

دانشکده مهندسی برق

یادگیری تقویتی در کنترل

تمرین هفتم: یادگیری تقویتی در کنترل بهینه و مقاوم

استاد: دکتر سعید شمقدری

دانشجو: سیده ستاره خسروی

زمستان 1403

چکیده

در تمرین سری هفتم یادگیری تقویتی در کنترل با 3 سوال از مبحث کنترل بهینه، یادگیری تقویتی انتگرالی و کنترل مقاوم مواجه هستیم، که در هر فصل به سوال و یا سوالات مطرح شده پاسخ داده شده است.

واژه‌های کلیدی: یادگیری تقویتی، کنترل بهینه، کنترل مقاوم

# فهرست مطالب

عنوان صفحه

[فهرست مطالب ‌ب](#_Toc186765301)

[فهرست تصاویر و نمودارها ‌ج](#_Toc186765302)

[فصل 1:کنترل بهینه و مقاوم 1](#_Toc186765303)

[1‌.1‌ مقدمه 1](#_Toc186765304)

[1‌.2‌ سوال اول 1](#_Toc186765305)

[1‌.3‌ سوال دوم 7](#_Toc186765306)

[1‌.4‌ سوال سوم 15](#_Toc186765307)

# فهرست تصاویر و نمودارها

عنوان صفحه

[شکل 1: رابطه ارزیابی پالیسی (اسلایدهای درس) 1](#_Toc186765310)

[شکل 2: بهبود پالیسی (اسلایدهای درس) 3](#_Toc186765311)

[شکل 3: مقدار K بهینه 5](#_Toc186765312)

[شکل 4: مقادیر K و P در 4 تکرار اول و همگرایی 5](#_Toc186765313)

[شکل 5: نمودار همگرایی مقادیر K و اندازه تغییرات K و P 6](#_Toc186765314)

[شکل 6: متغیرهای حالت سیستم در 10 ثانیه ابتدایی 6](#_Toc186765315)

[شکل 7: مقادیر P براساس دستور CARE 8](#_Toc186765316)

[شکل 8: مقادیر P و K1 و K2 12](#_Toc186765317)

[شکل 9: پاسخ حالت‌ها 13](#_Toc186765318)

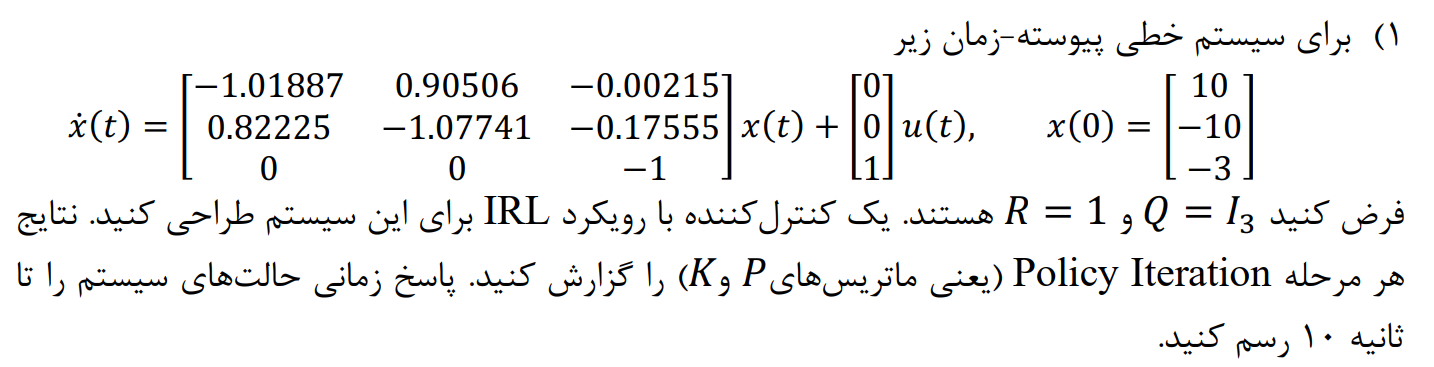
[شکل 10: نمودار همگرایی rd 14](#_Toc186765319)

# کنترل بهینه و مقاوم

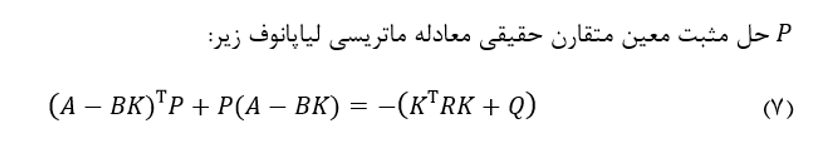
## مقدمه

در این فصل به 3 سوال مربوط به این فصل پاسخ داده می‌شود.

## سوال اول

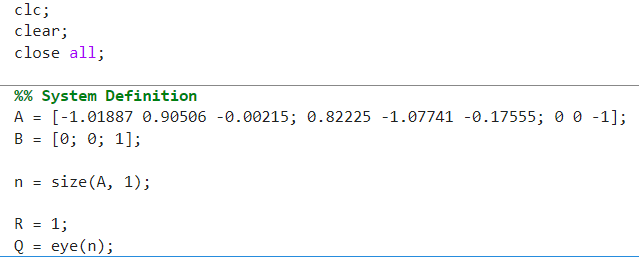
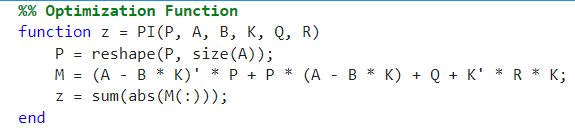
صورت سوال:

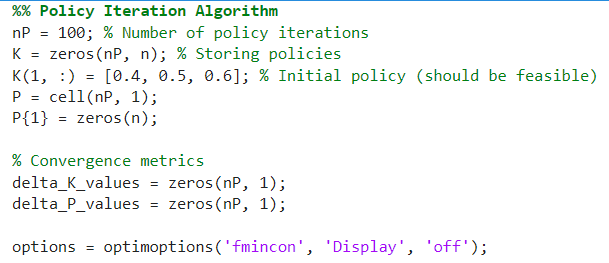
پاسخ:

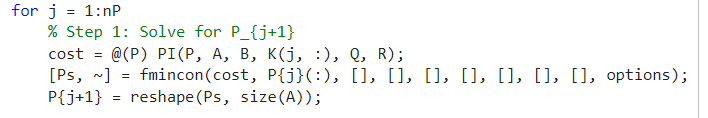
در این بخش برای حل سوال، ابتدا لازم است مانند تمرین قبلی تابعی برای بدست آوردن مقدار P در گام ارزیابی پالیسی ایجاد کنیم. به همین منظور از رابطه زیر که مربوط به IRL است کمک می‌گیریم.

شکل 1: رابطه ارزیابی پالیسی (اسلایدهای درس)

به همین ترتیب، کد زیر که مربوط به تایع این بخش است نوشته می‌شود که درون حلقه آموزش با حل آن به وسیله fmincon مقدار P بدست می‌آید.

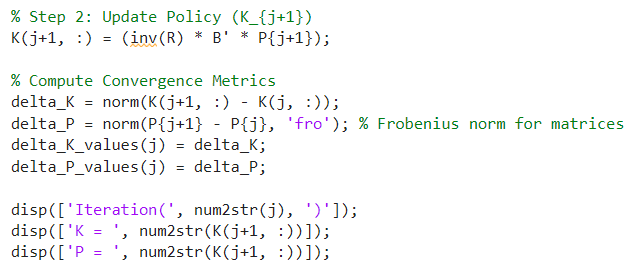
سپس دینامیک سیستم را تعریف می‌کنیم.

در ادامه تنظیمات فرایند آموزش را انجام می‌دهیم، تعداد دفعاتی که حلقه آموزش قرار است اجرا شود را مشخص می‌کنیم. متغیرهایی برای ذخیره مقادیر P و K تعریف می‌کنیم و یک K ابتدایی که feasible باشد نیز انتخاب می‌کنیم. تنظیمات حل fmincon نیز در این بخش مشخص می‌شود.

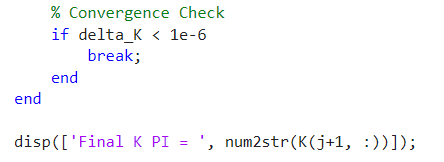
حال حلقه اصلی برنامه را می‌نویسیم.

در گام اول مطابق کد بالا با استفاده از تابعی که نوشته بودیم و دستور fmincon مقدار P را بدست می‌آوریم. سپس بر اساس رابطه‌ی آمده درون اسلایدهای درس:

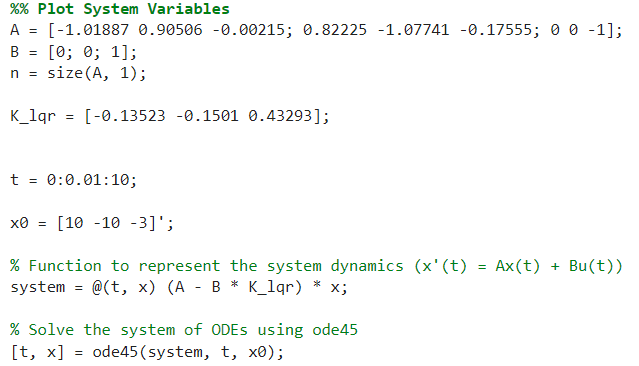
شکل 2: بهبود پالیسی (اسلایدهای درس)

بهبود پالیسی را درون کد انجام می‌دهیم، مقادیر خطا را محاسبه کرده و ذخیره می‌کنیم و سپس در هر تکرار مقادیر P و K را نمایش می‌دهیم.

در ادامه اگر خطای K، کمتر از 1e-6 باشد، الگوریتم متوقف می‌گردد و K نهایی نمایش داده می‌شود.



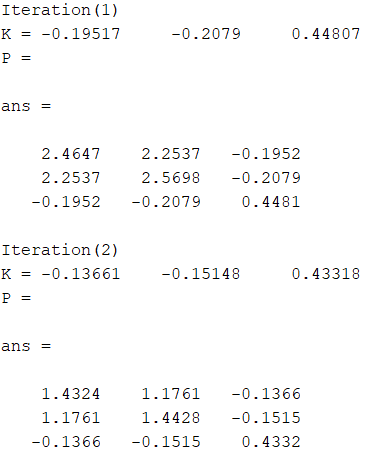
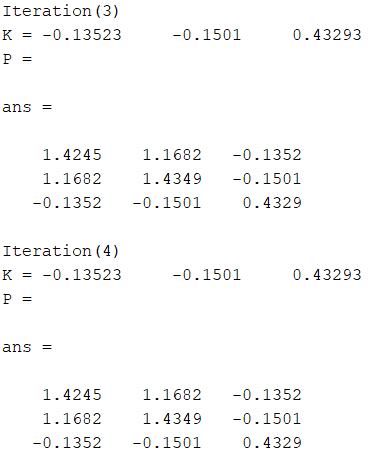
در ادامه‌ی کد دو تابع برای رسم Kها، Pها و خطای آن‌ها نوشته شده است که توضیح آن‌ها صرف نظر می‌گردد.

