

# **Matlab Final Project Report**

Linear Control Systems

Shiva Vafadar  
810899074

## تحلیل سیستم

الف) دو تابع تبدیل خواسته شده به صورت زیر میشود:

$$T = k_t i \quad e = k_e \dot{\theta}$$

$$J\ddot{\theta} + b\dot{\theta} = k i \quad \text{و} \quad L \frac{di}{dt} + Ri = V - k\dot{\theta}$$

$$s(Js + b)\theta(s) = kI(s)$$

$$(Ls + R)I(s) = V(s) - k s \theta(s)$$

$$\Rightarrow P(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{k}{(Js + b)(Ls + R) + k^2}$$

$$\underline{G(s)} = \frac{k/s}{V(s) (Js + b)(Ls + R) + k^2}$$

ب) با تعیین متغیرهای حالت مناسب، توابع تبدیل بدست آمده در سوال را در فضای حالت به شکل زیر قرار دادیم:

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{k}{J} \\ -\frac{k}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} +$$

$$\begin{bmatrix} 0 \\ 1/L \end{bmatrix} V$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}$$

ت) تابع تبدیل که به شکل زیر میشود:

```
1 J = 0.01;  
2 b = 0.1;  
3 K = 0.01;  
4 R = 1;  
5 L = 0.5;  
6 s = tf('s');  
7 P_motor = K/((J*s+b)*(L*s+R)+K^2)
```

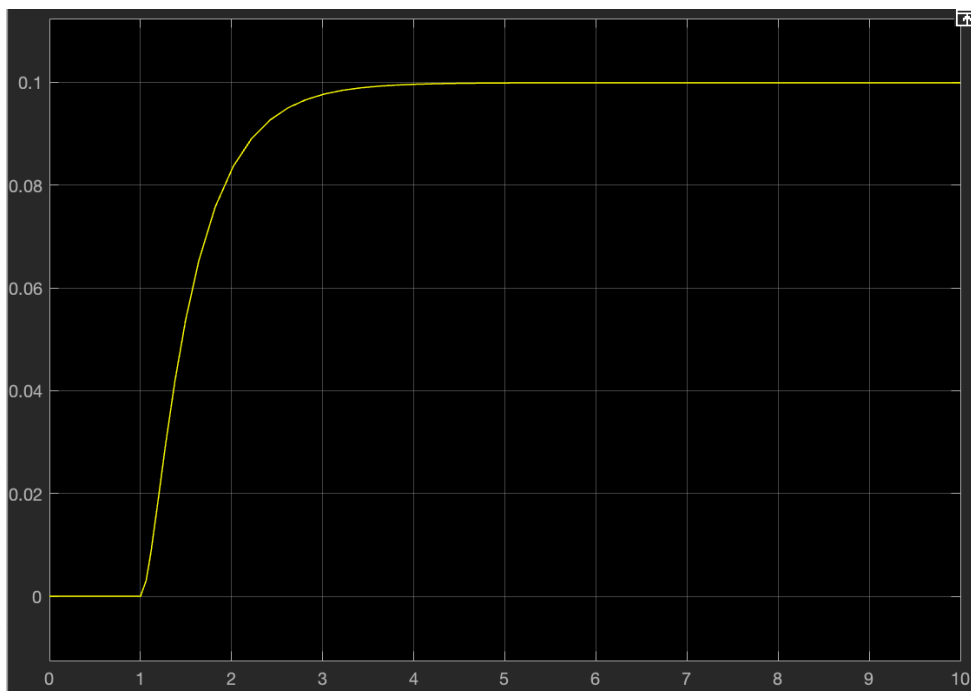
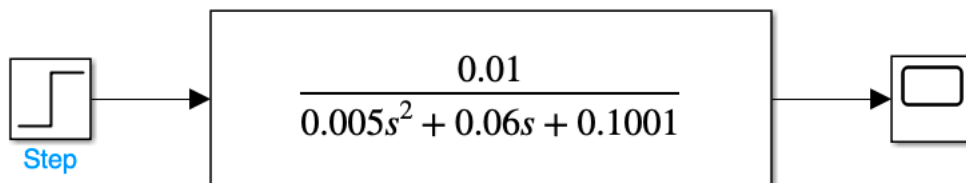
P\_motor =

0.01

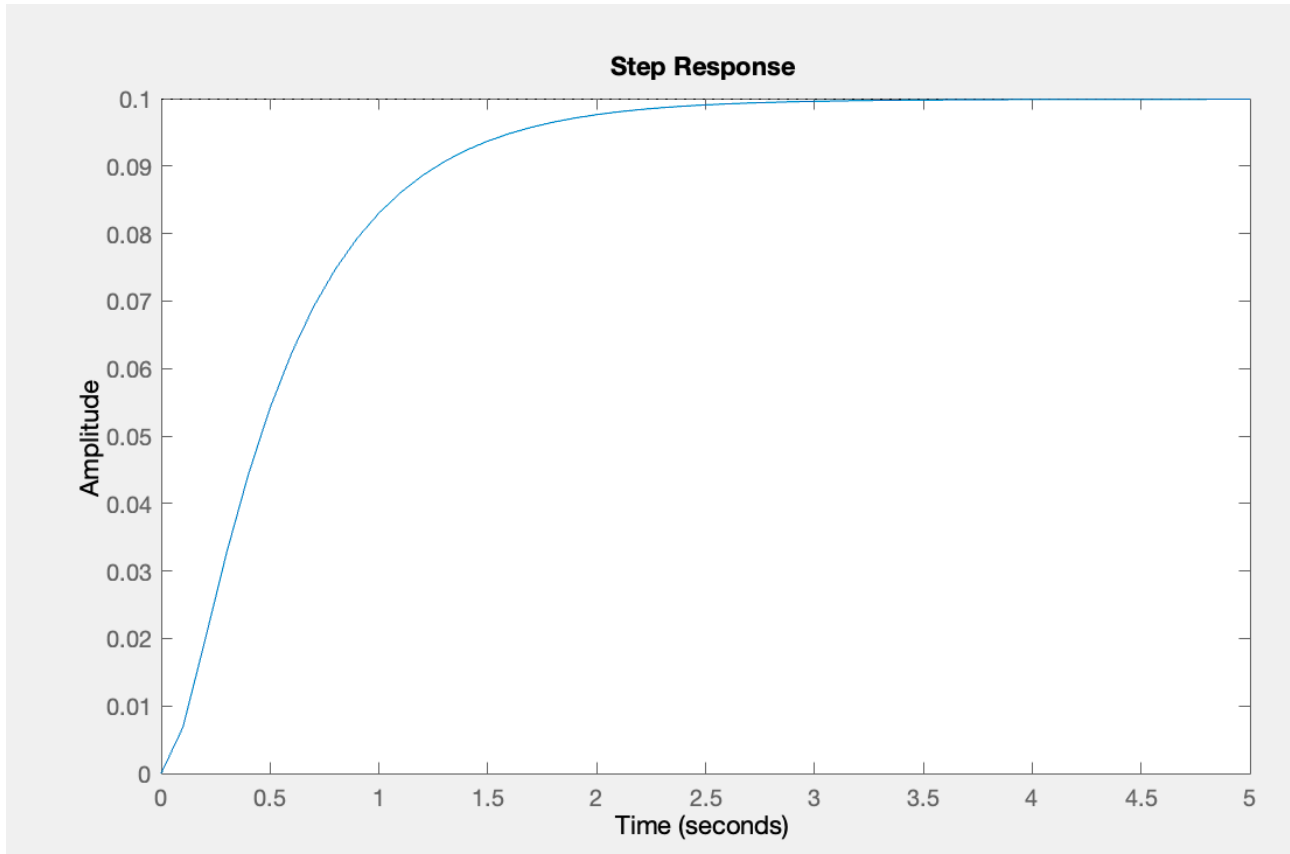
-----  
0.005 s^2 + 0.06 s + 0.1001

Continuous-time transfer function.

سپس پاسخ پله به این تابع تبدیل را به دست میآوریم. اینبار با استفاده از محیط سیمولینک:



همچنین میتوانیم با دستور `linearSystemAnalyzer` نیز بررسی کنیم و ورودی پله بدهیم :

