درس آز کنترل خطی

تمرین سری اول

استاد درس: دکتر موشایی

محمدپارسا دینی

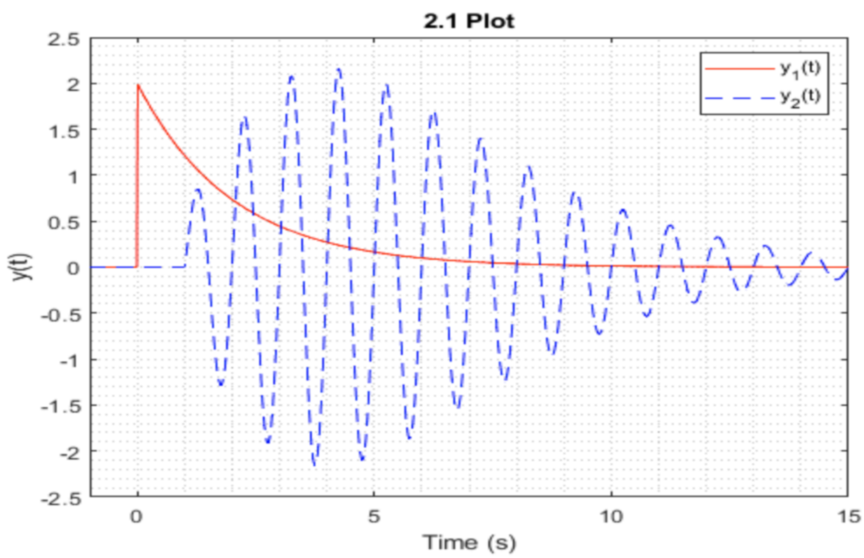
400101204

بخش 2.1) ترسیم

کد مربوطه:

```
% time vector
t = linspace(-1, 15, 1000);

% unit step functions
u1 = t >= 0;        % u(t)
u2 = t >= 1;        % u(t-1)

y1 = 2 * exp(-0.5 * t) .* u1;
y2 = (t.^2 .* sin(2 * pi * t) .* exp(-0.5 * t)) .* u2;

figure('Name','2.1 Plot');

% Plot y1(t)
plot(t, y1, 'LineWidth', 0.7,'Color',[1 0 0]);
% But 0.95 linewidth was closer to the plot
% plot(t, y1, 'LineWidth', 0.95,'Color',[1 0 0]);

hold on;                        % Hold the plot to add y2(t)

% Plot y2(t)
plot(t, y2, 'LineWidth', 0.7,'LineStyle','--','Color',[0 0 1]);
% But 0.95 linewidth was closer to the plot
% plot(t, y2, 'LineWidth', 1,'LineStyle','--','Color',[0 0 1]);

title('2.1 Plot');
xlabel('Time (s)');
xlim([-1 15])
ylabel('y(t)');
legend('y_1(t)', 'y_2(t)');
grid minor;
hold off;
```

خروجی ترسیم: (فایل '.fig' ضمیمه شده است)



آز کنترل
تمرین سری اول

بخش 2.2) ساختن مدل های خطی تغییرناپذیر با زمان

قسمت a)

```
% Part (a)

% Define state-space matrices
A = [-4, 2, 1; 1, -4, 1; -1, 0, -3];
B = [1; 0; 1];
C = [1, 1, 1];
D = 0;

% Create state-space model
sys_ss = ss(A, B, C, D);

% Convert to transfer function
sys_tf = tf(sys_ss);
disp('Transfer Function:');
sys_tf
```

```
Transfer Function:

sys_tf =

     2 s^2 + 17 s + 35
  -----------------------
  s^3 + 11 s^2 + 39 s + 48

Continuous-time transfer function.
```

قسمت b)

Part (b)

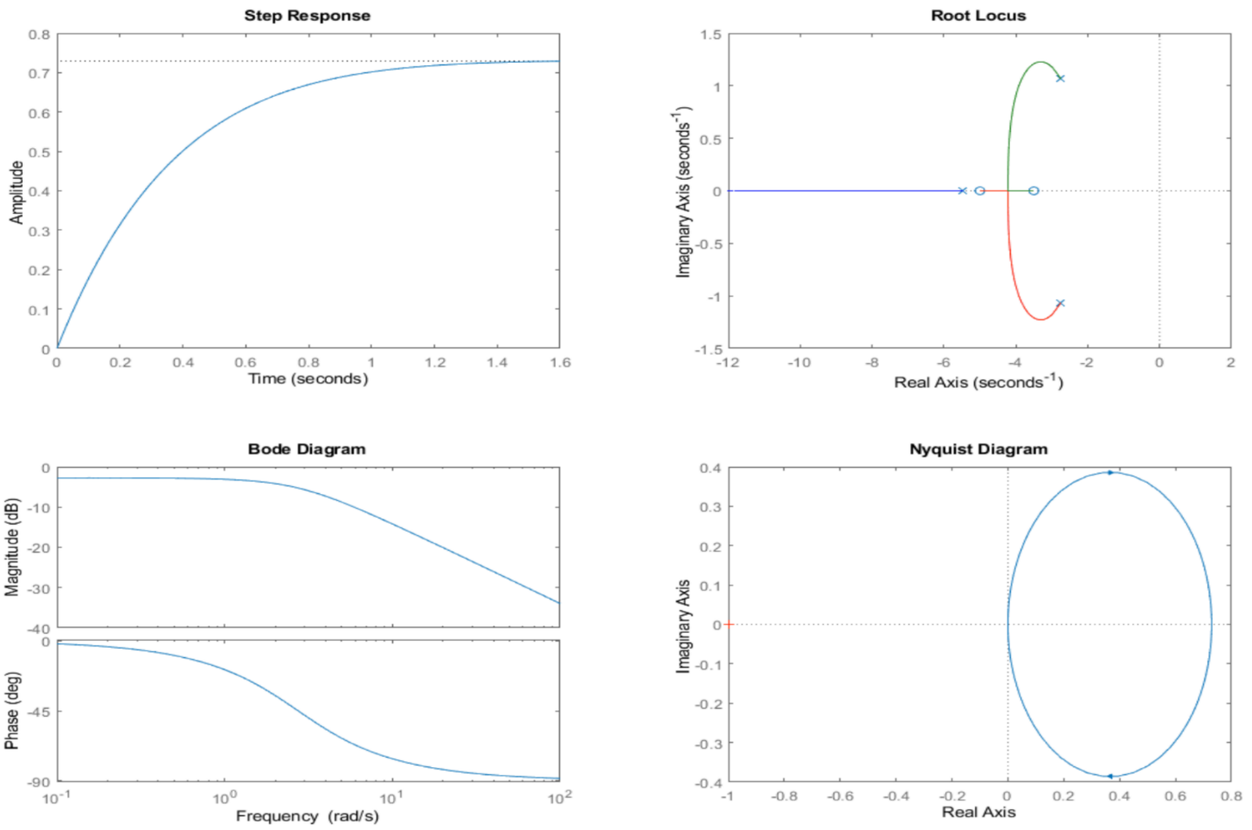
```
% Poles and zeros
poles = pole(sys_tf);
zeros = zero(sys_tf);
gain = dcgain(sys_tf); % DC gain of the system

disp('Poles:');
disp(poles);
disp('Zeros:');
disp(zeros);
disp('Gain:');
disp(gain);
```

```
Poles:
  -5.4856 + 0.0000i
  -2.7572 + 1.0715i
  -2.7572 - 1.0715i

Zeros:
   -5.0000
   -3.5000

Gain:
    0.7292
```