در کد دوم مربوط به این سوال نیز دینامیک دیفرانسیلی سیستم در حضور K بهینه که در کد اول بدست آمد، تعریف می‌شود و با استفاده از ode45 معادله آن حل شده و متغیرهای حالت را رسم می‌کنیم. در اینجا شرایط اولیه سیستم را مانند تمرین قبلی لحاظ کردیم.

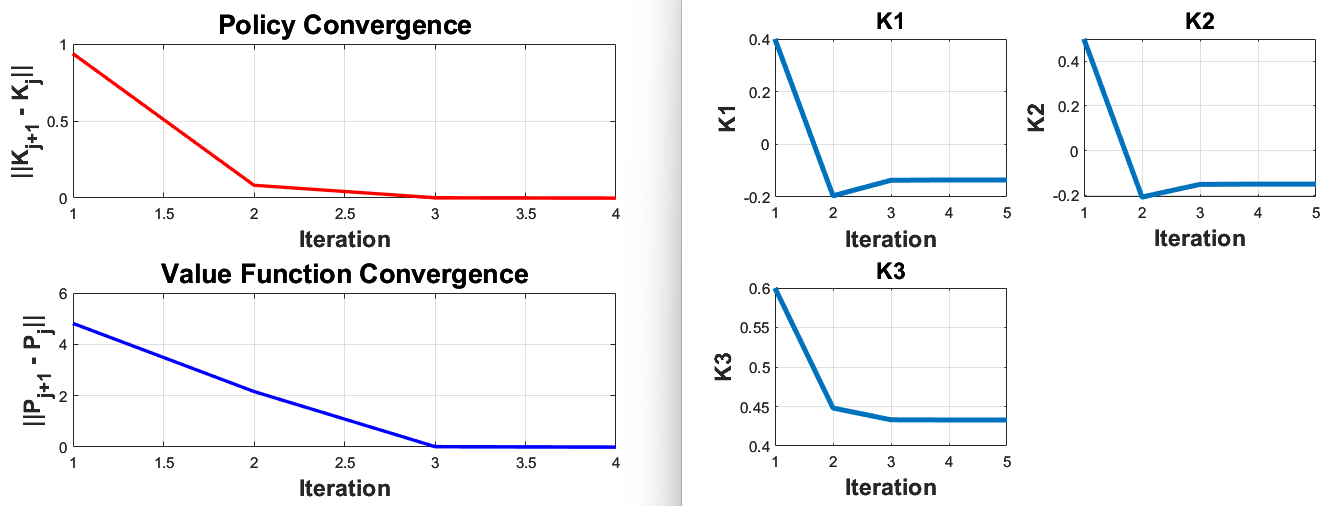
حال به بررسی خروجی‌های سیستم می‌پردازیم.

با اجرای کد اول مقدار K بهینه نهایی پس از 4 تکرار به صورت زیر بدست می‌آید.

شکل 3: مقدار K بهینه

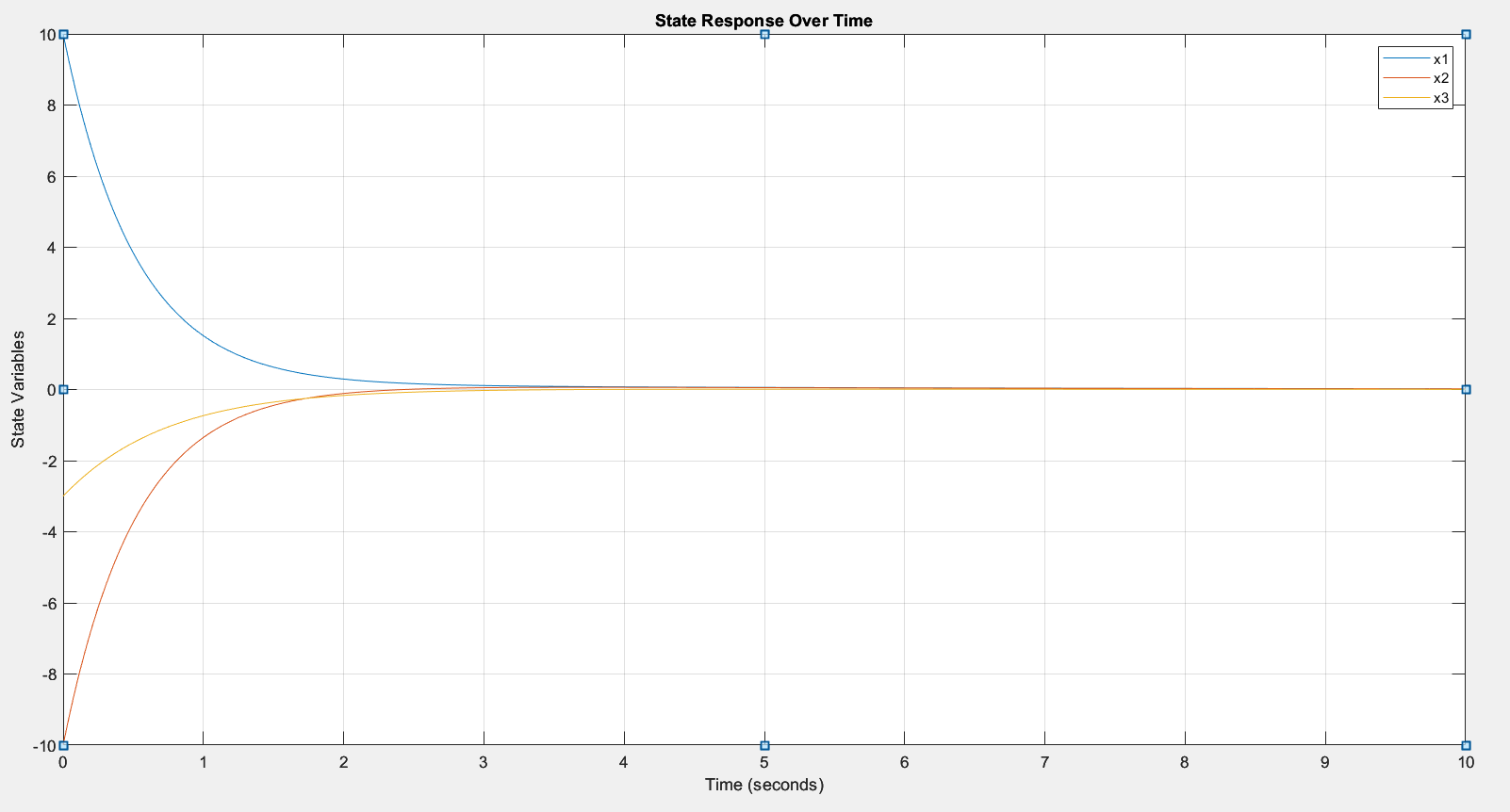
در هر تکرار نیز مقادیر K و P به صورت زیر بدست می‌آید.