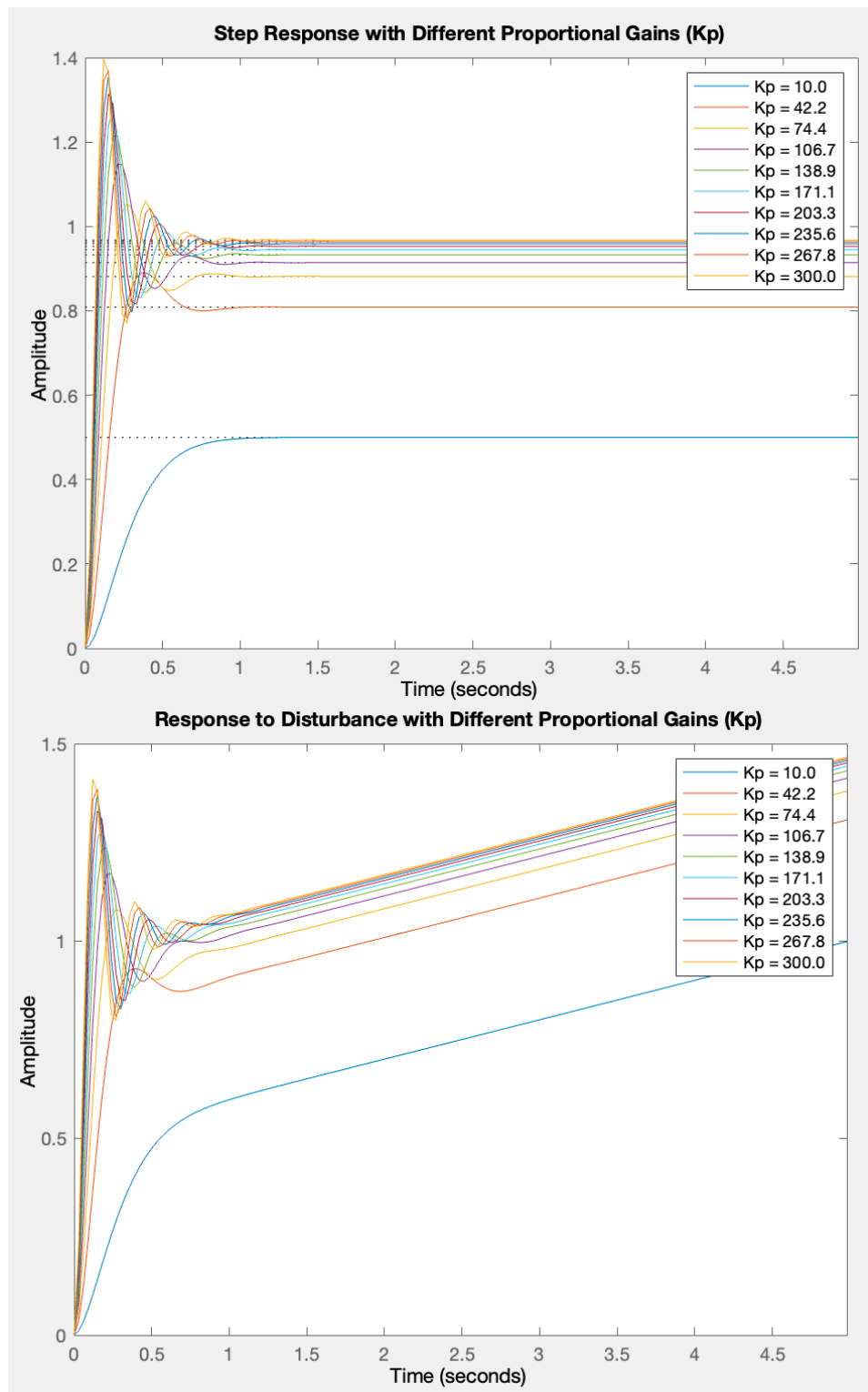


همانطور که در نمودار مشاهده می شود وقتی 1 ولت به سیستم اعمال می شود، موتور فقط می تواند حداکثر سرعت 0.1 رادیان در ثانیه را ارائه دهد که ده برابر کمتر از سرعت مورد نظر ما است. همچنین، موتور 2.07 ثانیه طول می کشد تا به سرعت ثابت خود برسد. بنابراین مملویات پروژه که زمان نشست کمتر از دو ثانیه است را ارضا نمی کند.

## بررسی انواع کنترلر در این سوال

(ث) با پیاده سازی تکه کد زیر، پاسخ به پله و پاسخ به اغتشاشات را مشاهده میکنیم:

```
Kp_values = linspace(10, 300, 10);
t = 0:0.03:5;
figure;
hold on;
grid on;
for Kp = Kp_values
    C = pid(Kp);
    sys_cl = feedback(C*G, 1);
    step(sys_cl, t);
end
title('Step Response with Different Proportional Gains (Kp)');
legend(arrayfun(@(Kp) sprintf('Kp = %.1f', Kp), Kp_values, 'UniformOutput', false));
figure;
hold on;
grid on;
disturbance = 0.1/s;
for Kp = Kp_values
    C = pid(Kp);
    sys_cl = feedback(C*G, 1);
    sys_cl_dist = sys_cl + disturbance;
    step(sys_cl_dist, t);
end
title('Response to Disturbance with Different Proportional Gains (Kp)');
legend(arrayfun(@(Kp) sprintf('Kp = %.1f', Kp), Kp_values, 'UniformOutput', false));
```



با اعمال این کنترلر می توانیم به زمان نشست دلخواه برسیم اما از تاثیر اغتشاشات جلوگیری نمیشود.

(ج)

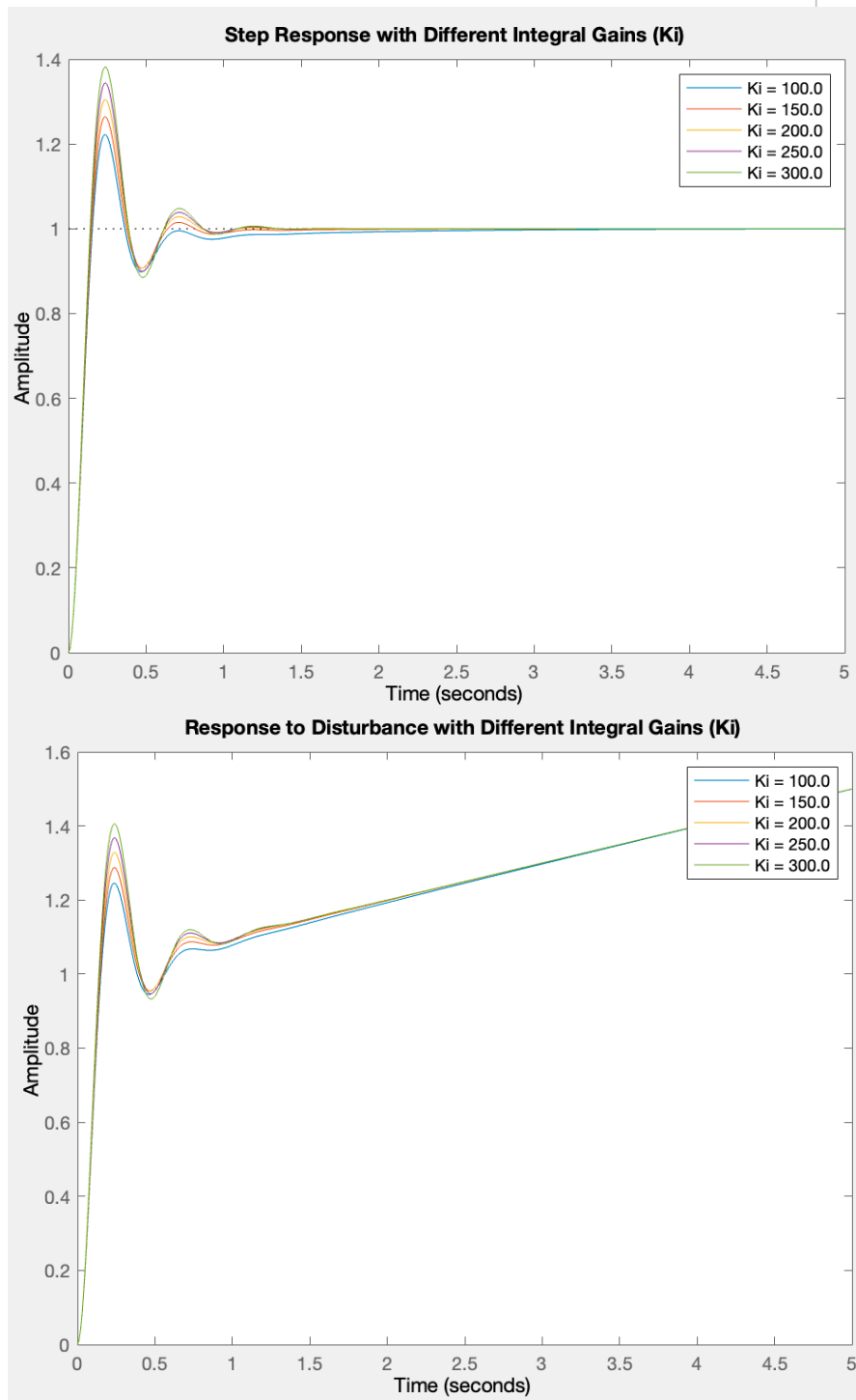
فرض میکنیم مقدار  $K_p = 100$  را داشته باشیم. حال مقادیر مختلف  $K_I$  را مابین بخش قبل بررسی میکنیم.

با پیاده سازی تکه کد زیر، پاسخ به پله و پاسخ به اغتشاشات را مشاهده میکنیم:

```

Kp = 100;
Ki_values = linspace(100, 300, 5);
t = 0:0.01:5;
figure;
hold on;
grid on;
for Ki = Ki_values
    C = pid(Kp, Ki);
    sys_cl = feedback(C*G, 1);
    step(sys_cl, t);
end
title('Step Response with Different Integral Gains (Ki)');
legend(arrayfun(@(Ki) sprintf('Ki = %.1f', Ki), Ki_values, 'UniformOutput', false));
figure;
hold on;
grid on;
disturbance = 0.1/s;
for Ki = Ki_values
    C = pid(Kp, Ki);
    sys_cl = feedback(C*G, 1);
    sys_cl_dist = sys_cl + disturbance;
    step(sys_cl_dist, t);
end
title('Response to Disturbance with Different Integral Gains (Ki)');
legend(arrayfun(@(Ki) sprintf('Ki = %.1f', Ki), Ki_values, 'UniformOutput', false));