آز کنترل
تمرین سری اول

قسمت c)

```matlab
% Part (c)

% Step response
figure;
step(sys_tf);
title('Step Response');

% Root locus
figure;
rlocus(sys_tf);
title('Root Locus');

% Bode plot
figure;
bode(sys_tf);
title('Bode Diagram');

% Nyquist plot
figure;
nyquist(sys_tf);
title('Nyquist Diagram');
```

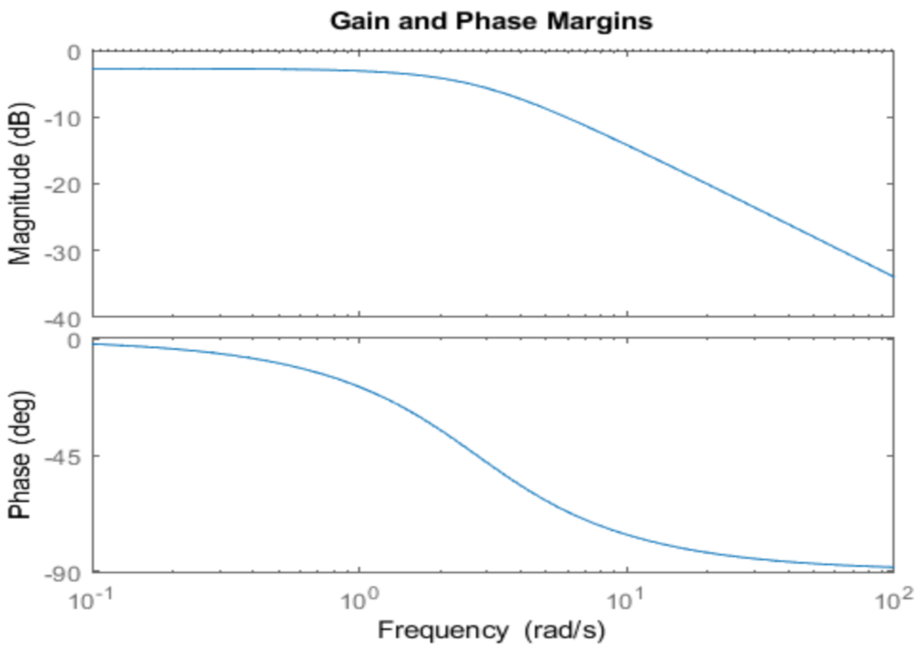


آز کنترل
تمرین سری اول

قسمت d)

```
% Part (d)

% Calculate margins explicitly using the margin command
[gm, pm, Wcg, Wcp] = margin(sys_tf);

if isinf(gm)
    disp('Gain Margin: Infinite (system is stable for all positive gains)');
else
    disp(['Gain Margin (dB): ', num2str(20*log10(gm))]);
end

if isinf(pm)
    disp('Phase Margin: Infinite (system is stable for all positive gains)');
else
    disp(['Phase Margin (degrees): ', num2str(pm)]);
end

if isnan(Wcg)
    disp('Gain Crossover Frequency: Undefined (system does not cross 0 dB)');
else
    disp(['Gain Crossover Frequency (rad/s): ', num2str(Wcg)]);
end

if isnan(Wcp)
    disp('Phase Crossover Frequency: Undefined (system does not reach -180° phase)');
else
    disp(['Phase Crossover Frequency (rad/s): ', num2str(Wcp)]);
end

% Plot margin diagram
figure;
margin(sys_tf);
title('Gain and Phase Margins');
```

```
Gain Margin: Infinite (system is stable for all positive gains)
Phase Margin: Infinite (system is stable for all positive gains)
Gain Crossover Frequency: Undefined (system does not cross 0 dB)
Phase Crossover Frequency: Undefined (system does not reach -180° phase)
```



آز کنترل
تمرین سری اول

همانطور که در بالا دید و در داخل کد نیز توضیح داده شده، مقادیر مارجین و کراس اور مقادیر خاص و با معنایی هستند.

در روش‌های مختلف برای تعیین حاشیه بهره و حاشیه فاز، از نمودارهای ریشه، بود، و نایکوئیست به عنوان ابزارهای تحلیلی استفاده می‌کنیم تا پایداری سیستم را در برابر تغییرات بهره بررسی کنیم.

1. روش نمودار ریشه (Root Locus): در این روش، مکان هندسی ریشه‌ها را نسبت به تغییرات پارامتر بهره رسم می‌کنیم. برای تعیین حاشیه بهره، نقطه‌ای که نمودار از محور موهومی عبور می‌کند را پیدا کرده و بهره بحرانی در این نقطه را محاسبه می‌کنیم. در این سیستم، چون نمودار مکان ریشه از محور موهومی عبور نمی‌کند، سیستم پایدار باقی می‌ماند، حتی با افزایش بهره. این نشان می‌دهد که حاشیه بهره برای سیستم بی‌نهایت است و سیستم در برابر تغییرات بهره بسیار مقاوم است.

2. روش نمودار بود (Bode Plot): در این روش، فرکانسی که در آن فاز سیستم به ۱۸۰- درجه می‌رسد را پیدا می‌کنیم. در این فرکانس، مقدار دامنه اندازه‌گیری شده و برای محاسبه حاشیه بهره استفاده می‌شود. در این سیستم، چون فاز نمودار بود هیچگاه به ۱۸۰- درجه نمی‌رسد، حاشیه فاز و حاشیه بهره هر دو بی‌نهایت در نظر گرفته می‌شوند. این نیز نشانگر پایداری سیستم حتی در شرایط بهره‌های بالا است.

3. روش نمودار نایکوئیست (Nyquist Plot): در این روش، توجه به نقطه‌ای است که نمودار مسیر خود را در نزدیکی نقطه \(-1\) در صفحه مختلط قرار می‌دهد. اگر نمودار نایکوئیست از این نقطه عبور کند یا آن را دور بزند، ممکن است سیستم به سمت ناپایداری برود. در این سیستم، نمودار نایکوئیست هیچگاه به این نقطه نزدیک نمی‌شود، بنابراین سیستم پایدار بوده و دارای حاشیه بهره بی‌نهایت است.

4. استفاده از دستور `margin` در متلب: با استفاده از دستور `margin` در متلب می‌توانیم مقادیر حاشیه بهره و حاشیه فاز را محاسبه کنیم. در این مورد خاص، خروجی دستور نشان می‌دهد که حاشیه بهره و حاشیه فاز هر دو بی‌نهایت هستند و فرکانس‌های عبور بهره و فاز به‌صورت `NaN` (تعریف‌نشده) نمایش داده می‌شوند. این نتایج حاکی از آن است که سیستم در مقابل تغییرات گسترده بهره پایدار می‌ماند.

نتیجه‌گیری از تمامی این روشها و تحلیل‌ها نشان می‌دهد که سیستم پایداری نامحدود دارد و تغییرات بهره تاثیری بر پایداری آن نخواهد گذاشت.