شکل 4: مقادیر K و P در 4 تکرار اول و همگرایی



شکل 5: نمودار همگرایی مقادیر K و اندازه تغییرات K و P

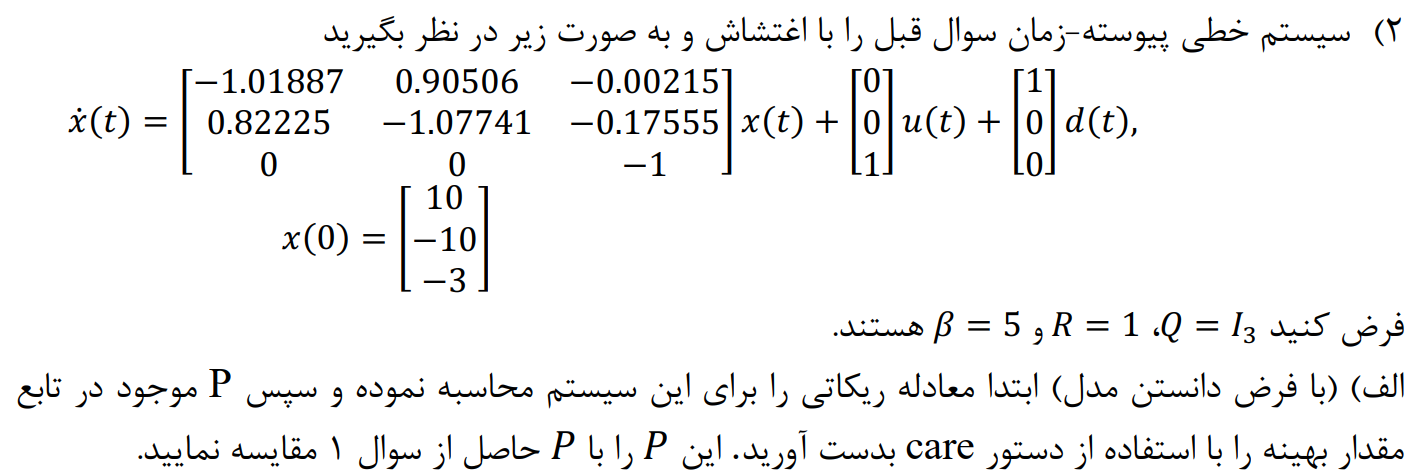
در شکل 5 نیز، مقادیر K در هر تکرار رسم شده و تغییرات اندازه P و K در مراحل آموزش رسم شده، که مانند آنچیزی که مقادیر عددی نشان می‌دهد همگرایی الگوریتم پس از 4 تکرار کاملا مشهود است.

در ادامه متغیرهای حالت سیستم رسم گردیده است که به صورت شکل 6 است.

شکل 6: متغیرهای حالت سیستم در 10 ثانیه ابتدایی

همانطور که مشاهده می‌شود متغیرهای حالت سیستم در 3 ثانیه ابتدایی به مقدار صفر میل می‌کنند.

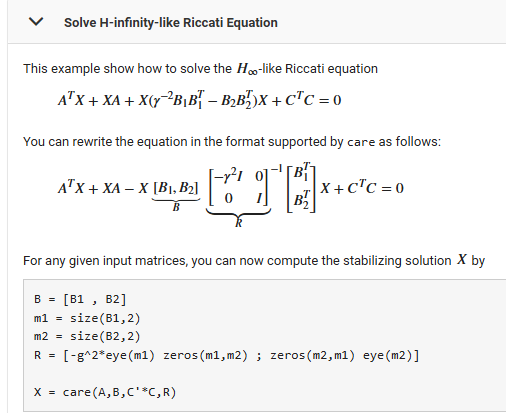
## سوال دوم

صورت سوال قسمت الف:

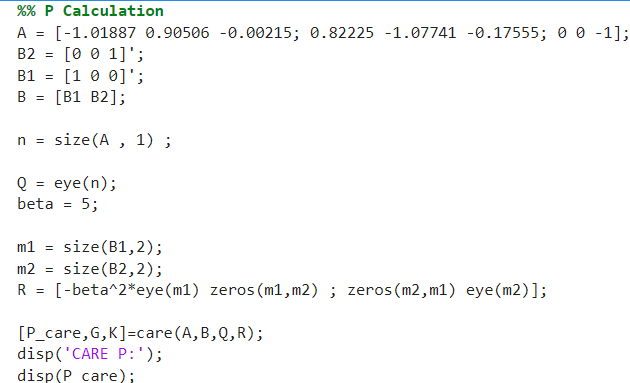
پاسخ:

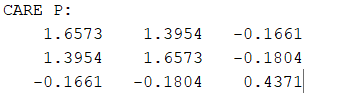
بر اساس مقاله پیوست برای سیستم مشابه زیر:

معادله GARE به صورت زیر می‌گردد:

که در آن S مقدار کرنل بهینه بزرگتر مساوی صفر و H هم نماینده همان Q است.

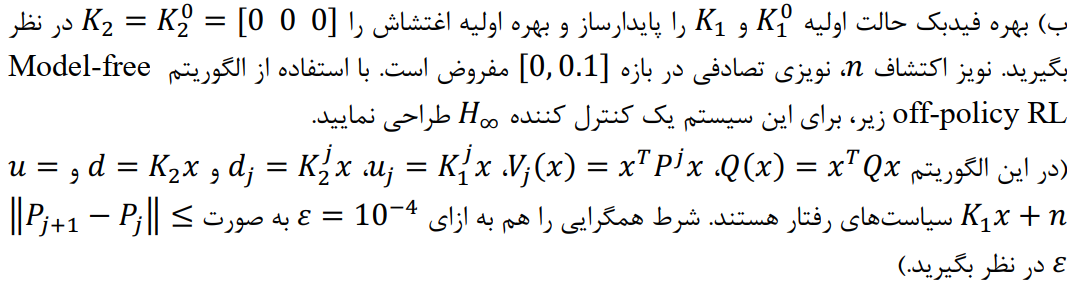
بر اساس توضیحات سایت متلب، که در تصویر روبرو موجود است، کد مناسب را می‌نویسیم.



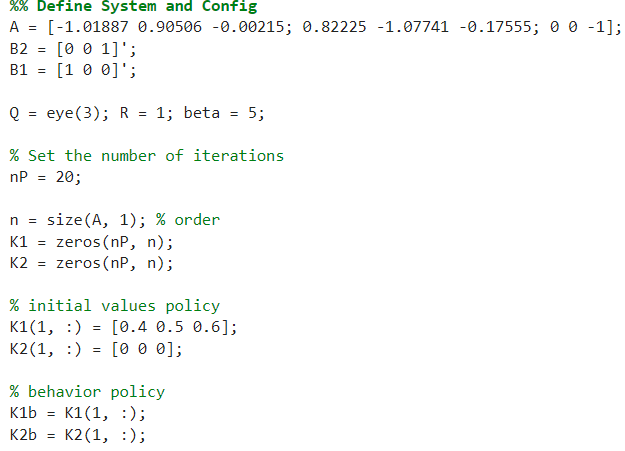
با استفاده از کد بالا نتیجه به صورت زیر خواهد بود:

شکل 7: مقادیر P براساس دستور CARE

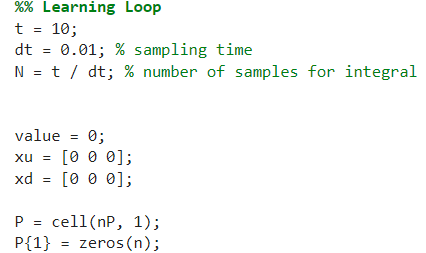
که این مقادیر بدست آمده به مقادیر بدست آمده در سوال اول نزدیک است.

**صورت سوال قسمت ب:**

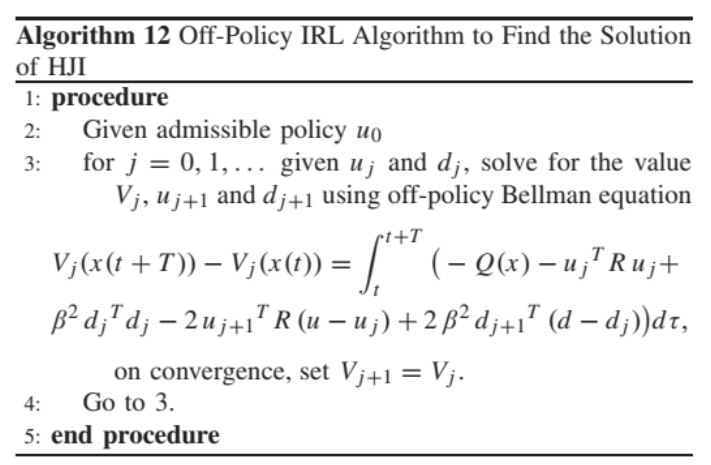
**پاسخ:**

در این قسمت ابتدا تنظیمات اولیه‌ی آموزش را تعیین می‌کنیم. (تعریف مقادیر بهره، دینامیک سیستم و ...)

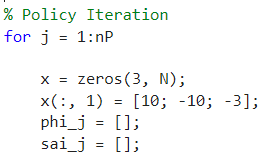
در ادامه تنظیمات حلقه آموزش را برای بحث انتگرال‌گیری و محاسبات P تنظیم می‌کنیم.