```

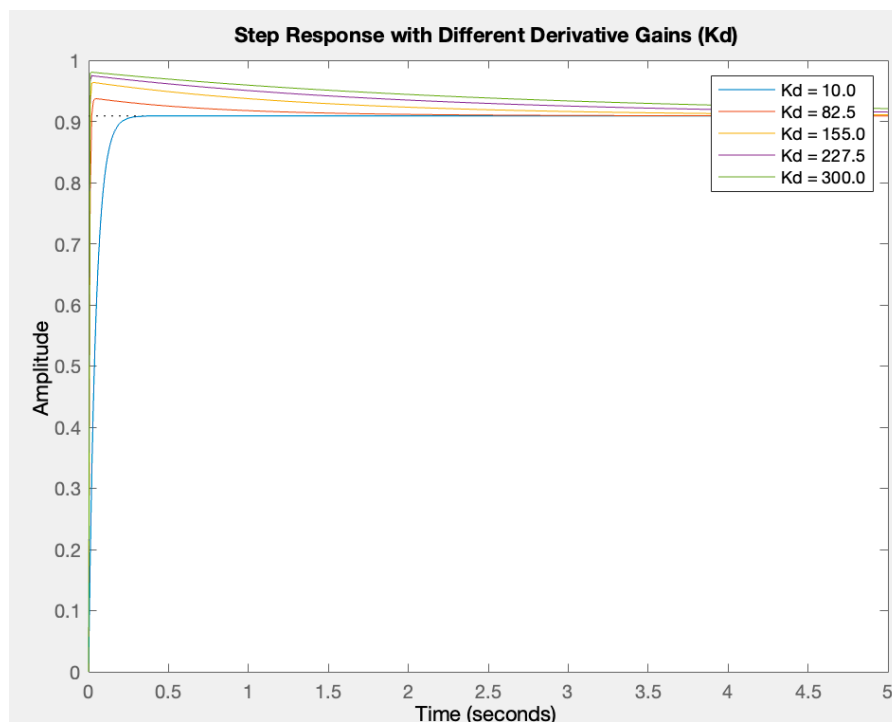


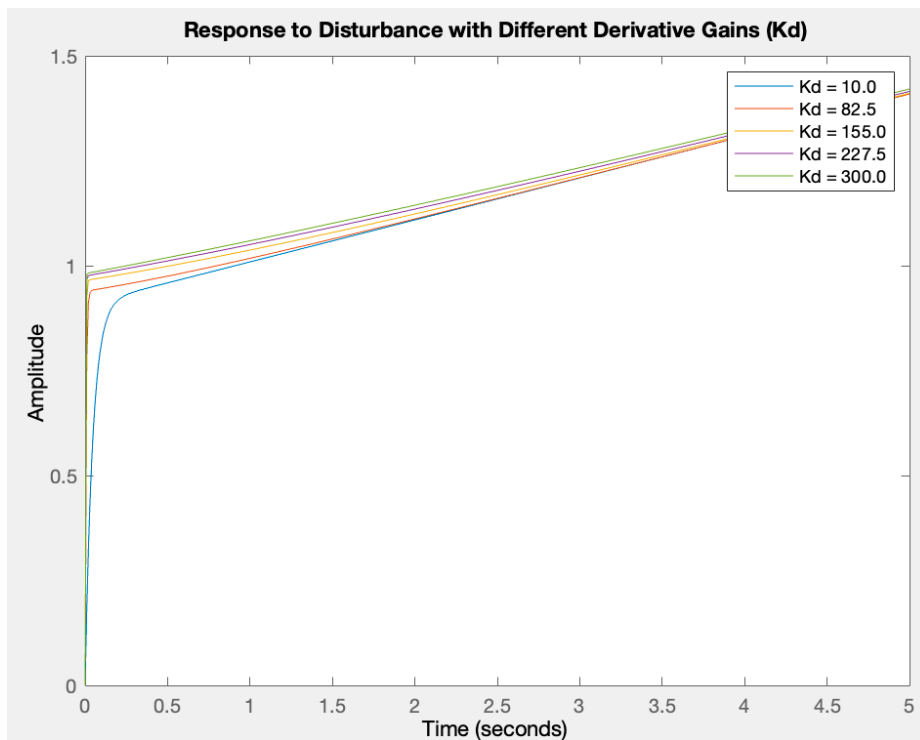
همانطور که مشاهده می شود با اعمال این کنترلر می توانیم به زمان نشست دلخواه، خطای ماندگار کمتر از یک درصد برسیم اما نتوانستیم به صورت کامل به فراجاهش دلخواه رسیده و تاثیر اغتشاش را حذف کنیم.

(ج)

حال با فرض  $K_p=100$ ، مقادیر مختلف برای  $K_d$  را مشابه با قسمت قبل، تست میکنیم، با پیاده سازی تکه کد زیر، پاسخ به پله و پاسخ به اغتشاشات را مشاهده میکنیم:

```
Kp = 100;
Kd_values = linspace(10, 300, 5);
t = 0:0.01:5;
figure;
hold on;
grid on;
for Kd = Kd_values
    C = pid(Kp, 0, Kd);
    sys_cl = feedback(C*G, 1);
    step(sys_cl, t);
end
title('Step Response with Different Derivative Gains (Kd)');
legend(arrayfun(@(Kd) sprintf('Kd = %.1f', Kd), Kd_values, 'UniformOutput', false));
figure;
hold on;
grid on;
disturbance = 0.1/s;
for Kd = Kd_values
    C = pid(Kp, 0, Kd);
    sys_cl = feedback(C*G, 1);
    sys_cl_dist = sys_cl + disturbance;
    step(sys_cl_dist, t);
end
title('Response to Disturbance with Different Derivative Gains (Kd)');
legend(arrayfun(@(Kd) sprintf('Kd = %.1f', Kd), Kd_values, 'UniformOutput', false));
```

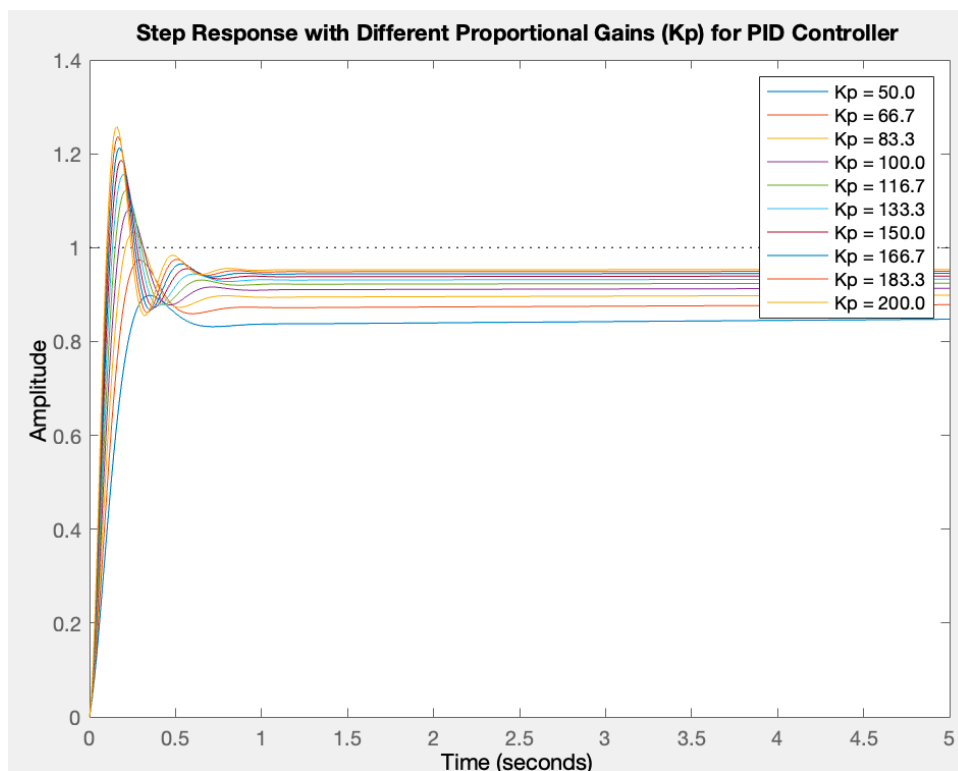




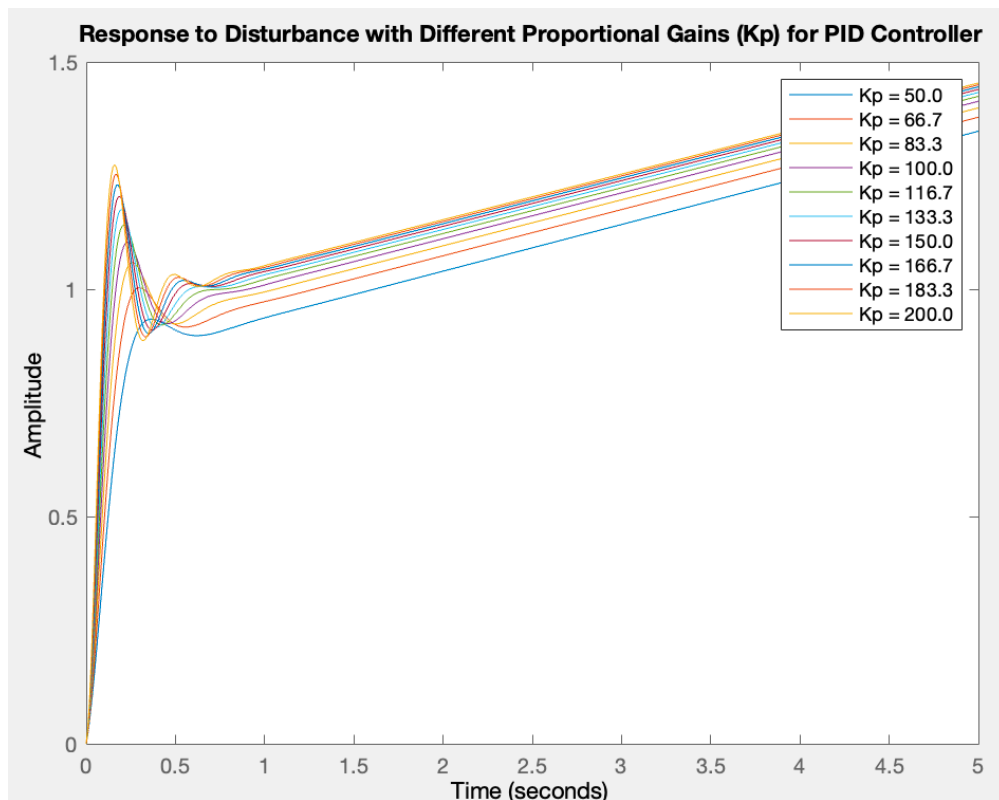
همانطور که مشاهده می شود با اعمال این کنترلر می توانیم به زمان نشست دلخواه، خطای ماندگار کمتر از یک درصد برسیم و همچنین فراجش کمتر از 5 درصد برسیم.

### بررسی کنترلر PID

ح) حال با فرض  $k_d=k_i=1$ ، مقادیر مختلف برای  $k_p$  را مشابه با قسمت قبل، تست میکنیم، با پیاده سازی تکه کد زیر، پاسخ به پله و پاسخ به اغتشاشات را مشاهده میکنیم:  
(تکه کد ها را دیگر قرار نمیدهیم چون مشابه بخشهای قبل میباشد. در صورت نیاز فایل پیوست شده را بررسی کنید.)