قسمت e)

```
% Part (e)

% Step response information
info = stepinfo(sys_tf);

disp('Step Response Properties:');
disp(['Rise Time: ', num2str(info.RiseTime)]);
disp(['Settling Time: ', num2str(info.SettlingTime)]);
disp(['Overshoot: ', num2str(info.Overshoot)]);
disp(['Peak: ', num2str(info.Peak)]);
disp(['Peak Time: ', num2str(info.PeakTime)]);
```

```
Step Response Properties:
Rise Time: 0.70611
Settling Time: 1.1442
Overshoot: 0.15356
Peak: 0.73029
Peak Time: 1.9709
```

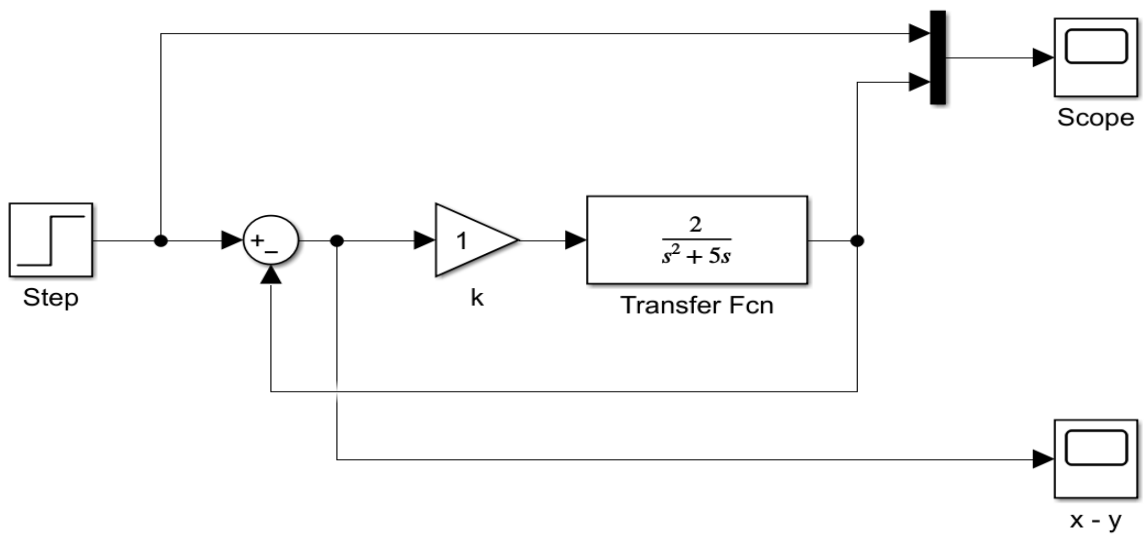
در این بخش، هدف بررسی مشخصات پاسخ پله (Step Response) سیستم است. این مشخصات شامل موارد زیر هستند که هر کدام جنبه خاصی از رفتار پایداری و سرعت واکنش سیستم به یک ورودی پله‌ای را نشان می‌دهند:

1. **درصد فراجهش (Percentage Overshoot) :** این مشخصه، میزان تجاوز مقدار پاسخ از مقدار نهایی را نشان می‌دهد و به صورت درصد بیان می‌شود. درصد فراجهش بیانگر میزان نوسان اولیه سیستم پیش از رسیدن به حالت پایدار است. اگر این مقدار زیاد باشد، سیستم بیش از حد از مقدار نهایی عبور کرده و پس از آن به مقدار نهایی بازمی‌گردد.

2. **زمان صعود(Rise Time) :** زمان صعود نشان‌دهنده مدت زمانی است که طول می‌کشد تا پاسخ سیستم از مقدار اولیه به درصد مشخصی از مقدار نهایی برسد (معمولاً از 10% به 90% مقدار نهایی). این مشخصه نشان می‌دهد که سیستم چقدر سریع می‌تواند به مقدار نهایی خود نزدیک شود. زمان صعود کمتر، بیانگر واکنش سریع‌تر سیستم است.
3. **زمان نشست(Settling Time) :** زمان نشست، مدت زمانی است که طول می‌کشد تا پاسخ سیستم در یک محدوده مشخص از مقدار نهایی (معمولاً 2% یا 5%) باقی بماند و نوسانات بیشتری نداشته باشد. این مشخصه به ثبات سیستم و مدت زمان لازم برای رسیدن به وضعیت پایدار اشاره دارد.
4. **خطای ماندگار (Steady-State Error) :** این خطا نشان‌دهنده اختلاف بین مقدار نهایی خروجی سیستم و مقدار مطلوب یا ورودی ایده‌آل است و نشان می‌دهد که آیا سیستم در بلندمدت می‌تواند به مقدار مطلوب نزدیک شود یا خیر.

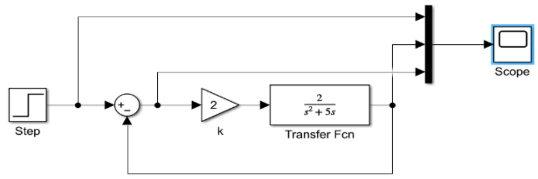
بخش 3.1) کنترل کننده تناسبی

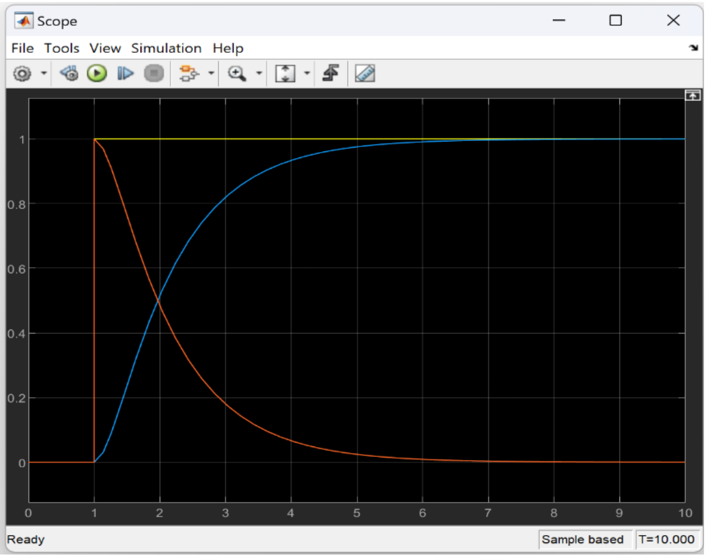
قسمت a) دیاگرام زیر در سیمولینک پیاده شد و ضمیمه شده است.

