آموزشکده فنی و کشاورزی فسا

واحد آزمایشگاه ها و کارگاه ها

آزمایشگاه کنترل خطی

آموزش نرم افزار متلب و SIMULINK

تنظیم

محمد علی بردبار

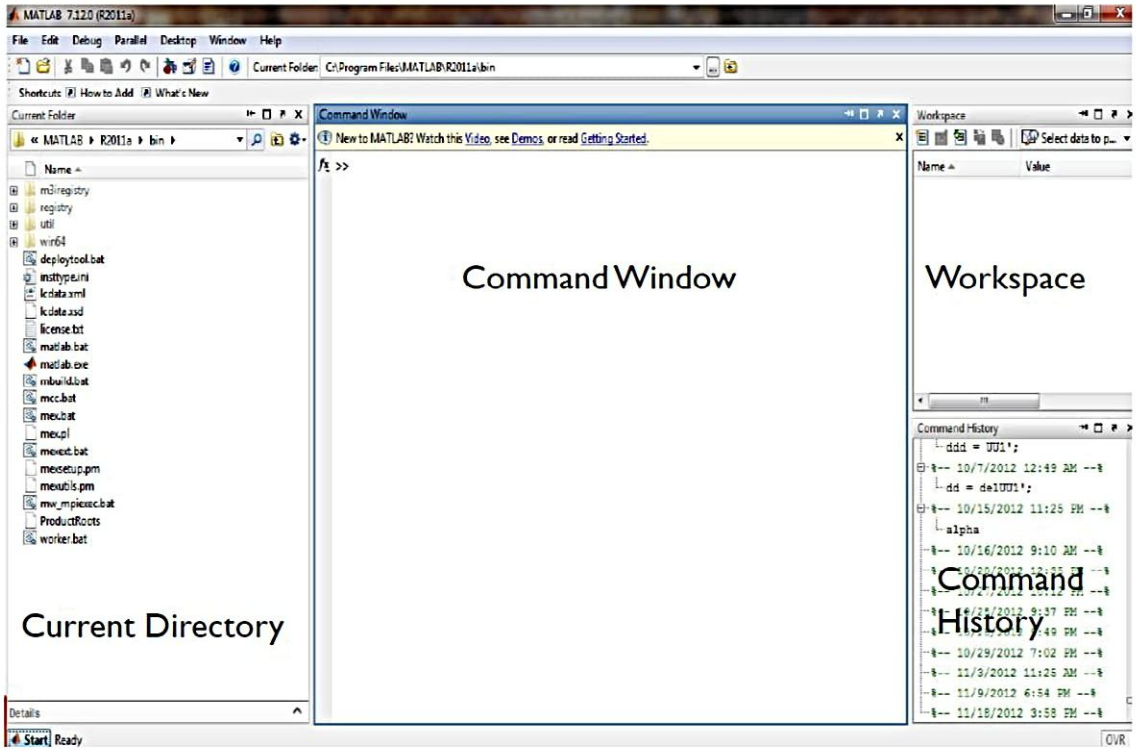
بخش اول

آشنایی با نرم افزار MATLAB

MATLAB

- MATLAB مخفف MATrix LABoratory است.
- MATLAB محاسبات ریاضی ، رسم نمودار و یک زبان قدرتمند را برای ارائه محیطی انعطاف پذیر ، به منظور انجام محاسبات فنی و دقیق ، با یکدیگر در هم آمیخته است.
- **کاربردهای عمده MATLAB عبارتند از :**
  - ✓ محاسبات عددی و بهبود الگوریتم ها
  - ✓ محاسبات سمبولیک (با توابع ریاضی سمبولیک درون ساختی MATLAB)
  - ✓ مدلسازی ، شبیه سازی و الگوسازی
  - ✓ جزیه و تحلیل اطلاعات و پردازش سیگنال
  - ✓ گرافیک و نمودارهای مهندسی و تصویر سازی های علمی
  - ✓ و ... .