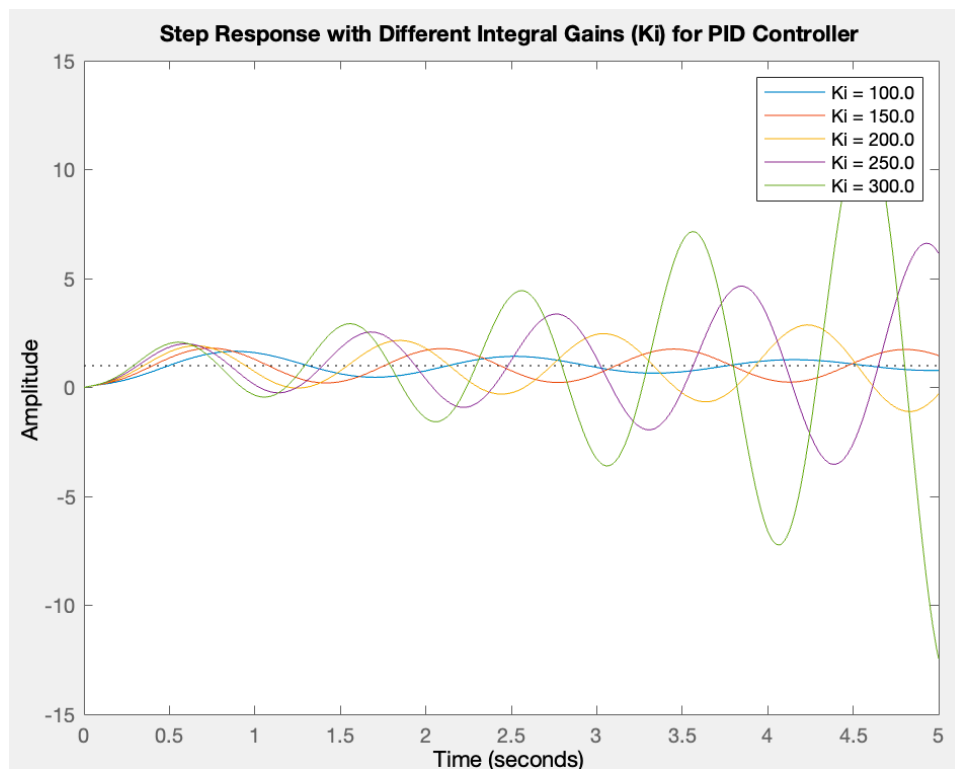


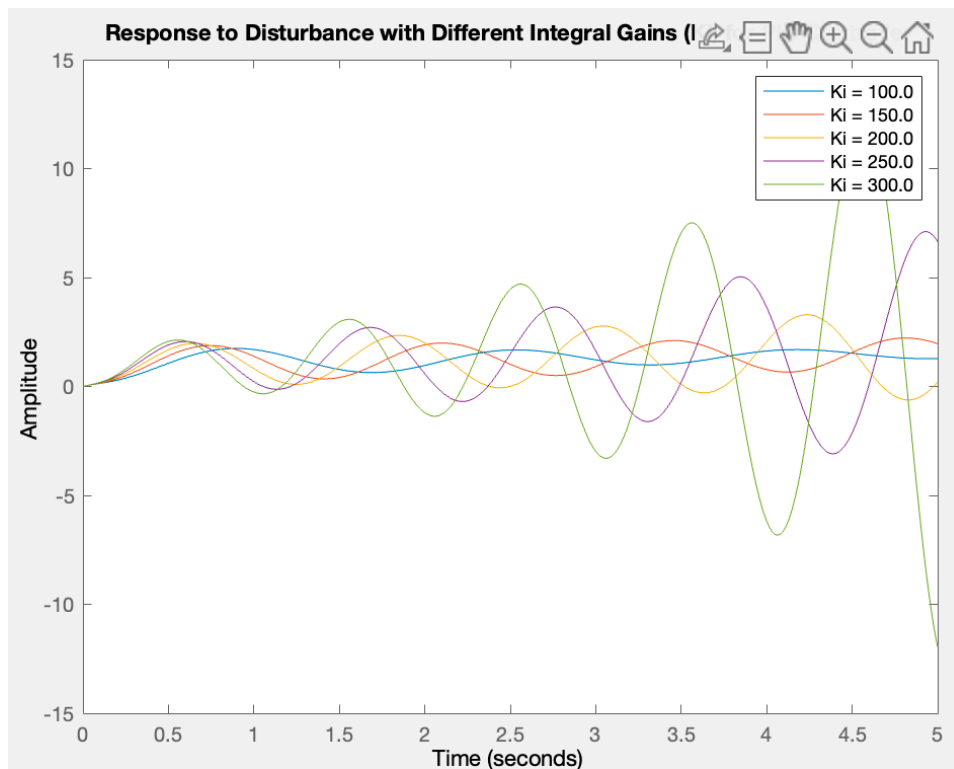


با افزایش  $k_p$  فرایند افزایش می یابد اما خطای ماندگار به صفر نزدیک می شود. در قسمت ث نیز همین روند را مشاهده می کردیم که از تاثیر  $p$  در کنترلر است که در زمان نشست و خطای ماندگار می تواند تاثیر گذار باشد.

(خ) حال با فرض  $k_d=k_p=1$ ، مقادیر مختلف برای  $k_i$  را مشابه با قسمت قبل، تست میکنیم، با پیاده سازی تکه کد زیر، پاسخ به پله و پاسخ به اغتشاشات را مشاهده میکنیم:

(تکه کد ها را دیگر قرار نمیدهیم چون مشابه بخشهای قبل میباشد. در صورت نیاز فایل پیوست شده را بررسی کنید.)

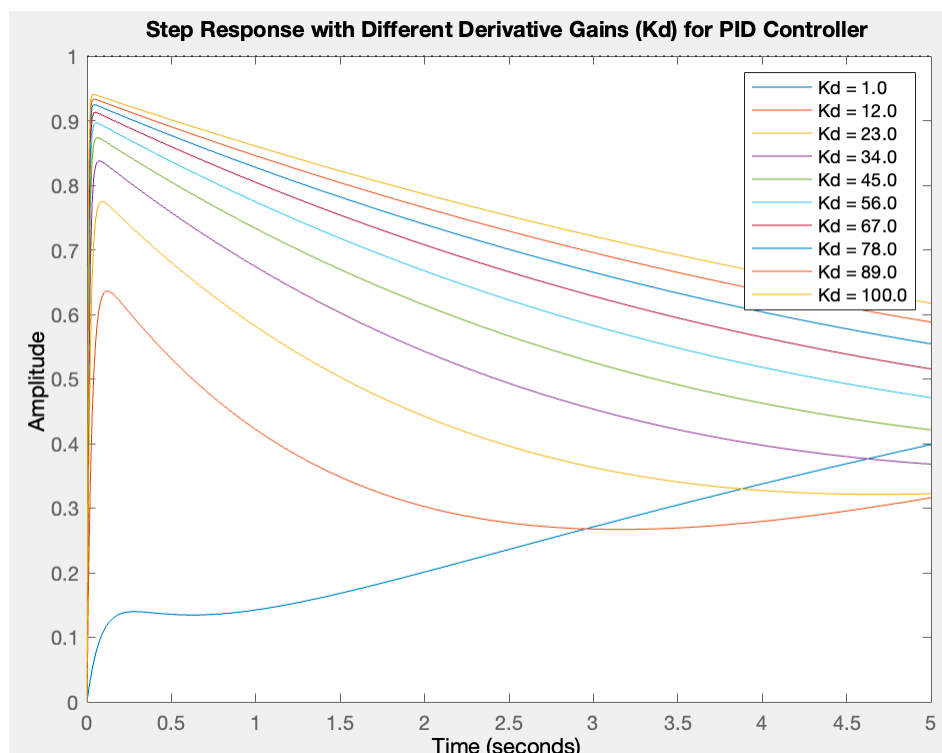


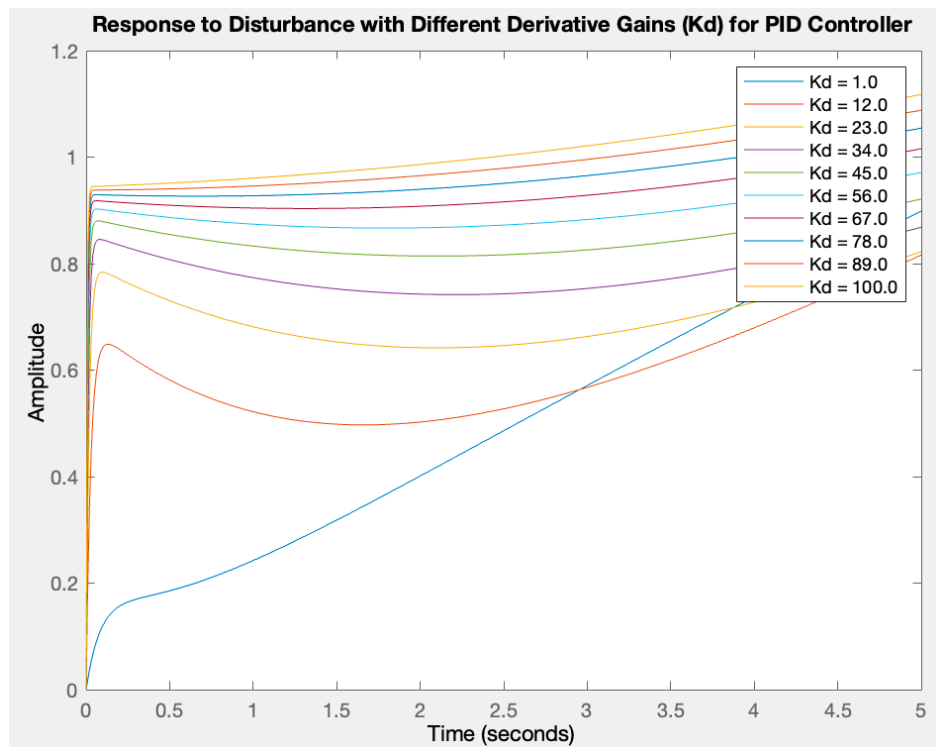


نکته قابل توجه این است که در این قسمت به حذف اغتشاش امید پیدا می کنیم. کنترل زمان نشست مناسبی نداریم. با افزایش مقدار  $k_i$  به فراجشش کمتری می رسیم. در قسمت ج عدد مناسبی برای  $k_p$  در نظر گرفته بودیم که زمان نشست مناسب نیز به ما داده بود.