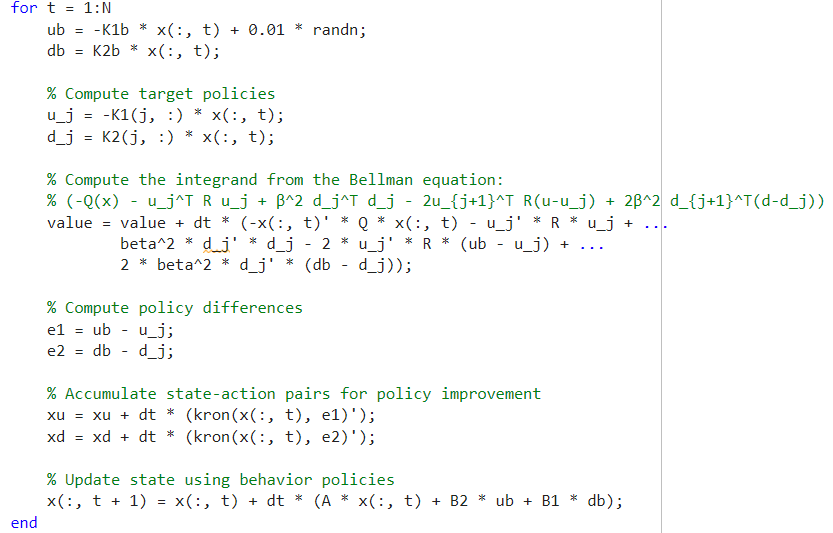
در اینجا مقدار اولیه ارزش، مقدار اولیه P که البته به صورت متغیر سلولی نیز تعریف شده است برابر با صفر یا ماتریس صفر لحاظ می‌شود. (مانند تمرین قبلی و سوال اول تمرین کنونی)

سپس حلقه‌ی اصلی برنامه را می‌نویسیم.

الگوریتم صورت سوال مفروض است:

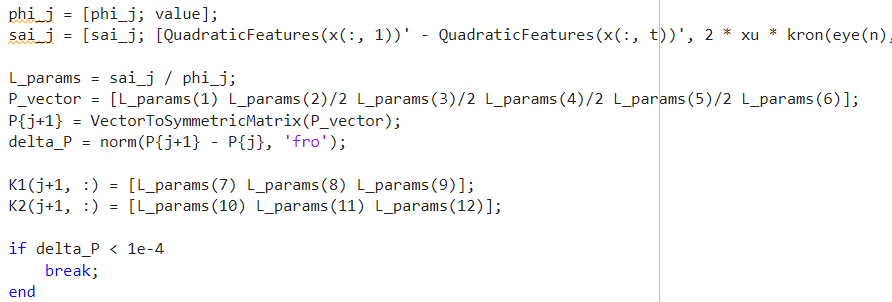
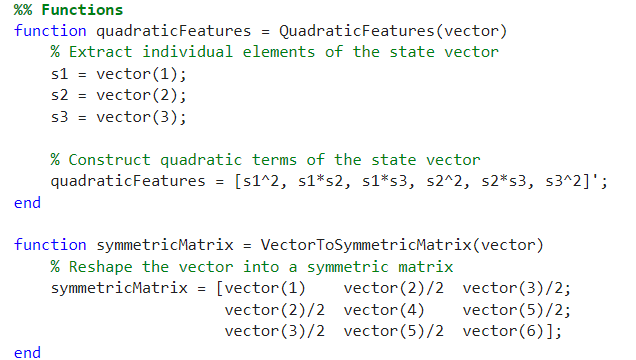
براساس این الگوریتم حلقه آموزش اصلی را می‌نویسیم.

ابتدا مقادیر سای و فی که براساس اسلایدهای درس لازم است برای محاسبه مقادیر بهره K1 و K2 استفاده شوند را تعریف می‌کنیم.

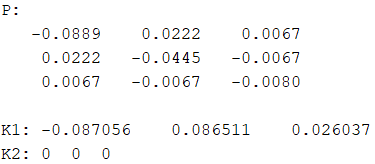
حال حلقه محاسبه انتگرال را می‌نویسیم.

ابتدا نویز پروب را برای محاسبه u رفتار لحاظ می‌کنیم. در مرحله بعد پالیسی هدف را محاسبه می‌کنیم. براساس الگوریتم داده شده و رابطه انتگرالی آن، بروزرسانی ارزش را پیاده سازی می‌کنیم. سپس خطا محاسبه می‌شود و در ادامه حالت‌ها براساس پالیسی‌های رفتار بروز می‌شوند.

در ادامه مقادیر فی و سای بروز شده و بر اساس آن‌ها پارامترهای L محاسبه می‌شوند.

و با استفاده از پارامترهای L نیز بهره K برای ورودی و اغتشاش محاسبه می‌شود. شرط توقف نیز براساس ماتریس P چک می‌شود.

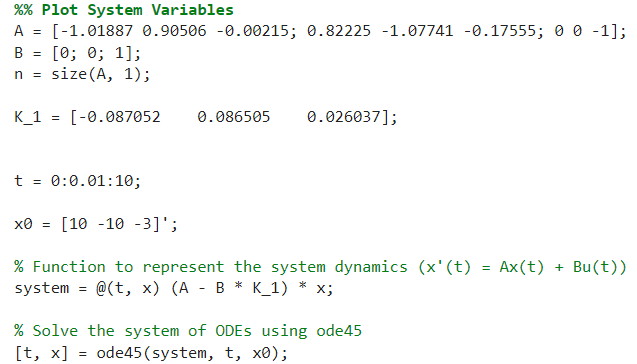
از توابع بالا نیز برای تبدیل P در حالت برداری به P در حالت ماتریسی متقارن و از تابع اول نیز برای محاسبه ضرب کرونکر بردار حالت در خودش استفاده می‌شود.