محیط نرم افزار MATLAB



محیط نرم افزار MATLAB

- Command Window: برای وارد کردن دستورات و اطلاعات به منظورنمایش نمودارها و گراف ها مورد استفاده قرار می گیرد.

- Current Directory: مشاهدهٔ پوشه ها و m-file ها

- Workspace:

۱- مشاهده متغیرهای برنامه

۲- با کلیک روی هر متغیر در این پنجره می توان مقدار آن متغیر را در پنجره Array Editor مشاهده کرد.

- Command History: دستورات قبلی در آن ذخیره می شود.

مقدمات MATLAB

❖ واردکردن دستورات :

✓ برای اجرای هر دستور در MATLAB پس از تایپ آن در Command Window کلید Enter را فشار می دهیم.

✓دستورات MATLAB نسبت به حروف کوچک و بزرگ حساس اند. (case sensitive) .

✓برای تایپ دستورات تماماً از حروف کوچک استفاده می شود.

✓برای اجرای یک M- فایل (مثلاً M- فایلی با نام Project_۱.m)، کافی است نام آن فایل را بدون پسوند وارد نمائید(Project_۱).

❖صرف نظر کردن از انجام یک دستور :

✓به منظور صرف نظر کردن از انجام یک دستور (قطع یک دستور Abrot) و جلوگیری از ادامه اجرای آن می توانید از کلیدهای Ctrl+c استفاده می شود.

مقدمات MATLAB

❖نقطه ویرگول (Semicolon ;) :

✓ اگر در آخر یک دستور نقطه ویرگول (;) تایپ شود ، خروجی دستور نشان داده نخواهد شد.

✓ برای جداکردن سطر های یک ماتریس .

❖علامت «٪» :

✓ هنگامی که علامت ٪ در ابتدای یک خط تایپ شود ، آن خط بعنوان توضیحات برنامه در نظر گرفته می شود . پس از فشردن کلید Enter می بینید که آن خط اجرا نخواهد شد و به رنگ سبز در می آید.

❖ دستور clc :

✓ تایپ دستور *clc* و فشردن کلید Enter پنجره Command Window را پاک خواهد کرد. به محض اجرای دستور *clc* پنجره پاک شده نمایش داده خواهد شد.

مقدمات MATLAB

❖ **Help :**

- ✓ برای مشاهده لیست توابع موجود در MATLABو همچنین طرز کار آنها و مشاهدهٔ مثالهای متنوعی از کاربردهای آنها.
- ✓ برای یافتن توضیحات مربوط به یک موضوع خاص ، مثلاً تابعِ معکوس کردن ماتریس inverse، در خط فرمان دستور *help inv* را تایپ کنید.

❖ **عبارات و متغیرها :**

- ✓ متغیرها را به فرم کلی زیر تعریف می کنیم :

مقادیر= متغیر >>

علامت تساوی (=) *مقادیر* را به متغیر نسبت می دهد.این مقادیر می توانند گونه های متفاوت داشته باشند،ماتریس ، بردار ، عدد ، رشته ، کاراکتر و ....

مقدمات MATLAB

❖ **نامگذاری متغیرها :**

یک متغیر نامی است که از یک حرف یا ترکیبی از چندین حرف و رقم ساخته میشود.نام متغیرها میتواند تا ۶۳ کاراکتر طول داشته باشد (در MATLAB ۶.۰ تا ۳۱ کاراکتر).MATLAB به حروف کوچک و بزرگ حساس است. برای مثال *xx*و*xX,Xx,XX* نام چهار متغیر متفاوت است.توجه داشته باشید ، نباید از نام توابع MATLABبرای متغیرها استفاده کرد. برای مثال از نامهایی چون:sqrt,exp,cos,sin و ... نباید استفاده شود. به محض استفاده از نام یک تابع برای تعریف یک متغیر ، دیگر قادر به استفاده از آن تابع نخواهید بود.

