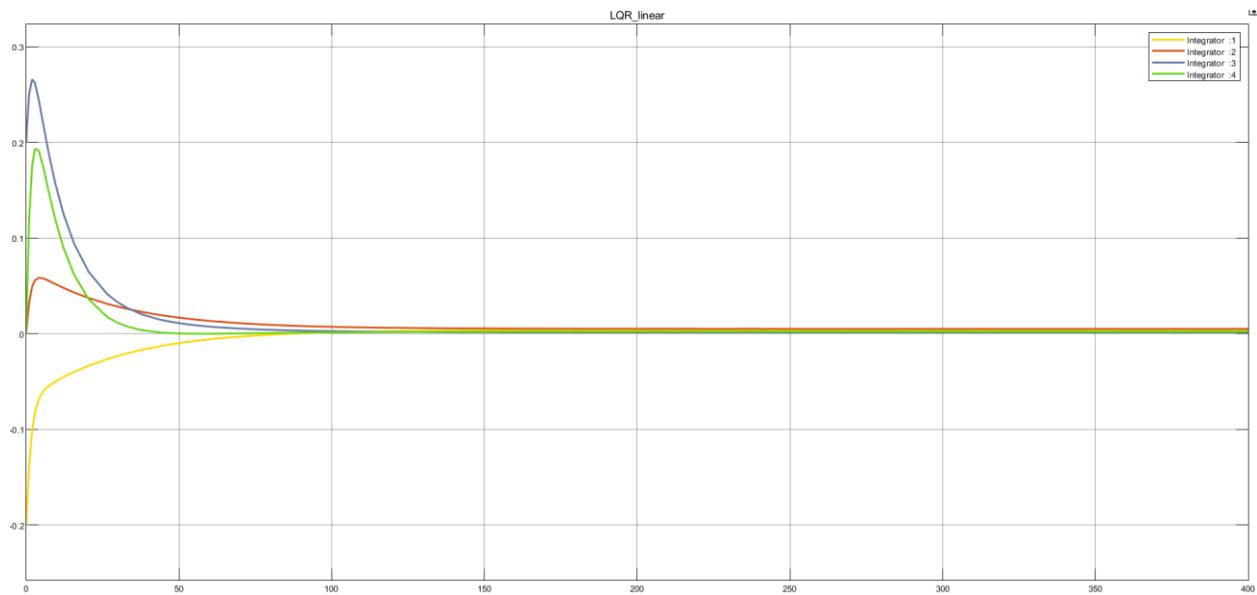
%% LQR
Q = diag([500 500 0 0]);
R = diag([0.01 0.01]);
[K_lqr,S_lqr,e_lqr] = lqr(A,B,Q,R);

```

حال مشابه روند شبیه سازی فیدبک حالت در فاز قبل، با استفاده از K_{lqr} به دست آمده سیستم را در حالت خطی شبیه سازی می کنیم.

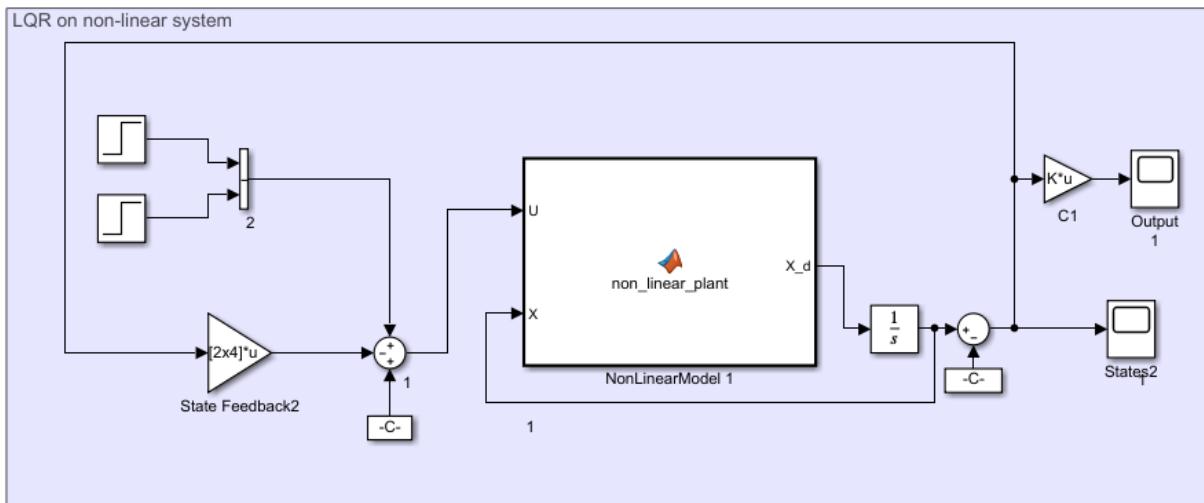


که حالت های سیستم به صورت زیر خواهند شد که به طور مطلوب و به سرعت میرا و پایدار می شوند:

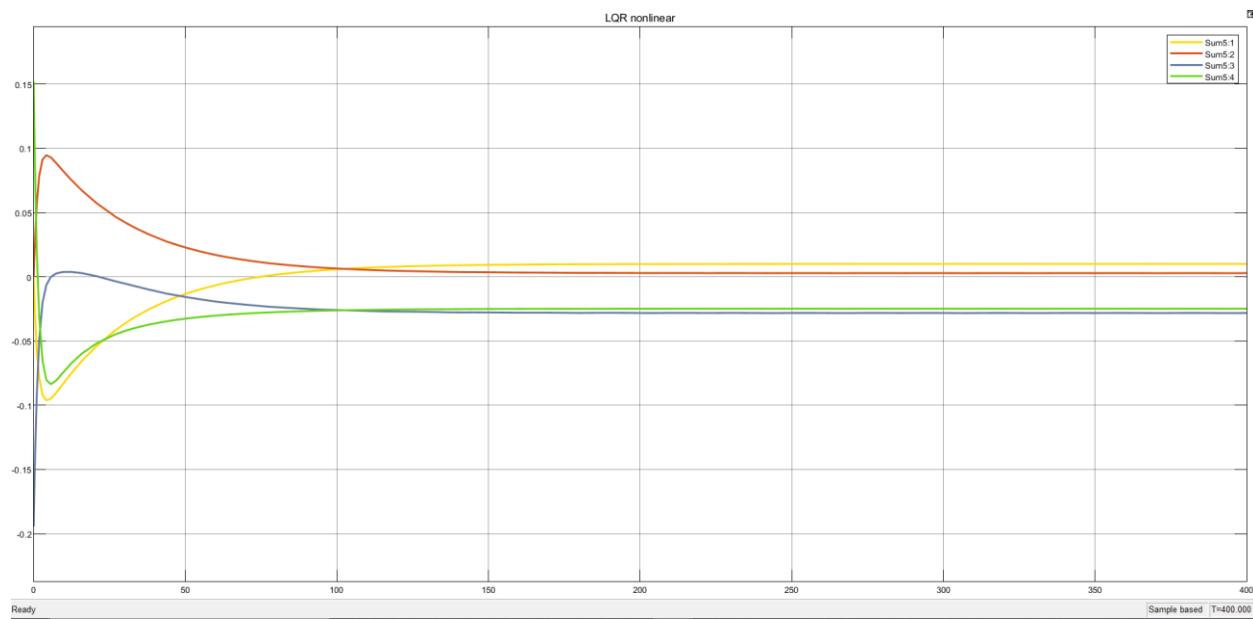


۵-۳ پیاده سازی کنترل کننده بهینه

در این قسمت مشابه قسمت قبل سیستم غیر خطی را با K_{lqr} به دست آمده و شرایط اولیه مشابه سیستم خطی شبیه سازی می کنیم.



نمودار حالت های سیستم به صورت زیر خواهد شد که بسیار شبیه به حالت خطی است و همه حالت ها به سرعت پایدار و میرا شده و به نتیجه مطلوب می رسیم.



جمع بندی

سیستم مورد بررسی ما سیستم چهار تانک است و مسئله اصلی آن کنترل سطح مایعات و تنظیم جریان بین مخازن است. مدل خطی این سیستم درجه ۶ است و دارای دو ورودی و دو خروجی میباشد. تمام مقادیر ویژه سیستم سمت چپ محور هستند و بنابراین سیستم دارای پایداری داخلی و پایداری BIBO می باشد. این سیستم هم رویت پذیر و هم کنترل پذیر است و هیچ مود رویت ناپذیر یا کنترل ناپذیر ندارد.