د) حال با فرض  $k_i=k_p=1$ ، مقادیر مختلف برای  $k_d$  را مشابه با قسمت قبل، تست میکنیم، با پیاده سازی تکه کد زیر، پاسخ به پله و پاسخ به اغتشاشات را مشاهده میکنیم:

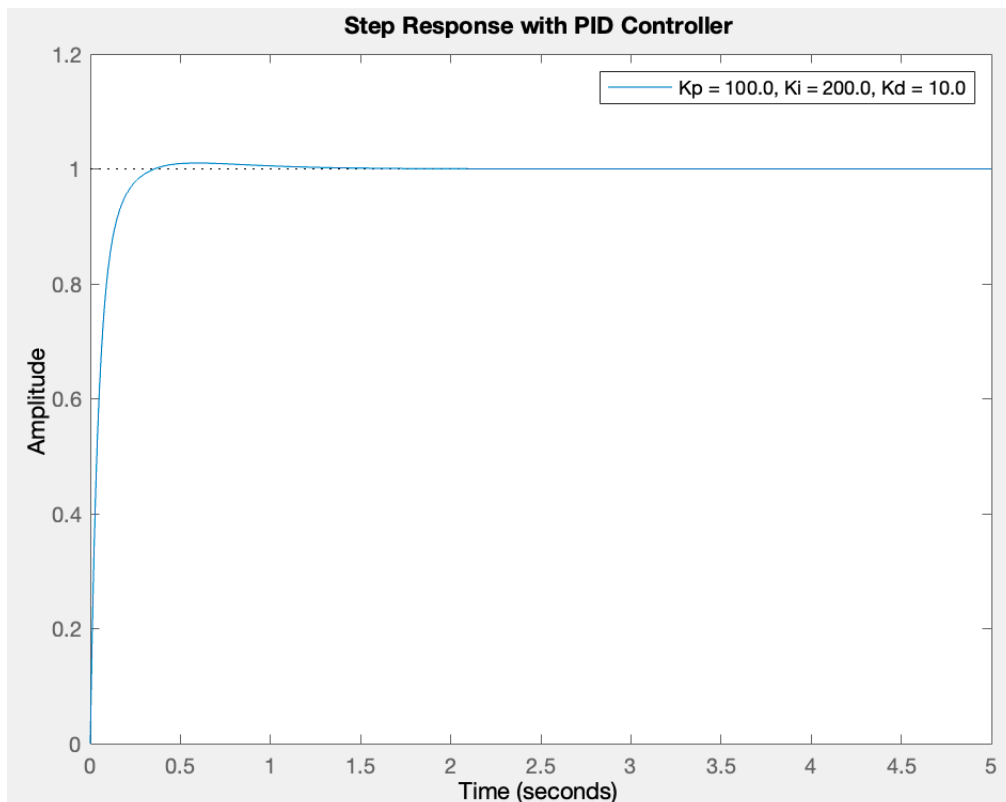
(تکه کد ها را دیگر قرار نمیدهیم چون مشابه بخشهای قبل میباشد. در صورت نیاز فایل پیوست شده را بررسی کنید.)

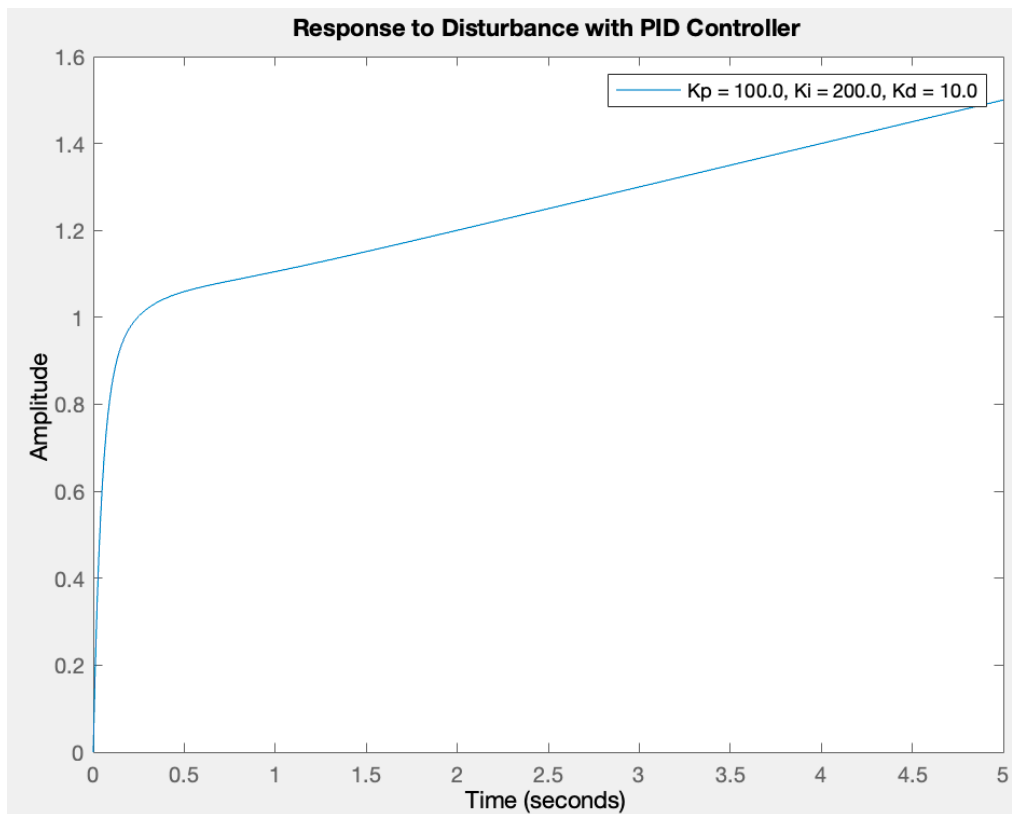




با افزایش  $K_d$  به فرا جهش کمتری می توانیم برسیم. در مقایسه با قسمت چ باید گفت زمان نشست مناسبی بدست نیامده. و در نهایت باید به مقادیر مناسبی برای همه ی ضرایب با توجه به سوال های گذشته برسیم.

ذ) با تحلیل از روی نمودار های بخش (بررسی انواع کنترلر در این سوال)، به یکسری مقادیر دست میایم که با توجه به توضیحاتی که در هر بخش داده شد، به طور تقریبی مقادیر زیر را برای کنترلر نهایی انتخاب کردیم:  
 $k_i=200, k_p=100, k_d=10$

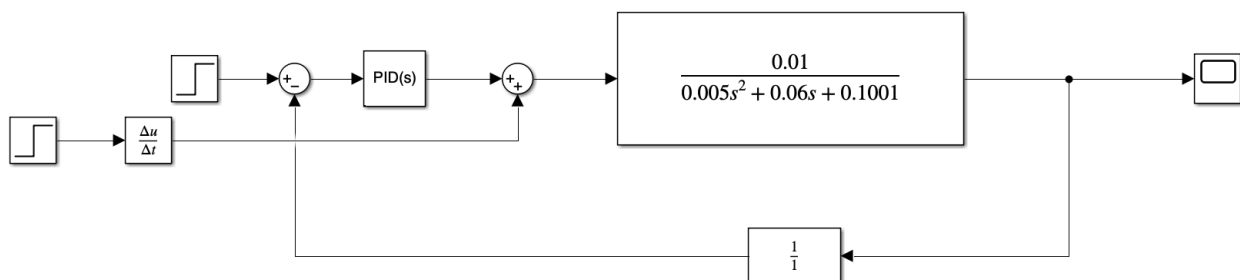




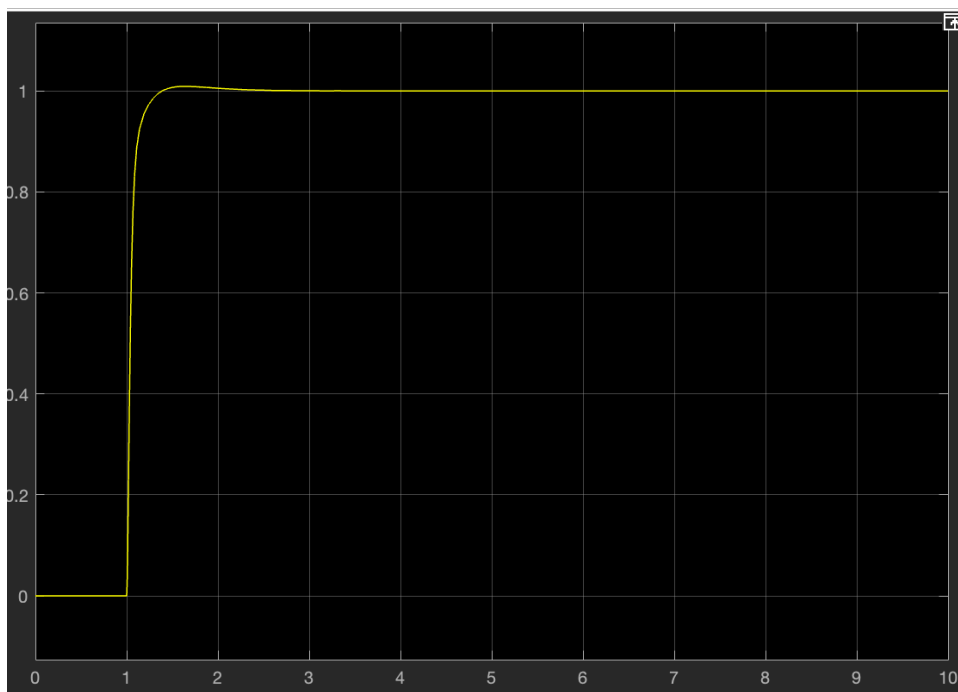
افزایش  $k_p$  می‌تواند زمان نشست را کاهش دهد، به این معنی که سیستم سریع‌تر به وضعیت پایداری خود می‌رسد. می‌تواند زمان خیزش را کاهش دهد، یعنی زمانی که سیستم به نقطه مطلوب وارد می‌شود را کمتر می‌کند. ممکن است موجب افزایش فرجهش شود. معمولاً باعث کاهش خطای حالت دائمی می‌شود، به شرطی که سیستم به وضعیت پایداری برسد. افزایش  $k_i$  می‌تواند خطای حالت دائمی را کاهش دهد، به ویژه برای سیستم‌هایی که با خطای حالت دائمی ثابت روبرو هستند. ممکن است موجب افزایش فرجهش شود، اگرچه این تأثیر معمولاً کمتر از  $k_p$  است.  $k_d$  معمولاً تأثیر کمتری بر زمان نشست دارد. افزایش  $k_d$  ممکن است زمان خیزش را کمتر کند، معمولاً موجب کاهش فرجهش می‌شود. تأثیر چندانی بر خطای حالت دائمی ندارد، مگر در سیستم‌هایی که دارای خطای حالت دائمی غیرصفر هستند. با توجه به این تأثیرات، تنظیم مناسب این ضرایب بر اساس ویژگی‌های خاص سیستم و موارد کاربردی مختلف ضروری است تا عملکرد مطلوبی از کنترل PID حاصل شود.