عملیات محاسباتی

عمل ریاضی	نماد	مثال
جمع	+	6+3=9
تفریق	-	6-3=3
ضرب	*	6*3=18
تقسیم راست (معمولی)	/	6/3=2
تقسیم چپ	\	6\3=3/6=1/2
توان	^	6^3=216

توابع مقدماتی ریاضی در MATLAB

جدول ۳-۲: برخی از توابع ریاضی معمول

تابع	توضیحات
abs(x)	قدر مطلق (اندازه) x را محاسبه می‌کند.
sqrt(x)	ریشهٔ دوم x را محاسبه می‌کند.
round(x)	مقدارx را به نزدیکترین عدد صحیح گرد می‌کند.
fix(x)	قسمت اعشاری عدد x را حذف می‌کند.
floor(x)	مقدارx را به نزدیکترین عدد صحیح کوچکتر گرد می‌کند (روند به پایین)
ceil(x)	مقدارx را به نزدیکترین عدد صحیح بزرگتر گرد می‌کند(روند به بالا)
sign(x)	تابع علامت (این دستور برای x های کوچکتر از صفر عدد«۱-»، برای مقادیر x بزرگتر از صفر عدد«۱» و برای x=0 مقدار صفر را برمی‌گرداند.
rem(x,y)	باقیمانده تقسیم x بر y را برمی‌گرداند برای مثال rem(25,4)=1 و rem(100,21)=16
exp(x)	مقدار e^x را محاسبه می‌کند (e پایه لگاریتم طبیعی است، تقریباً برابر با۲,۷۱۸۴۲۸۲)
log(x)	مقدارLn(x) را محاسه می‌کند، لگاریتم طبیعی x در پایه e
log10(x)	مقدارLog10(x) را محاسبه می‌کند، لگاریتم معمولی x در پایهٔ ۱۰

توابع مقدماتی ریاضی در MATLAB

جدول ۴-۲ توابع نمایی

تابع	توضیحات
exp(x)	نماییِ(e^x)
log(x)	لگاریتم طبیعی
log10(x)	لگاریتم پایهٔ ۱۰
sqrt(x)	ریشهٔ دوم

جدول۵-۲: توابع گرد کردن

تابع	توضیحات	مثال
round(x)	گرد کردن به نزدیک‌ترین عدد صحیح	>> round(20/6) ans = 3
fix(x)	گرد کردن به سمت صفر	>> fix(13/6) ans = 2
ceil(x)	گرد کردن به سمت بی‌نهایت	>> ceil(13/5) ans = 3
floor(x)	گرد کردن به سمت منفی بی‌نهایت	>> floor(– 10/4) ans = – 3
rem(x,y)	باقیمانده تقسیم x بر y را برمی‌گرداند	>> rem(14,3) ans = 2
sign(x)	تابع علامت	>> sign(7) ans = 1

توابع مقدماتی ریاضی در MATLAB

جدول ۶-۲: توابع مثلثاتی و هایپربولیک

تابع	توضیحات
sin(x)	سینوس x را محاسبه می‌کند(x برحسب رادیان)
cos(x)	کسینوس x را محاسبه می‌کند(x برحسب رادیان)
tan(x)	تانژانتx را محاسبه می‌کند(x برحسب رادیان)
asin(x)	آرک سینوس x را محاسبه می‌کند.x باید عددی بین ۱– و ۱ باشد. این تابع زاویه‌ای برحسب رادیان بین π/۲– و π/۲ برمی‌گرداند
acos(x)	آرک کسینوس x را محاسبه می‌کند.x باید عددی بین ۱– و ۱ باشد. این تابع زاویه‌ای برحسب رادیان بین ۰ و π برمی‌گرداند
atan(x)	آرک تانژانت x را محاسبه می‌کند. این تابع زاویه‌ای برحسب رادیان بین π/۲– و π/۲ بر می‌گرداند.
atan2(y,x)	آرک تانژانتy/x را محاسبه می‌کند. این تابع با توجه به علامت‌های x وy زاویه‌ای برحسب رادیان بین π و π– برمی‌گرداند.
sinh(x)	سینوس هایپربولیک x را محاسبه می‌کند که برابر است با $(e^x - e^{-x})/2$
cosh(x)	کسینوس هایپربولیک x را محاسبه می‌کند که برابر است با $(e^x + e^{-x})/2$