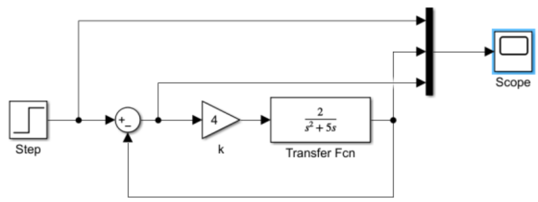


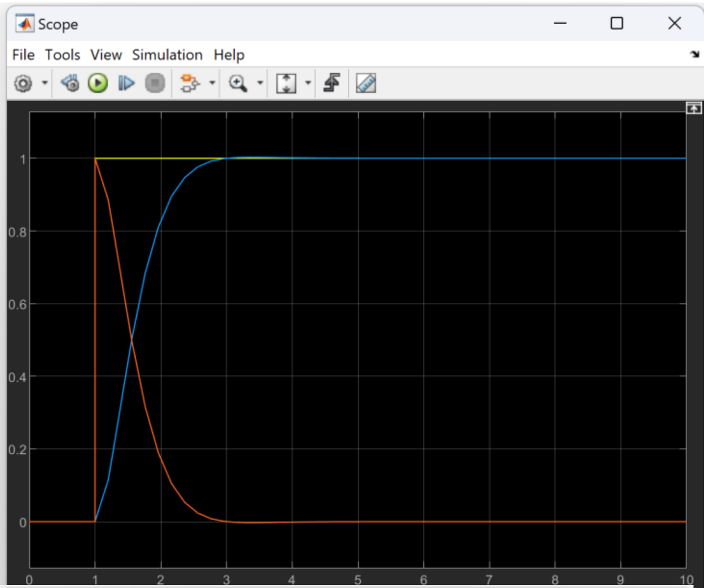
آز کنترل
تمرین سری اول

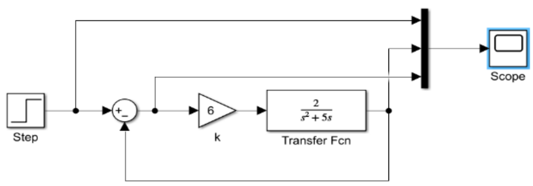
قسمت b)