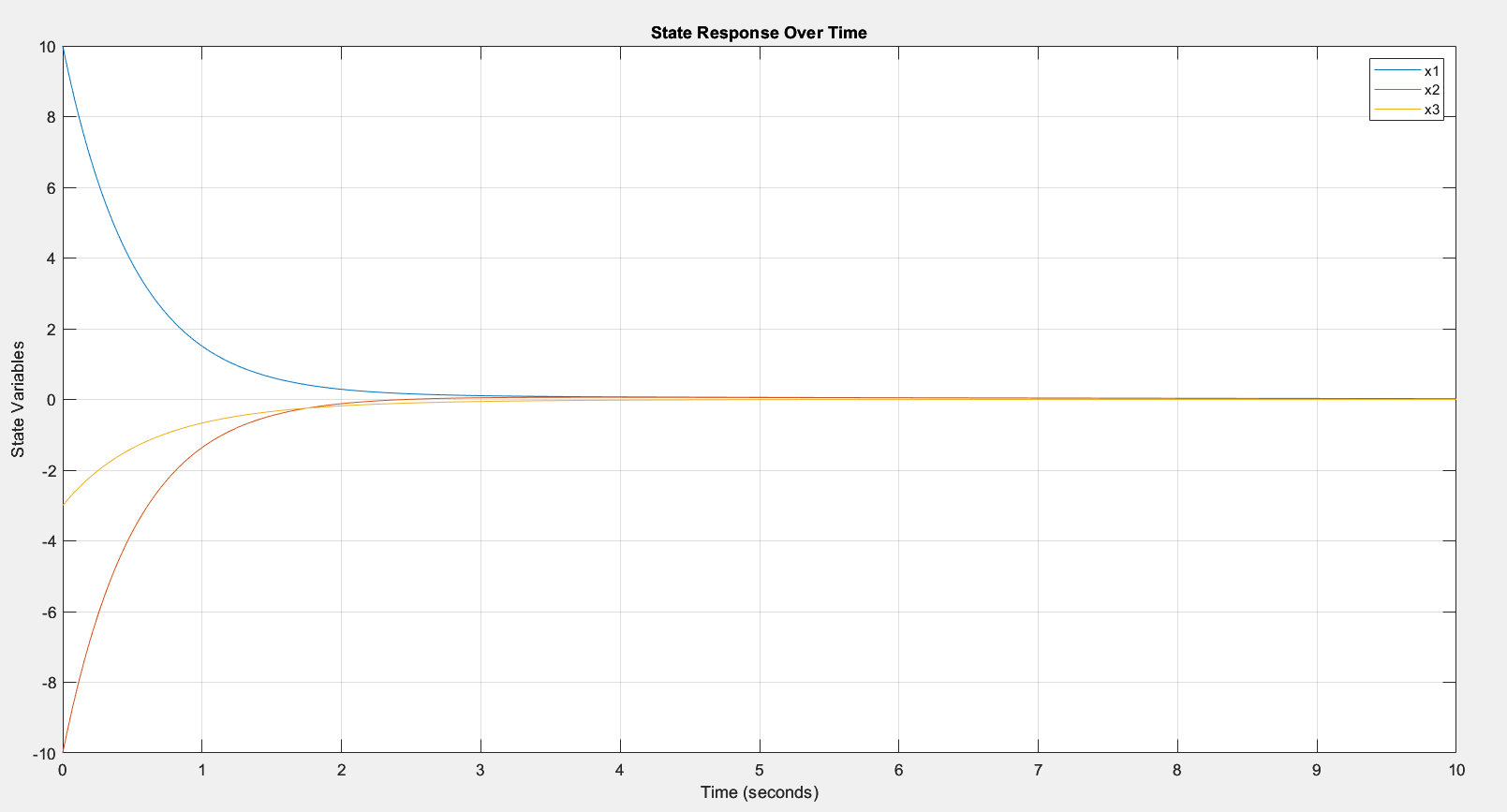
با اجرای کد به مقادیر زیر دست می‌یابیم.

شکل 8: مقادیر P و K1 و K2

**صورت سوال قسمت پ:**

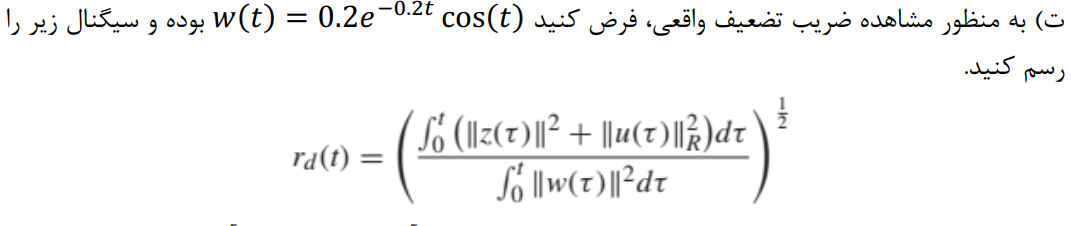
**پاسخ:**

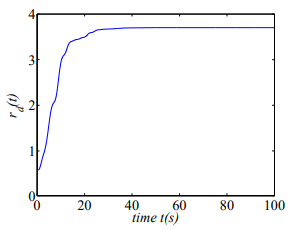
 با توجه به اینکه مقدار درایه‌های بردار K2 صفر است و نویز را نیز باید حذف کنیم، بخش بعدی سوال به صورت زیر پیاده سازی گردید:

پاسخ حالت‌ها نیز به صورت زیر است:

شکل 9: پاسخ حالت‌ها

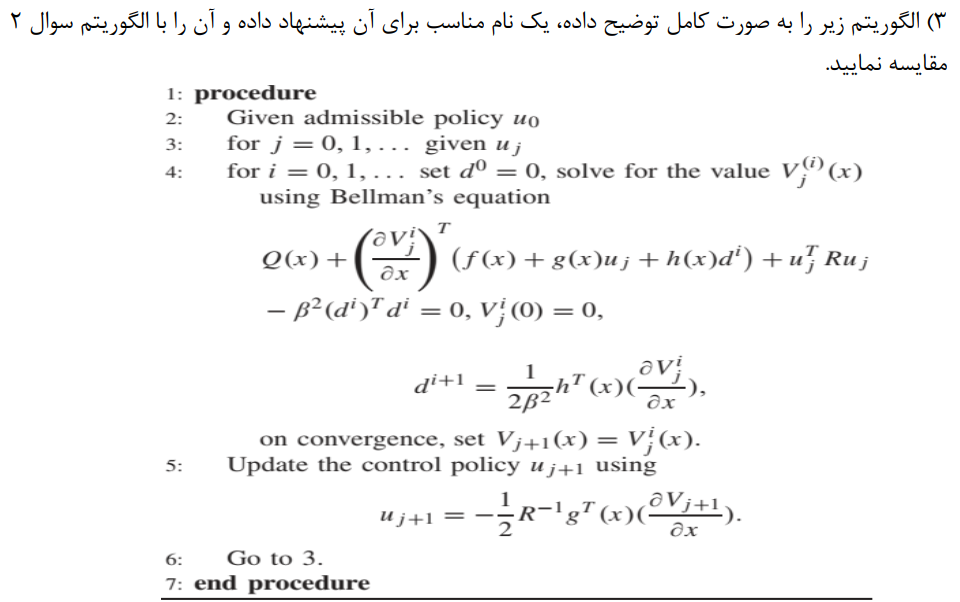
**صورت سوال قسمت ت:**

**پاسخ:**

مطابق محاسبات، نمودار این بخش به صورت زیر است و انتظار می‌رود مقدار rd به 3.7024 میل کند.

شکل 10: نمودار همگرایی rd

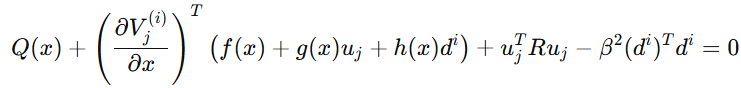
## سوال سوم

**صورت سوال:**

**پاسخ:**

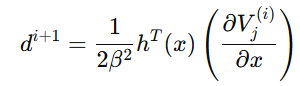
این الگوریتم، یک روش تکراری است و برای حل مسئله کنترلی مبتنی بر معادلات بلمن به کار می‌رود. هدف آن محاسبه تابع ارزش V(x)، کنترل بهینه و مقدار اغتشاش بهینه است. این فرآیند با استفاده از یک سیاست اولیه آغاز می‌شود و به صورت تدریجی به پاسخ‌های بهینه همگرا می‌شود.

ابتدا باید با یک سیاست اولیه مجاز و شدنی فرایند آغاز شود. مقدار اولیه اغتشاش نیز صفر لحاظ می‌شود.

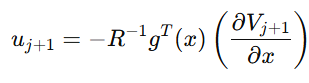
در این روش دو حلقه بروز رسانی داریم، در هر تکرار j از حلقه بیرونی، یک حلقه داخلی برای حل معادله بلمن وجود دارد. (معادله زیر)

که در آن Vj اولیه، صفر است و این معادله به صورت تحلیلی یا عددی برای تابع ارزش حل می‌شود.

سپس مرحله به روز رسانی انجام می‌گردد.

اغتشاش به صورت زیر به روز می‌گردد:

این به روز رسانی تا زمانی که مقدار ارزش همگرا شود ادامه می‌یابد.

سیاست کنترلی نیز از طریق رابطه زیر به روز می‌گردد:

حلقه بیرونی نیز در حال تکرار است تا زمانی که ارزش و سیاست کنترلی همگرا شوند، به مقدار بهینه برسند.

این روش، به دلیل داشتن دو فرآیند همزمان (ارزیابی و بهبود)، به روش GPI نیز نزدیک است. می‌توان برای آن از نام Generalized Policy Iteration for H∞​ Robust Control نیز استفاده نمود.

در مقایسه دو الگوریتم، الگوریتم دوم با استفاده از دو حلقه تکرار داخلی و بیرونی به دنبال یک حل دقیق‌تر و جامع‌تر است. از این رو، برای مسائلی که شامل سیستم‌های غیرخطی یا اختلالات پیچیده هستند، مناسب‌تر است. الگوریتم اول ساده‌تر است و برای حالت‌های ساده‌تر یا خطی کاربرد دارد.

در الگوریتم دوم برخلاف الگوریتم اول، معادله بلمن در یک حلقه درونی به صورت تکراری حل می‌شود، و از دو حلقه تکرار استفاده می‌گردد. رویکرد الگوریتم اول به صورت Offpolicy و IRL است.

الگوریتم دوم برخلاف قبلی از رویکرد انتگرالی اما استفاده نمی‌کند و می‌توان از آن به عنوان Offpolicy Differential RL نام برد. و همچنین احتمالا این الگوریتم برای مسئله H∞​ Tracking می‌باشد.