توابع مقدماتی ریاضی در MATLAB

جدول ۷-۲: عملیات ریاضی روی اعداد مختلط

عملیات	نتیجه
$c_1 + c_2$	$(a_1 + a_2) + i(b_1 + b_2)$
$c_1 + c_2$	$(a_1 - a_2) + i(b_1 - b_2)$
$c_1 \bullet c_2$	$(a_1a_2 - b_1b_2) + i(a_1b_2 - a_2b_1)$
c_1/c_2	$\left(\frac{a_1a_2 + b_1b_2}{a_2^2 + b_2^2}\right) + i\left(\frac{a_2b_1 - b_2a_1}{a_2^2 + b_2^2}\right)$
$\|c_1\|$	$\sqrt{(a_1^2 + b_1^2)}$ (اندازهٔ c_1)
c_1^*	$a_1 - ib_1$ (مزدوج c_1)
فرض کنید $c_1 = a_1 + b_1$ و $c_2 = a_2+b_2$	

توابع مقدماتی ریاضی در MATLAB

جدول ۸-۲: توابع مربوط به اعداد مختلط

تابع	توضیحات
conj(x)	مزدوج مختلط،عددمختلط x را محاسبه می‌کند. بنابراین اگر x=a+ib باشد آنگاه conj(x) برابر است با a-ib
real(x)	قسمت حقیقیِ عدد مختلط x را محاسبه می‌کند.
imag(x)	قسمت موهومی عدد مختلط x را محاسبه می‌کند.
abs(x)	این تابع قدرمطلق اندازهٔ عدد مختلط x را برمی‌گرداند. $x = a+ib \rightarrow abs(x)=\|x\|=\sqrt{a^2+b^2}$
angle(x)	این تابع زاویهٔ عدد مختلط x را محاسبه می‌کند که زاویه‌ای بین π و $-\pi$ می‌باشد.

متغیرهای از پیش تعریف شده

جدول ۹-۲: متغیرهای از پیش تعریف شده

متغیر از پیش تعریف شده	توضیحات
ans	حاصل محاسبات انجام شده توسط یک عبارت یا دستور را در صورتی‌که در متغیری ذخیره نشده باشد نشان می‌دهد.
pi	عدد π
eps	معرف یک مقدار بسیار کوچکِ بزرگتر از صفر.
inf	بی‌نهایت را نشان می‌دهد که مثلاً به عنوان نتیجهٔ تقسیم بر صفر نمایش داده می‌شود. همراه با این متغیر یک پیغام هشدار نیز نمایش داده می‌شود.
i	نشان‌دهندهٔ مقدار $\sqrt{(-1)}$، که عبارت است از:0+1.000i (متغیر اعداد مختلط)
j	شبیه i عمل می‌کند.
NaN	بیانگر مبهم بودن است. در جایی می‌آید که عددی (یا پاسخی) وجود نداشته باشد. معمولاً در نتیجهٔ یک عبارت یا دستورِ تعریف نشده، اتفاق می‌افتد. مثل حالت تقسیم صفر بر صفر 0/0
clock	زمان جاری سیستم را در یک بردار سطریِ ۶ آرایه‌ای شامل سال، ماه، روز، ساعت، دقیقه و ثانیه نشان می‌دهد.
date	تاریخ جاری را در یک کاراکتر با فرمتِ رشته‌ای نشان می‌دهد.

دستورات مربوط به مدیریت متغیرها

جدول ۱۰-۲: دستورات مربوط به مدیریت متغیرها

دستور	توضیحات
clear	تمامی متغیرهای موجود در Work space را از حافظه پاک می‌کند.
clear x,y,z	فقط متغیرهایی که به آنها اشاره شده است (مثلاً در اینجا x,y,z) را حذف می‌کند.
who	متغیرهای موجود در Work space را نشان می‌دهد.
whos	لیستی از متغیرهای موجود در حافظه و اطلاعات مربوط به اندازه، بایت‌ها و کلاسشان را نمایش می‌دهد.