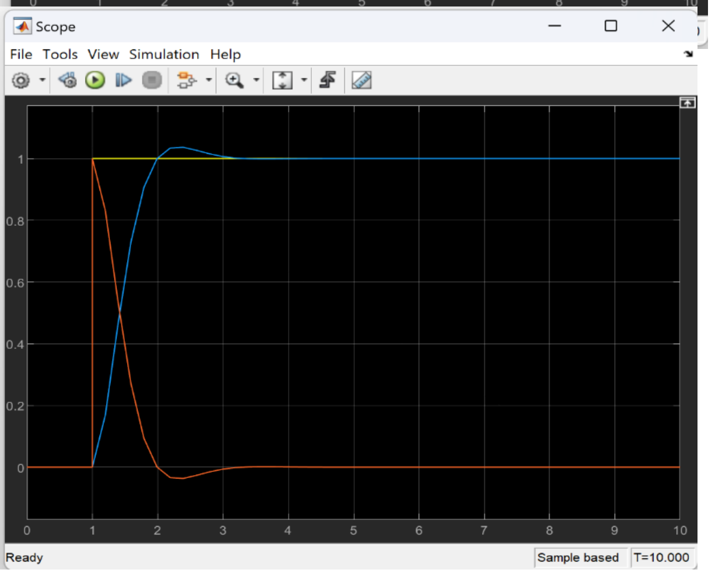


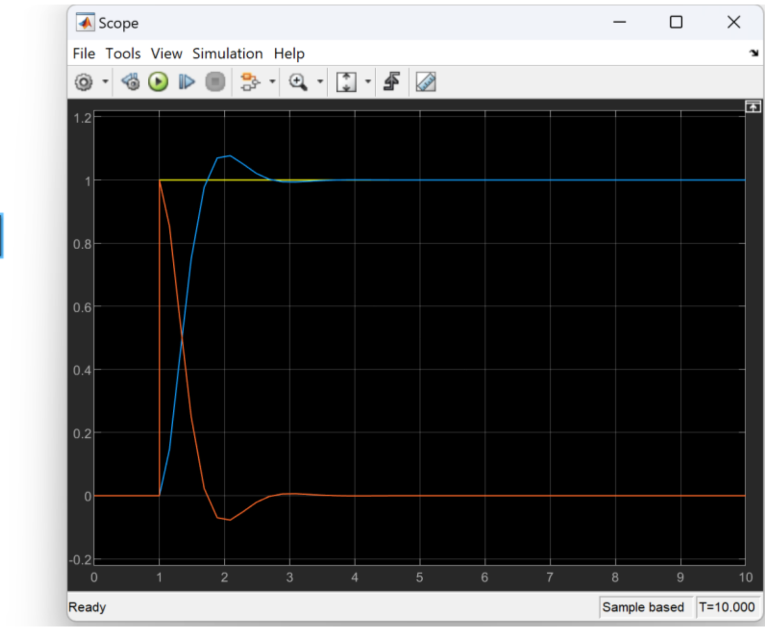


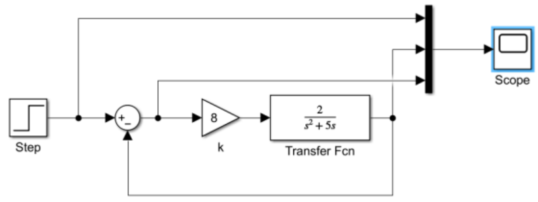


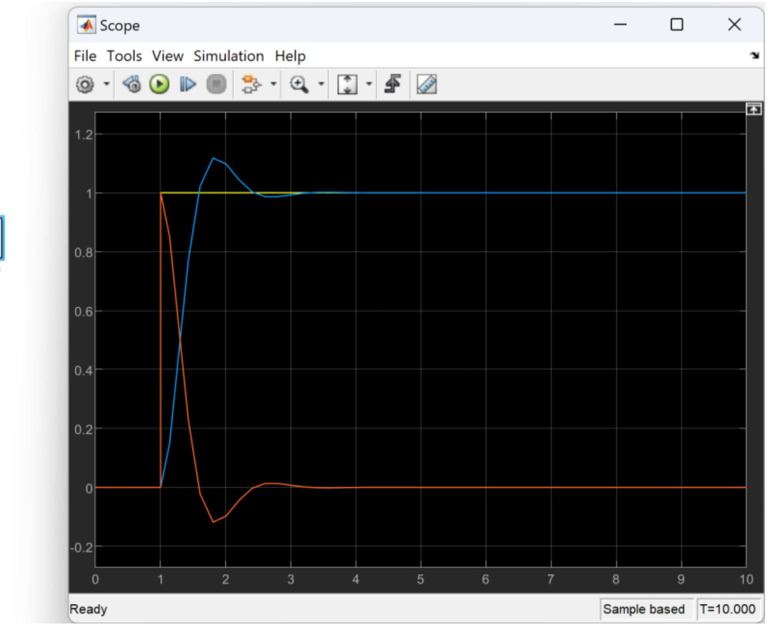


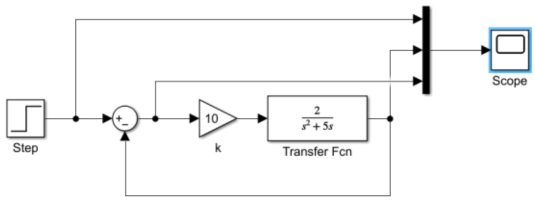






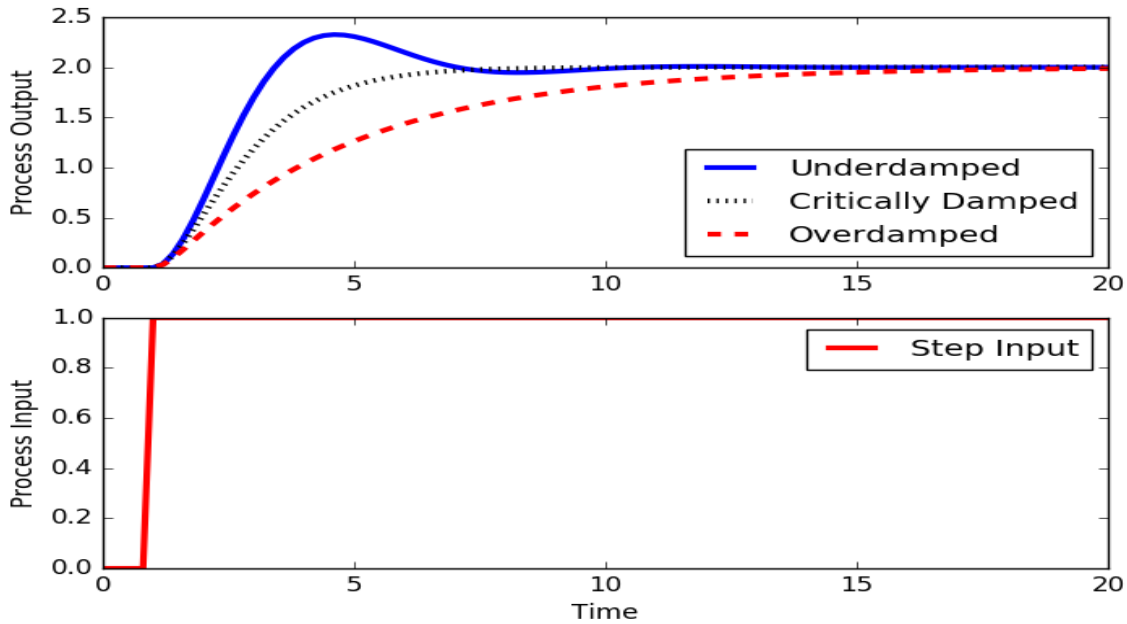






مشاهده می‌کنیم که سیستم مُد های مختلفی عوض می‌کند. ابتدا پاسخ پله سیستم، میرای شدید (Overdamp) است، اما با افزایش مقدار k ابتدا به حالت میرای بحرانی (Critically Damped) رفته و نهایتاً میرای ضعیف (Underdamped) می‌شود که سبب می‌شود سیستم در زمان طولانی تری به حالت پایدار خود برسد. در حالت آخر، بالازدگی ها و اورشوت ها کاملاً قابل مشاهده است. در حالاتی که دیدیم همگی خروجی‌ها پایدار هستند.

شکل کلی پاسخ هایی که در اینجا نیز دیده شدند:

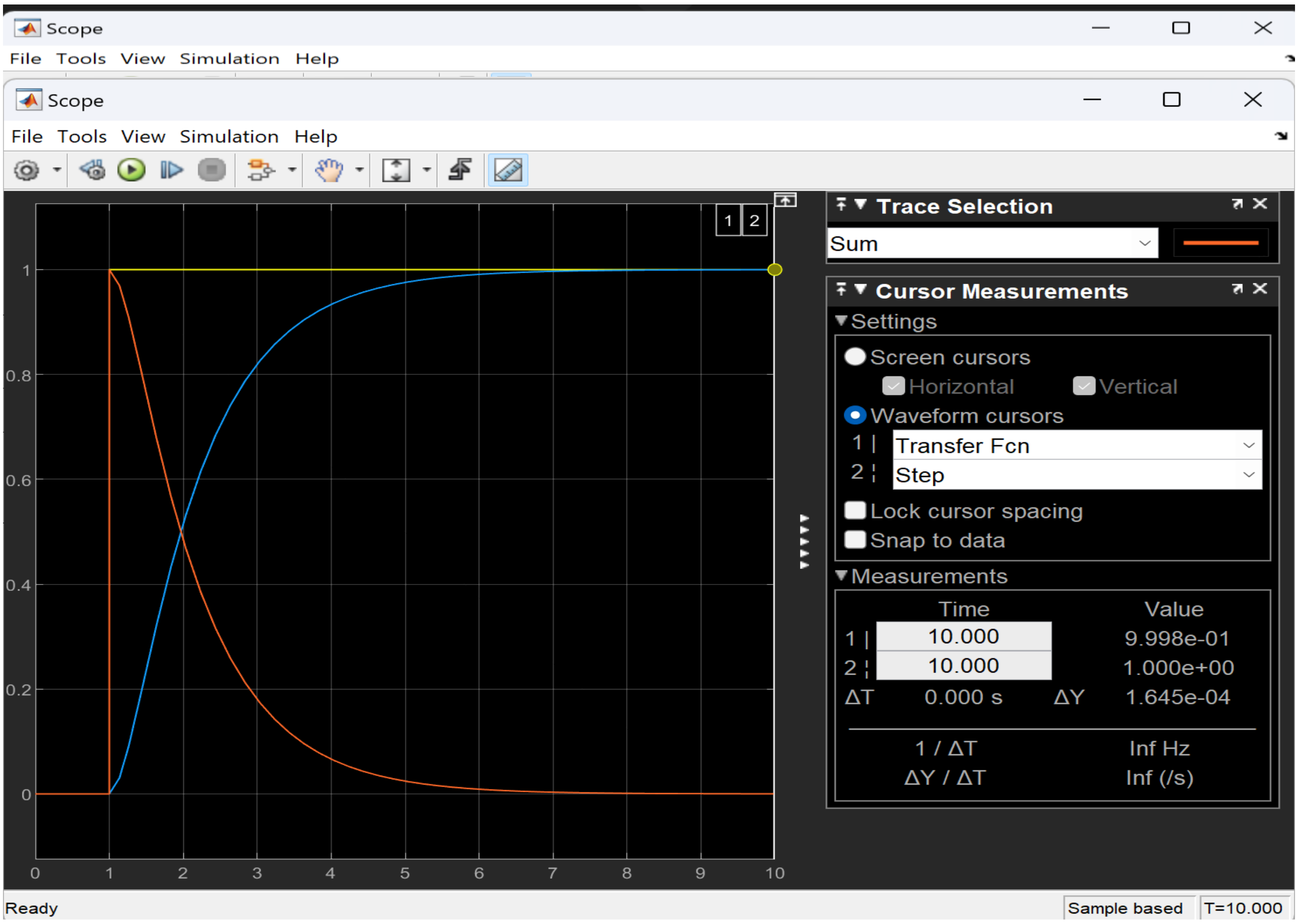


قسمت c) ثانیه دهم را تقریبی از رفتار سیستم در بی‌نهایت فرض می‌کنیم.