پس از بررسی نکات اولیه سیستم یک کنترل فیدبک حالت و در ادامه پیش جبرانساز های استاتیکی و دینامیکی طراحی کرده و سیستم خطی شده و غیر خطی را با این کنترل کننده شبیه سازی و مقایسه می کنیم. مشاهده می شود که سیستم غیر خطی حول نقطه تعادل مشابه سیستم خطی شده عمل می کند. همچنین پیش جبرانساز دینامیکی برخلاف پیش جبرانساز استاتیکی در برابر اغتشاش بسیار مقاوم تر است. همچنین در بخش ردیابی سیگنال سینوسی دریافتیم که با اضافه کردن ترم انگرال گیر در کنترلر می توان سیگنال سینوسی را به خوبی ردیابی کرد.

در بخش رویتگر های مرتبه کامل و مرتبه کاهش یافته دریافتیم که باید قطب های رویتگر را از قطب های فیدبک حالت دورتر قرار دهیم تا قسمت تخمین حالت ها سریعتر انجام شوند و با مقایسه نمودار های این بخش دریافتیم که رویتگر مرتبه کامل خطای کمتری به نسبت رویتگر مرتبه کاهش یافته دارد.

همچنین توانستیم با استفاده از رویتگر ها و پیش جبرانساز های استاتیکی و دینامیکی سیگنال ورودی مرجع پله را تعقیب کنیم. در آخر نیز با یک کنترل کننده بهینه (LQR) سرعت سیستم را افزایش داده و نتیجه آن را در سیستم های خطی شده و غیر خطی شبیه سازی و مقایسه می کنیم.

منابع و مراجع

[۱] Wael A. Altabey, Model Optimal Control of the Four Tank System, *International Journal of Systems Science and Applied Mathematics*. Volume ۱, Issue ۴, November ۲۰۱۶ , pp. ۳۰-۴۱. doi: ۱۰.۱۱۶۴۸/j.ijssam.۲۰۱۶۰۱۰۴,۱۱

[۲] J. Jayaprakash and M. E. Hari Kumar, "State variable analysis of four tank system," ۲۰۱۴ International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), ۲۰۱۴, pp. ۱-۸, doi: ۱۰.۱۱۰۹/ICGCCEE.۲۰۱۴,۶۹۲۲۳۴۱.