## مدلسازی در سیمولینک

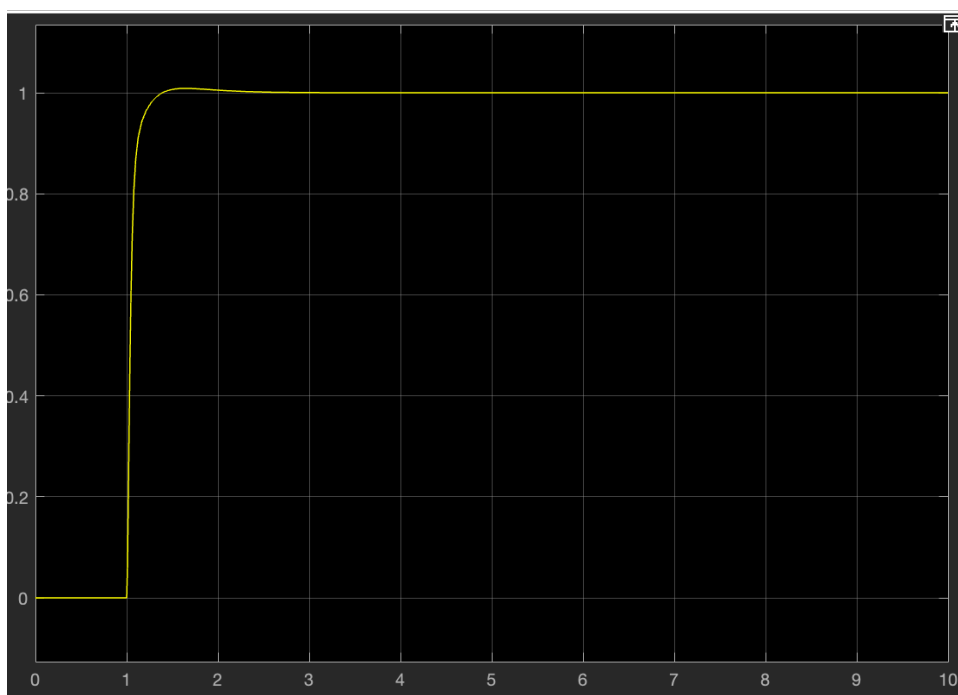
(ر) برای این بخش، سیستم را به شکل زیر در سیمولینک مدل کردم:



اگر سیم ورودی اغتشاش ضربه را حذف کنم، پاسخ به شکل زیر میشود:

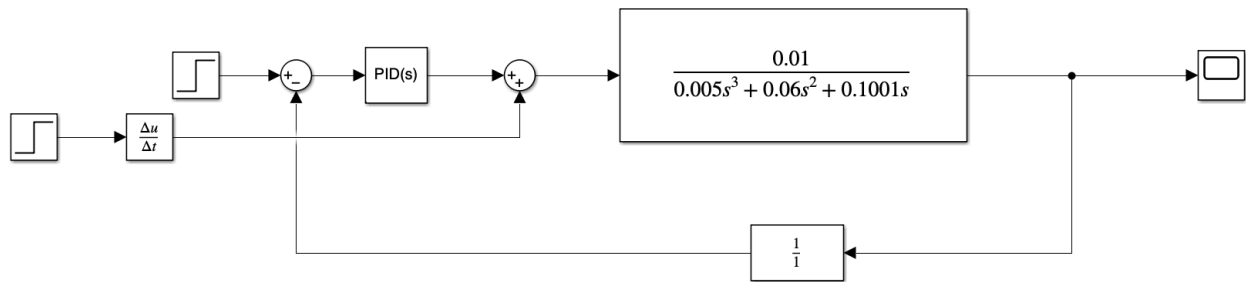


حال ورودی پاسخ ضربه را پس از کنترلر در سیستم قرار مدهم. مانند شکل بالا، اینبار با وجود اغتشاش، پاسخ به شکل زیر شد:



میبینیم که اغتشاش وارد شده، تاثیری در پاسخ ما نداشته.

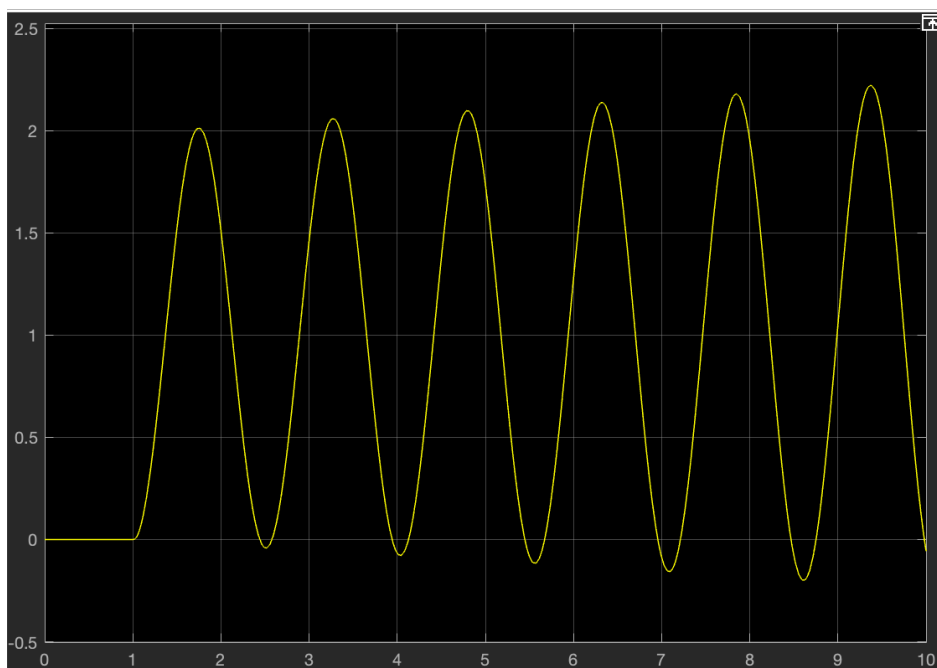
(ز) تابع تبدیل که خروجی سرعت زاویه ای داشت در بخش قبل بررسی شد. اینبار میخواهیم تابع تبدیل با خروجی موقعیت را مورد بررسی قرار دهیم. برای این کار، ابتدا سیستم را در محیط سیمولینک مدل میکنیم که به شکل زیر درآمد:



اگر سیم ورودی اغتشاش ضربه را حذف کنم، پاسخ به شکل زیر میشود:



حال ورودی پاسخ ضربه را پس از کنترلر در سیستم قرار مدهم. مانند شکل بالا، اینبار با وجود اغتشاش، پاسخ به شکل زیر شد:



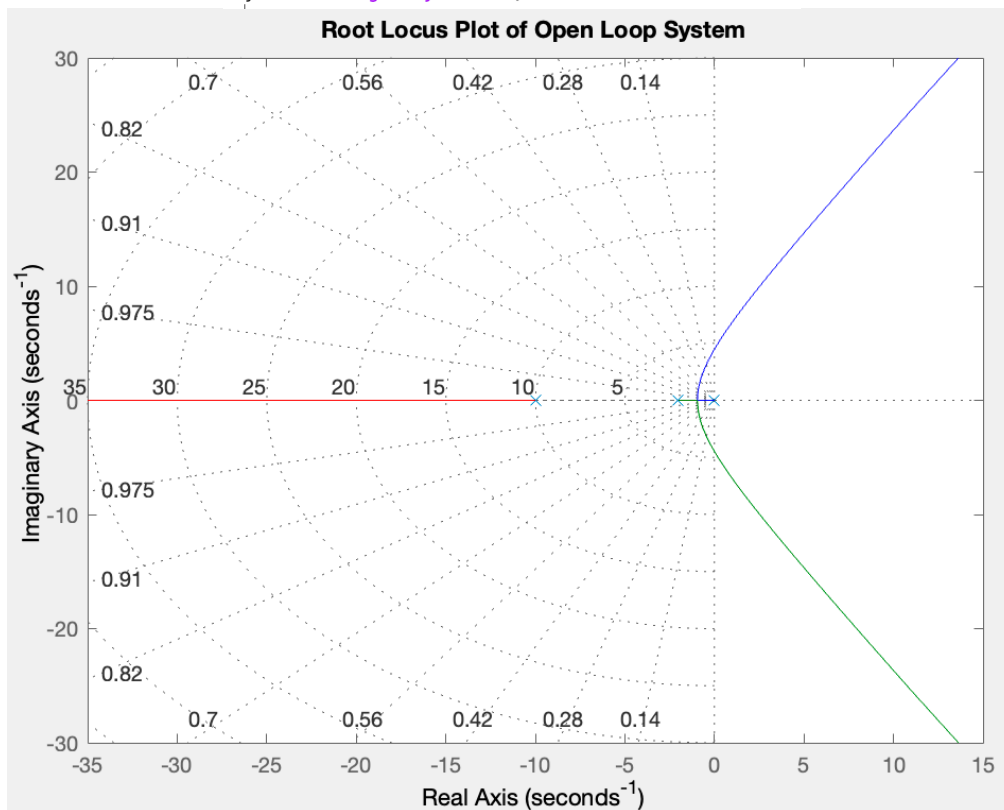
میبینیم که اغتشاش وارد شده، تاثیری در پاسخ ما نداشته.