$$E = |x - y(t = 1)|$$

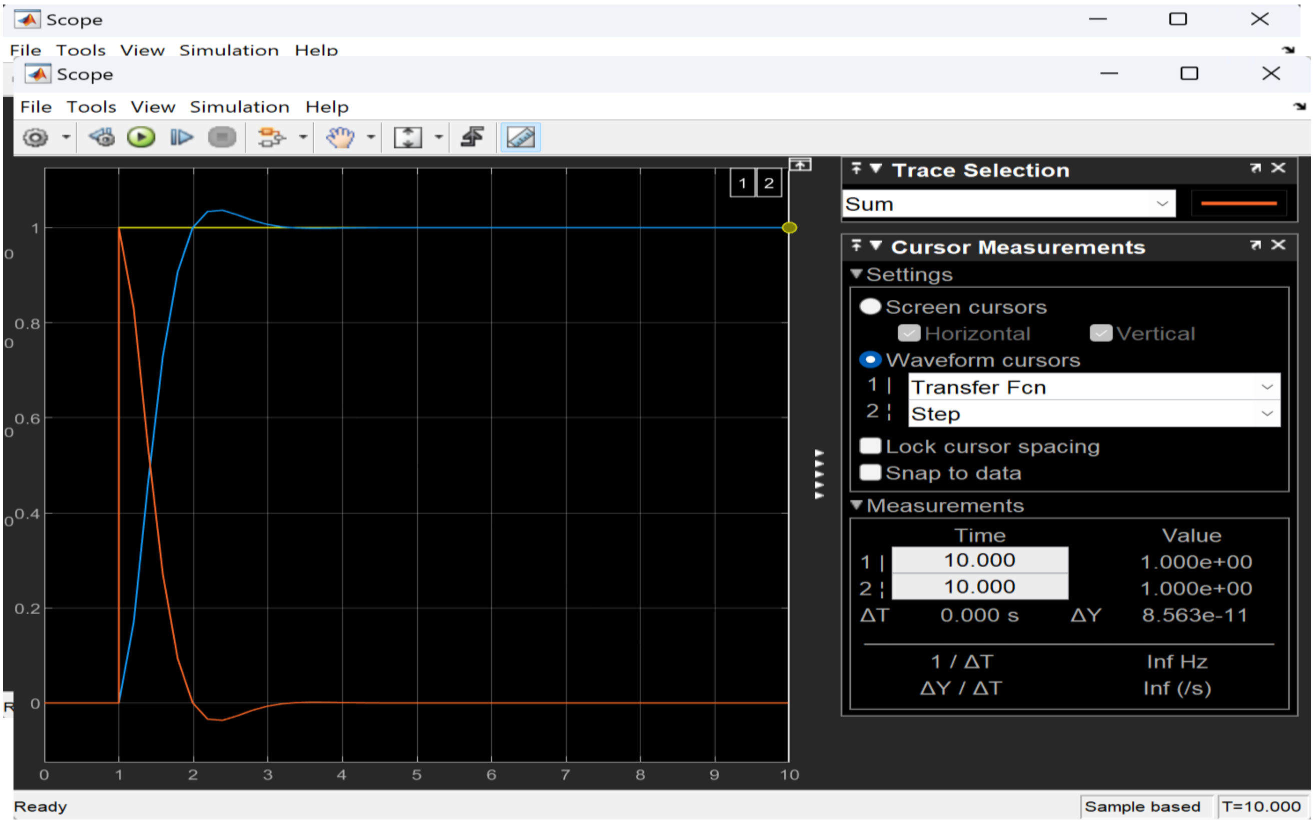
آز کنترل
تمرین سری اول

به ازای k=1:



به ازای k=2:

به ازای k=4:

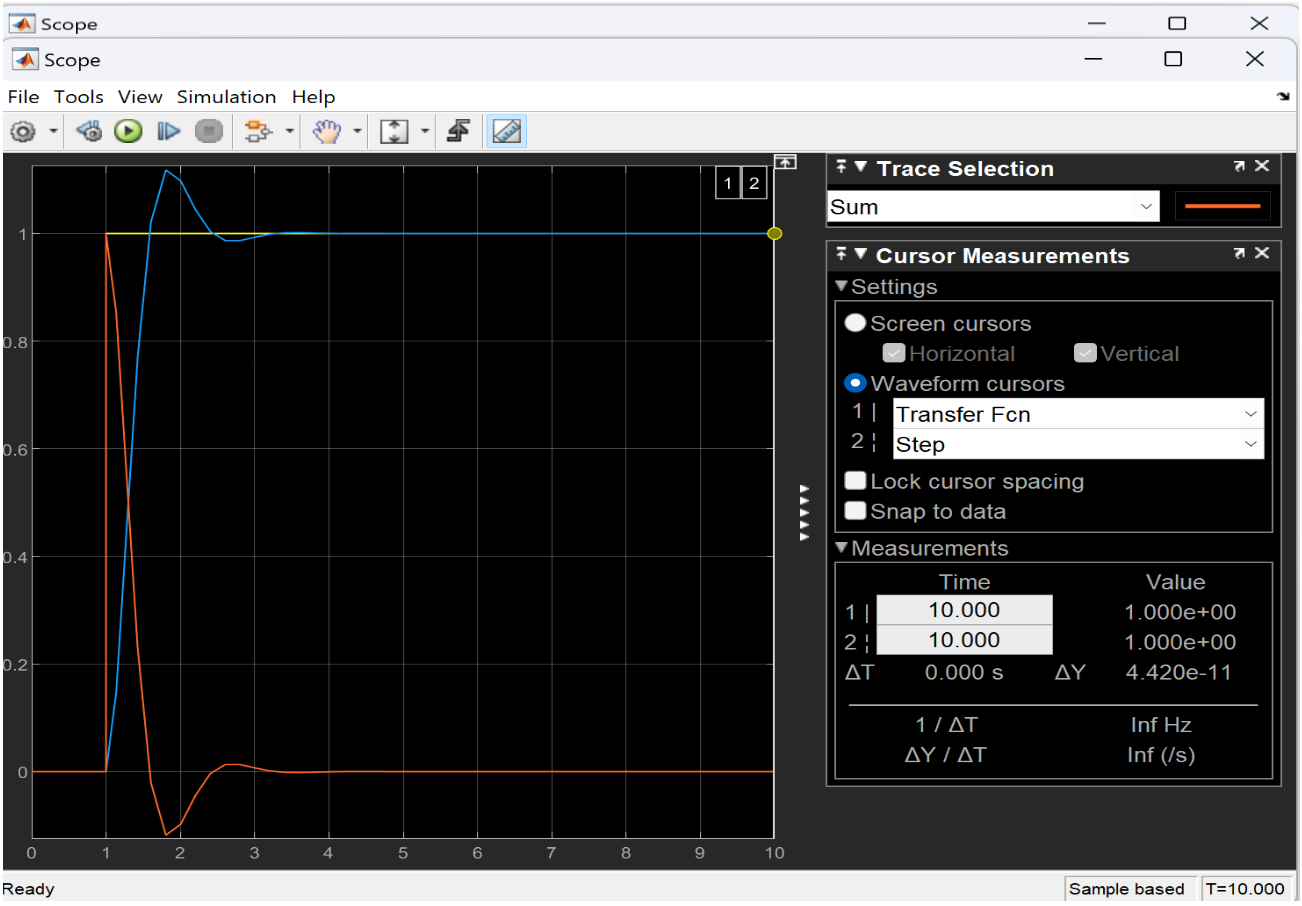


به ازای k = 6:

آز کنترل
تمرین سری اول

به ازای k=8:

به ازای k = 10:

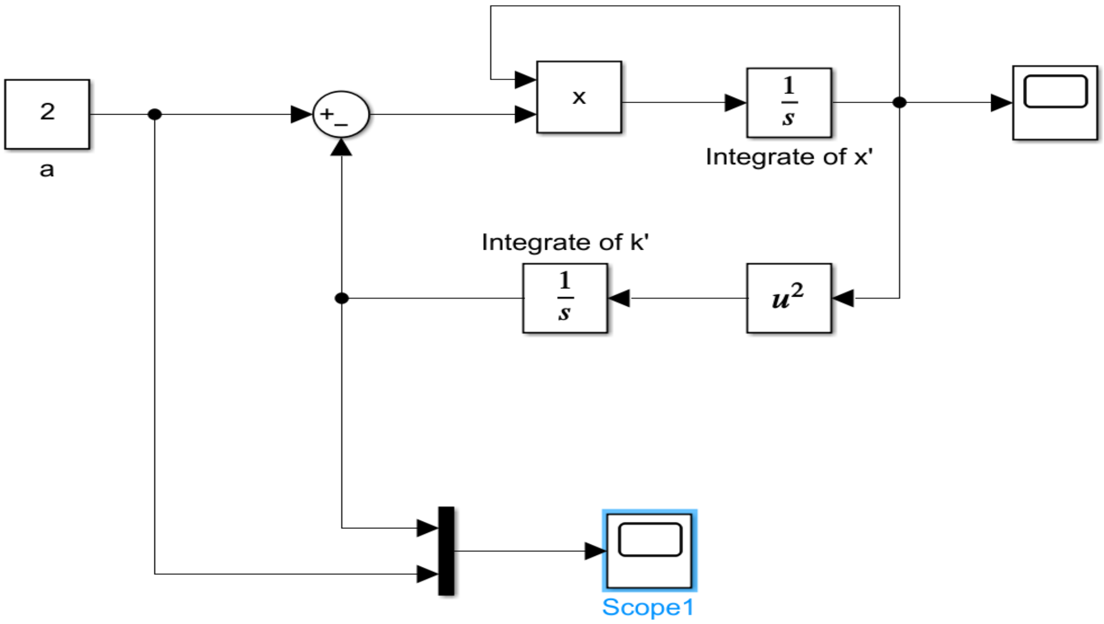


همانطور که مشاهده شد، خطای حالت ماندگار از $2.14e-02$ به $4.42e-11$ رسید که یعنی این خطا به ازای مقادیر کم k قابل بررسی و نهایتاً به ازای مقادیر بزرگ قابل صرف نظر است.

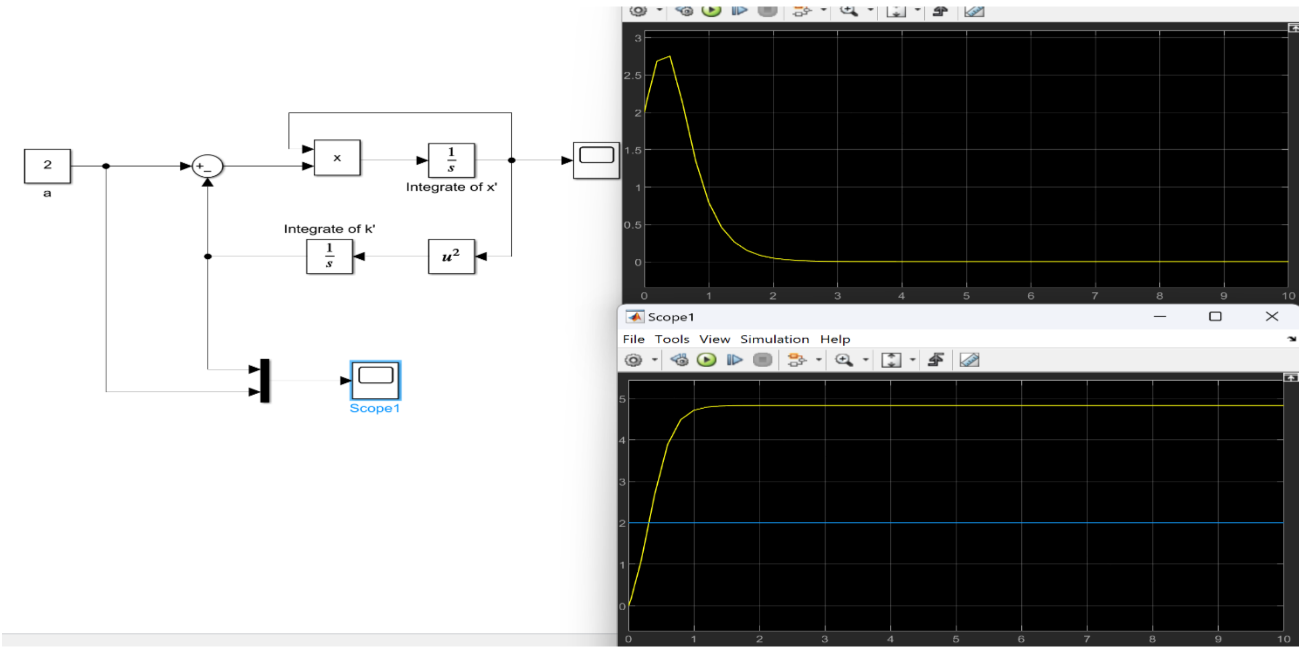
بخش 3.2) کنترل کننده تطبیقی

قسمت a)