## بررسی مکان هندسی ریشه های سیستم

(ژ) با اجرای تکه کد زیر به نمودار مکان هندسی میرسیم که به شکل زیر درمیاید:

```
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;
s = tf('s');
G = K/(s*((J*s+b)*(L*s+R)+K^2));

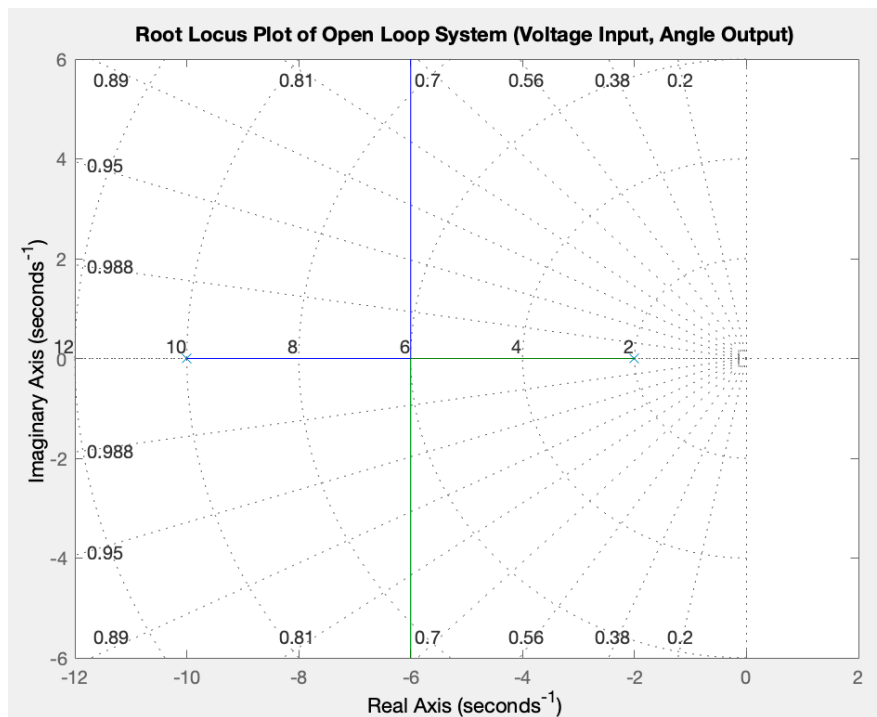
rlocus(G);
grid on;
title('Root Locus Plot of Open Loop System');
xlabel('Real Axis');
ylabel('Imaginary Axis');
```



به ازای  $k$  هایی که روی نقاط مرزی (روی محور imaginary) به بعد، سیستم ناپایدار میشود. (س) با اجرای تکه کد زیر به نمودار مکان هندسی میرسیم که به شکل زیر درمیاید:

```
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;
s = tf('s');
G = K/((J*s+b)*(L*s+R)+K^2);

rlocus(G);
grid on;
title('Root Locus Plot of Open Loop System (Voltage Input, Angle Output)');
xlabel('Real Axis');
ylabel('Imaginary Axis');
```



همه ی ریشه ها منفی هستند . بنابراین سیستم پایدار است.

### بررسی پاسخ فرکانسی سیستم

ش) با اجرای تکه کد زیر، نمودار های بود و نایکویست به شکل زیر میشوند:

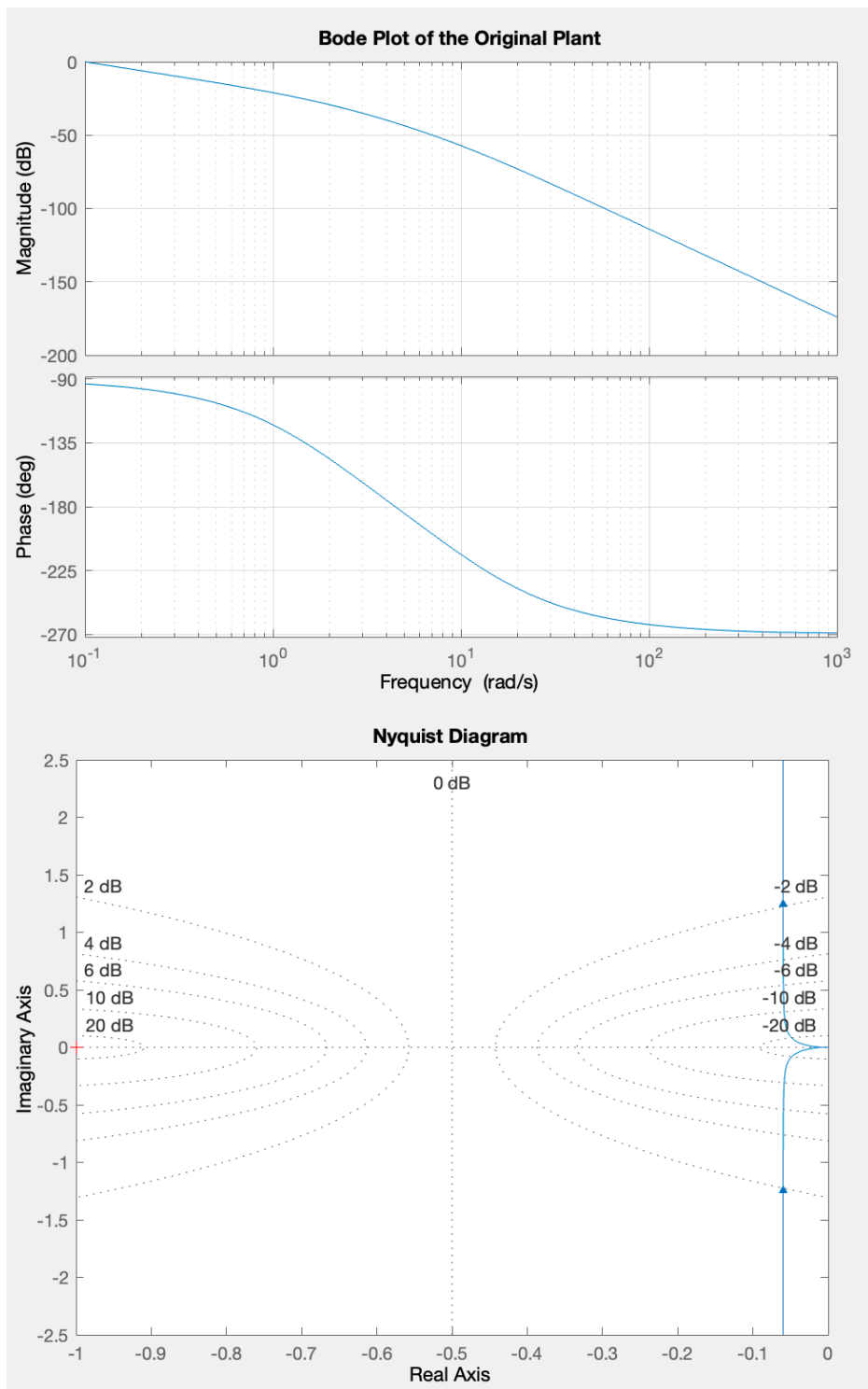
```
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;

s = tf('s');
G = K/(s*((J*s+b)*(L*s+R)+K^2));

figure;
bode(G)
title('Bode Plot of the Original Plant')
grid on;

figure;
nyquist(G);
grid on;
```





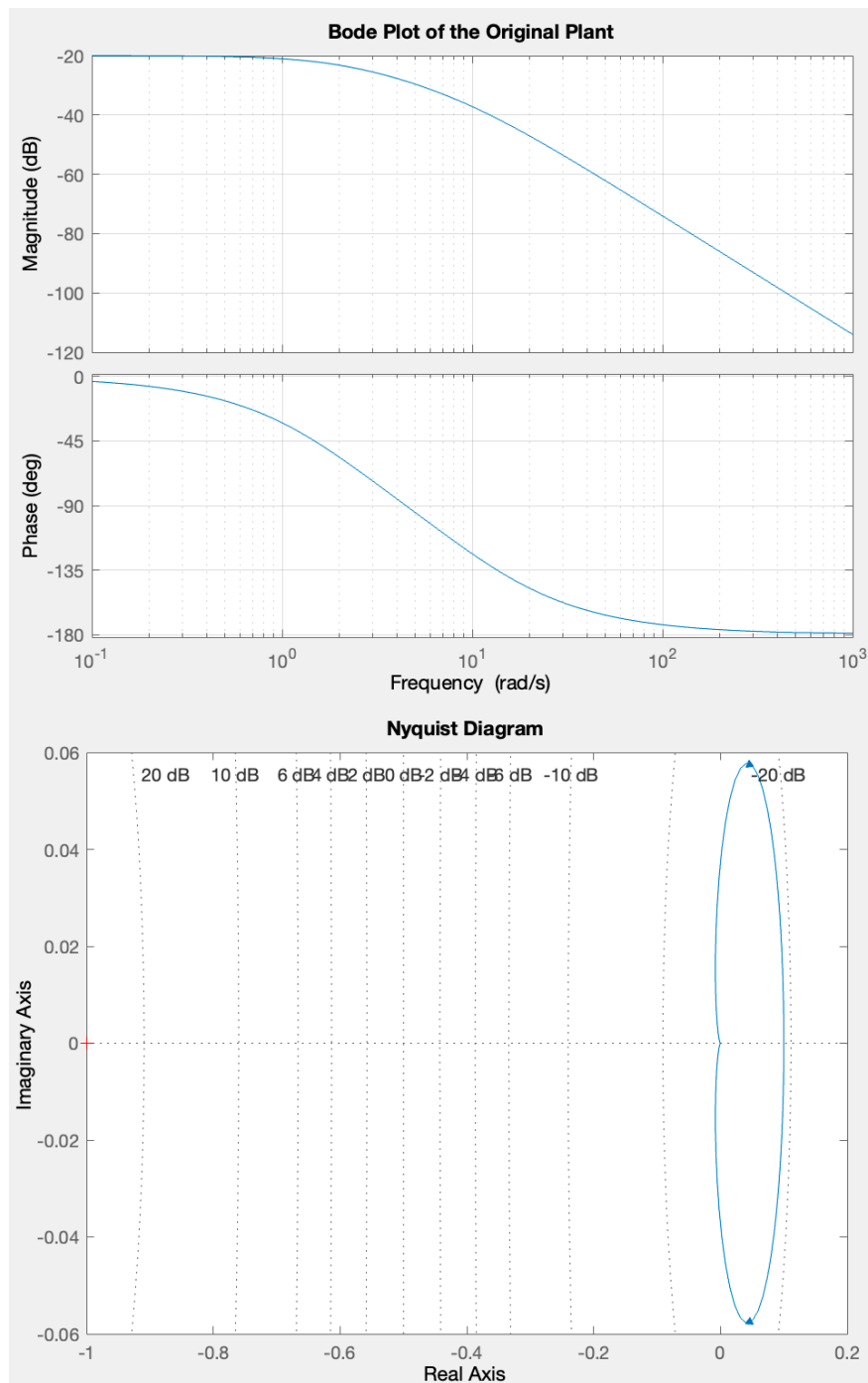
```
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;
```

```
s = tf('s');
G = K/((J*s+b)*(L*s+R)+K^2);
```

```
figure;
bode(G)
title('Bode Plot of the Original Plant')
grid on;
```

```
figure;
nyquist(G);
grid on;
```

(ص)



طبق اصل پایداری نایکوئیست، پایدار هستند.  
(ض)

```
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;

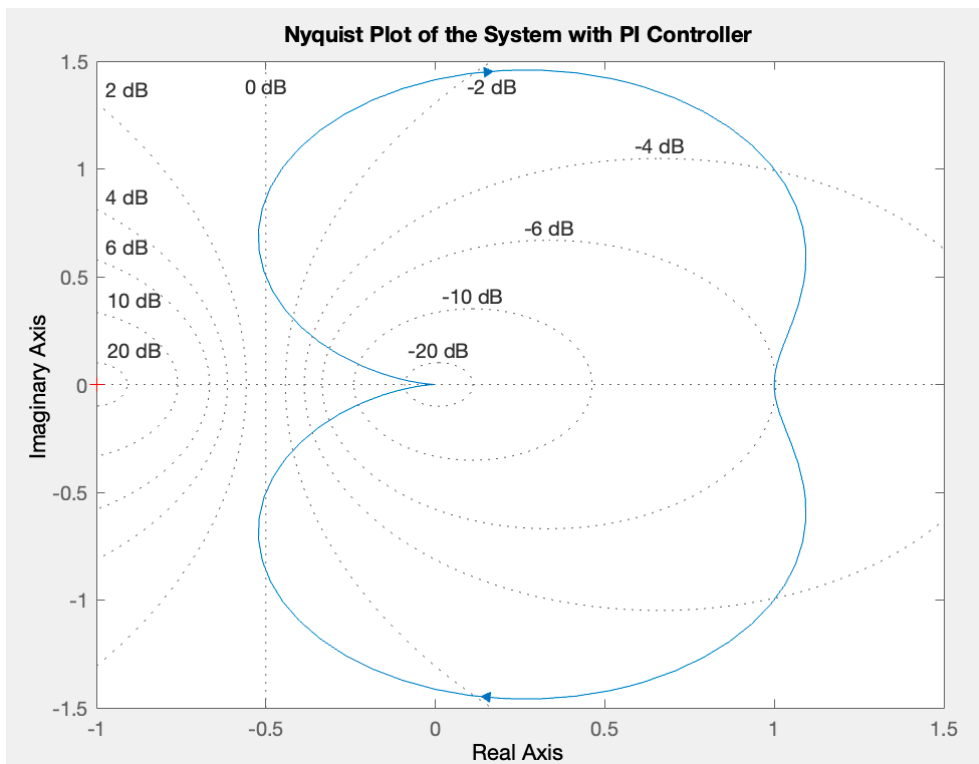
s = tf('s');
G = K / ((J*s + b) * (L*s + R) + K^2);

Kp = 100;
Ki = 200;

C = Kp + Ki/s;

T = feedback(G*C, 1);

figure;
nyquist(T);
grid on;
title('Nyquist Plot of the System with PI Controller');
```



ط) اثر تاخیر بر نمودار نایکوئیست:

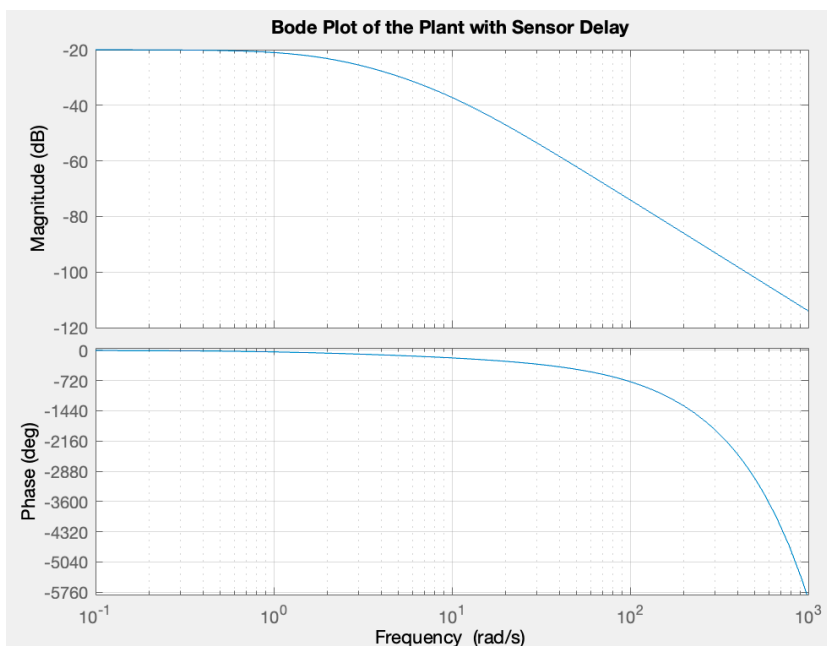
نمودار نایکوئیست نشان دهنده تغییرات فاز و مقیاس سیستم در طول تمامی فرکانس‌ها است. تاخیر در سنسور ممکن است به صورت زیر تأثیر بگذارد:

تغییر در مسیر نمودار: تاخیر می‌تواند باعث تغییر در مسیر نمودار نایکوئیست شود، به خصوص در فرکانس‌های بالا که تأثیر تاخیر به وضوح قابل مشاهده است.

تغییر در پایداری: اگر تاخیر زیاد باشد، ممکن است باعث نوسانات در نمودار نایکوئیست شود که نشان‌دهنده نوعی ناپایداری یا نوسان در سیستم است.

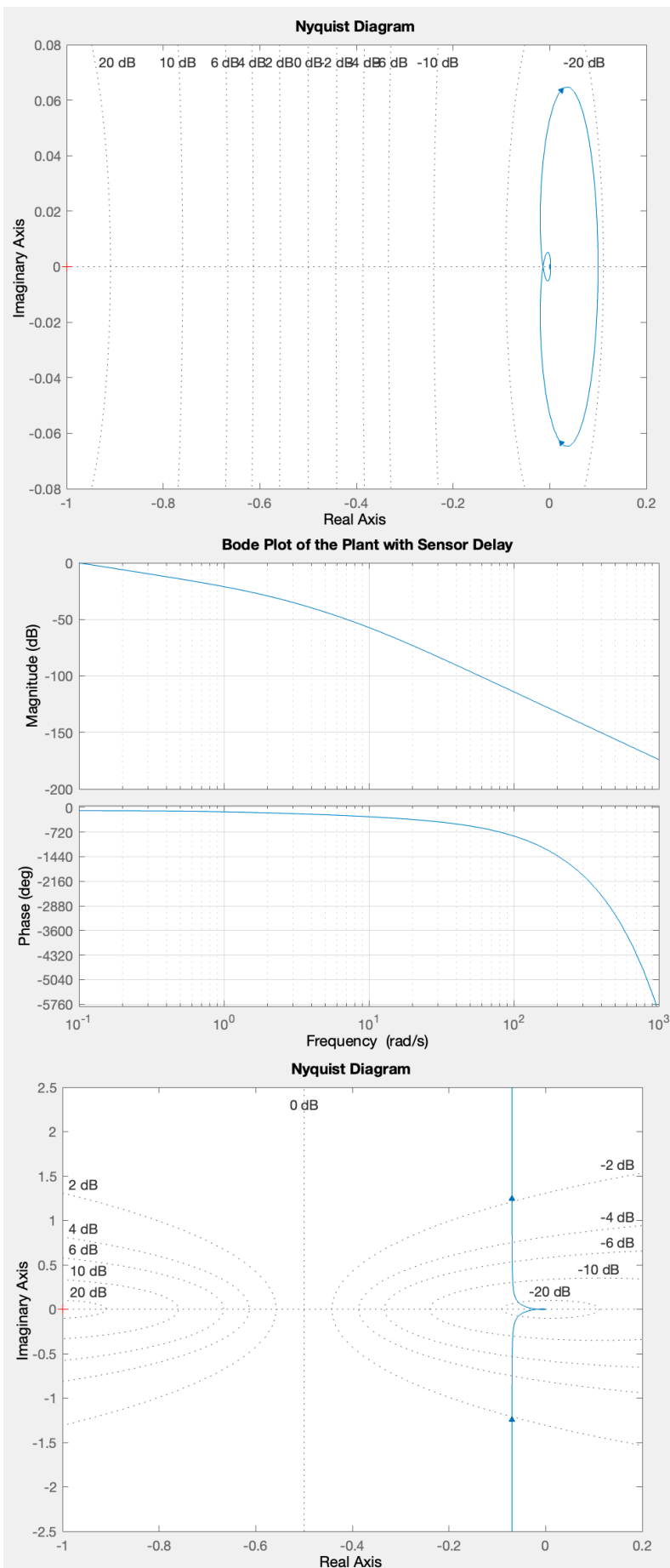
اثر تاخیر بر نمودار بُد:

نمودار بُد نشان دهنده پاسخ فرکانسی سیستم به اندازه مختلف فرکانس‌ها است. تاخیر در سنسور می‌تواند به دو شکل اصلی تأثیر بگذارد:



تغییر فاز: تاخیر ممکن است باعث تغییر فاز سیستم در فرکانس‌های مختلف شود. این تغییر فاز ممکن است در نقاط مختلف نمودار بُد رخ دهد، به خصوص در فرکانس‌های بالا که اثر تاخیر به وضوح قابل مشاهده است.

تغییر در پهنای باند: تاخیر می‌تواند باعث شود که پهنای باند سیستم به نسبت کمتر شود. این به این معنی است که سیستم به فرکانس‌های بالاتری پاسخ نمی‌دهد.



در اینجا من تاخیر را ۰.۱ قرار دادم.