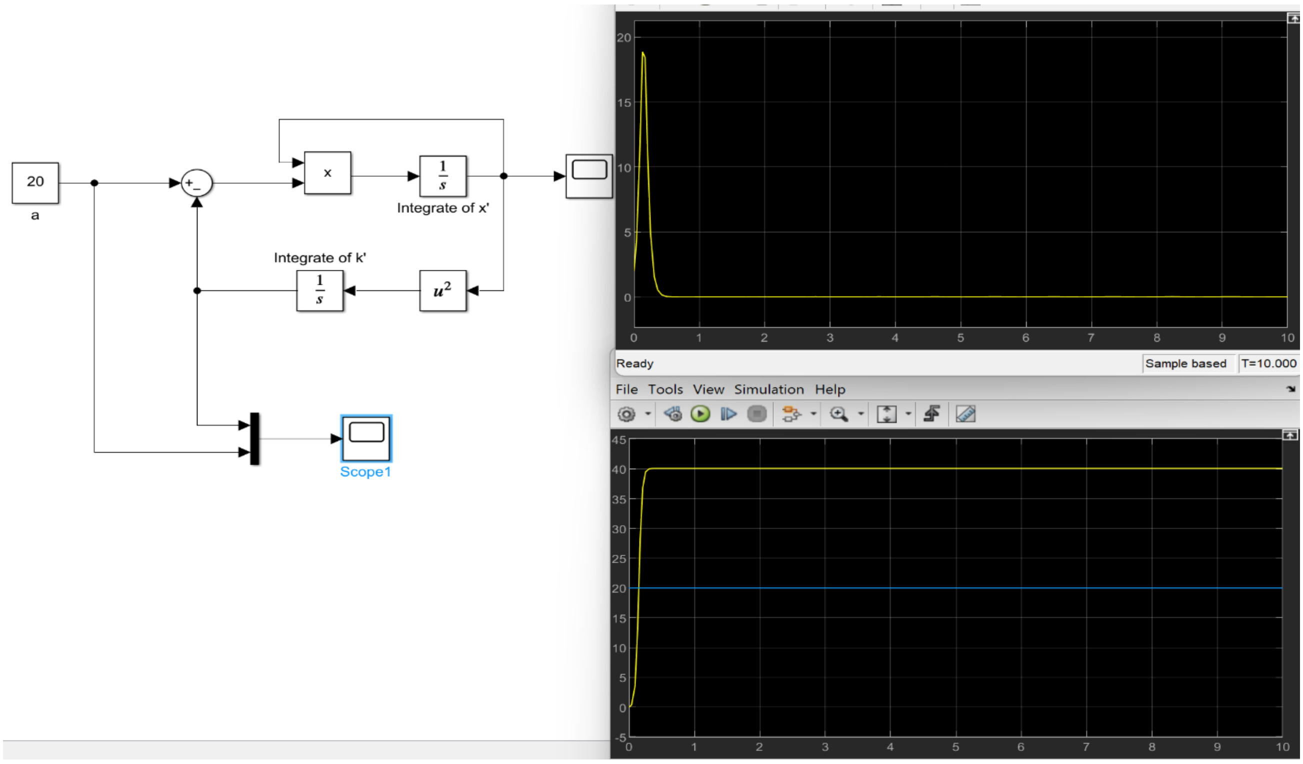
دیاگرام زیر در سیمولینک طراحی و بررسی شد و فایل آن نیز ضمیمه است:



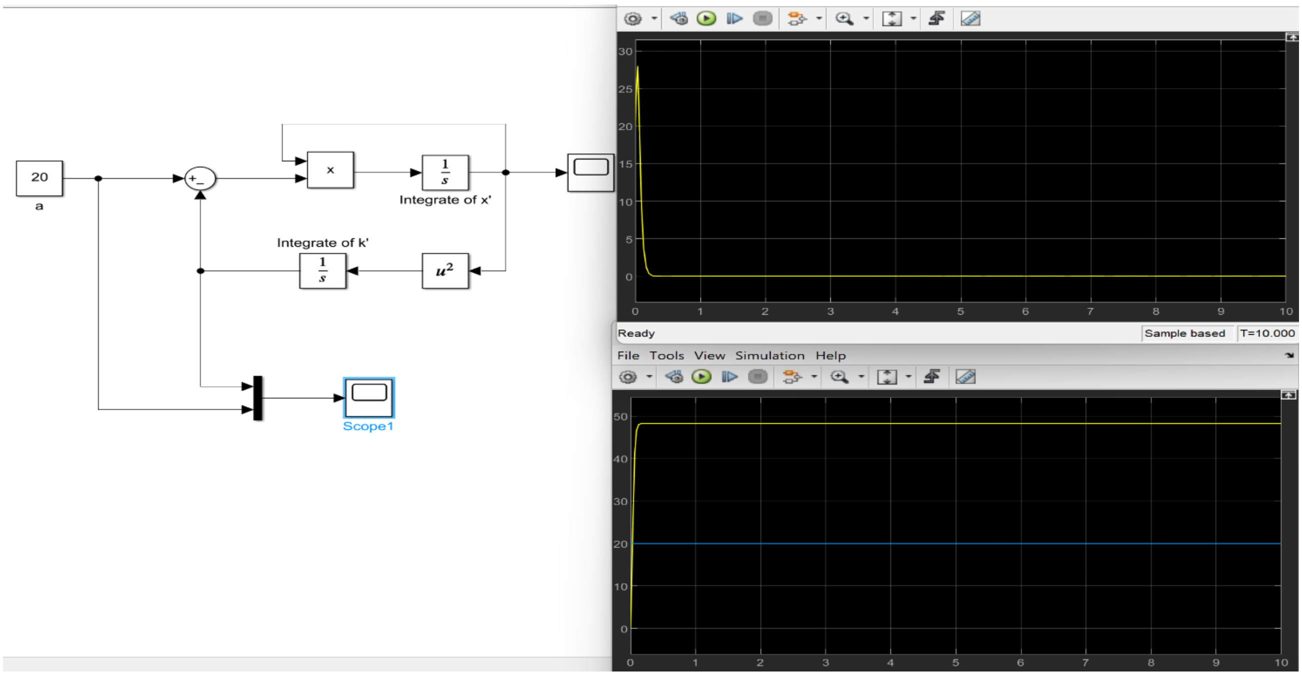
به ازای a = 2 و x(0)=2 :



بـه ازای $a = 20$ و $x(0)=2$:



بـه ازای $a = 20$ و $x(0)=20$:



قسمت b)

آز کنترل
تمرین سری اول

کنترل‌کننده تطبیقی با تنظیم دینامیکی بهره ($k(t)$) بر اساس وضعیت فعلی سیستم، پارامتر ناشناخته (a)

را جبران می‌کند. در سیستم مورد نظر، معادله حالت با کنترل‌کننده تطبیقی به‌صورت زیر است:

$$\dot{x}(t) = (a - k(t))x(t)$$

برای پایداری سیستم، باید ($a - k(t) < 0$) باشد که این شرط با انتخاب ($k(t) > a$) برآورده می‌شود. اما چون (a) ناشناخته است، کنترل‌کننده با افزایش تدریجی ($k(t)$) نسبت به خروجی سیستم ($x(t)$) این پارامتر را جبران می‌کند. قاعده تطبیق بهره به‌صورت زیر است:

$$\dot{k}(t) = x^2(t), \quad k(0) = 0$$

از آنجا که ($x^2(t)$) همیشه غیرمنفی است، این قاعده موجب می‌شود که ($k(t)$) یا افزایش یابد یا ثابت بماند، بسته به مقدار ($x(t)$):

- وقتی ($x(t)$) بزرگ باشد، ($\dot{k}(t) = x^2(t)$) نیز بزرگ می‌شود و باعث افزایش سریع ($k(t)$) می‌شود. این افزایش در ($k(t)$) به جبران بی‌ثباتی کمک می‌کند و اطمینان حاصل می‌کند که ($a - k(t)$) منفی شده و سیستم پایدار می‌شود.

- وقتی ($x(t)$) به صفر نزدیک می‌شود، ($\dot{k}(t)$) نیز کندتر می‌شود و به ($k(t)$) اجازه می‌دهد تا به مقداری ثابت برسد که سیستم را پایدار نگه می‌دارد. وقتی ($x(t)$) برابر با صفر است، سیستم به تعادل رسیده و بهره ($k(t)$) افزایش نمی‌یابد.

این تنظیم تطبیقی مبتنی بر بازخورد، به کنترل‌کننده اجازه می‌دهد تا در لحظه، به مقدار ناشناخته یا متغیر (a) پاسخ دهد و بدون نیاز به آگاهی دقیق از دینامیک سیستم، پایداری را به‌دست آورد. نتایج قسمت a نیز این مهم را تایید می‌کنند.