تمرین - حاصل عبارت زیر را بیابید

$$\frac{17\left[\sqrt{5}-1\right]}{\left[15^2-13^2\right]}+\frac{5^7\log_{10}(e^3)}{\pi\sqrt{121}}+\ln(e^4)+\sqrt{11}$$

آرایه ها و ماتریس ها

❖ **آرایه:**

✓ یک آرایه، لیستی از اعداد قرار گرفته در یک سطرو/یا ستون است.

✓ بردار یک آرایه یک بعدی و ماتریس یک آرایه دو بعدی است.

❖ **بردار سطری:**

مثال

$x = [۷,-۱, ۲,-۵,۸]$ یا $x = [۷\ -۱\ \ ۲\ -۵\ \ ۸]$

بردار ستونی:

$x = [\ ۷$ یا $x = [۷;-۱;۲;-۵;۸]$

-۱

۲

-۵

$۸]$

آرایه ها و ماتریس ها

❖ **ماتریس:**

مثال

$$A = \begin{bmatrix} 1 & 3 & -4 \\ 0 & -2 & 8 \end{bmatrix}$$

$A=[۱\ \ ۳\ -۴;۰\ \ -۲\ \ ۸]$

تمرین - ماتریس زیر را در MATLAB تعریف کنید .

$$B = \begin{bmatrix} -5x & \ln 2x + 7\sin 3y \\ 3i & 5 - 13i \end{bmatrix}$$

آدرس دهی محدوده ای از آرایه ها

- ✓ برای آدرس دهی محدوده ای از درایه ها در یک بردار یا ماتریس می توان از « : » استفاده نمود.

❖ استفاده از « : » (کالن) در یک بردار :

- ✓ **V(:)** – این دستور به تمام درایه های بردار V (چه سطری و چه ستونی) اشاره می کند.
- ✓ **V(*m*:*n*)**– این دستور به درایه های mام تا nام بردار اشاره می کند. برای مثال:

```
>> V = [۲ ۵ -۱ ۱۱ ۸ ۴ ۷ -۳ ۱۱]
>> u = V(۲: ۸)
u = ۵ -۱ ۱۱ ۸ ۴ ۷ -۳
```

آدرس دهی محدوده ای از آرایه ها

❖ استفاده از « : » در ماتریس :

جدول زیر موارد مختلف استفاده از : برای آدرس دهی داریه ها در یک ماتریس را ارائه می دهد.

جدول ۱۶-۲: موارد استفادهٔ« : » در ماتریس

دستور	توضیحات
A(:,*n*)	به ستون n ام ماتریس A اشاره دارد.
A(*n*,:)	به تمامی درایه‌های سطر n ام ماتریس A اشاره دارد.
A(:,*m*:*n*)	به ستون‌های m ام تا n ام ماتریس A اشاره دارد.
A(*m*:*n*,:)	به سطرهای m ام تا n ام ماتریس A اشاره دارد.
A(*m*:*n*,*p*:*q*)	به درایه‌های m ام تا n ام ستون‌های p تا q ماتریس A اشاره دارد

آدرس دهی محدوده ای از آرایه ها

مثال - پاک کردن محدوده ای از درایه ها

V = [۲ ۵ -۱ ۱۱ ۸ ۴ ۷ -۳ ۱۱]

V(۲ : ۵) = [ ]

V = ۲ ۴ ۷ -۳ ۱۱

تمرین : درایه های 5ام تا 8ام بردار بالا را با 1 جایگزین نمایید

عملیات روی آرایه ها

- **جمع و تفریق ماتریسها**

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}$$

آنگاه حاصل جمع یا تفریق آنها ($A \pm B$) به صورت زیر بدست می آید :

$$\begin{bmatrix} a_{11} \pm b_{11} & a_{12} \pm b_{12} & a_{13} \pm b_{13} \\ a_{21} \pm b_{21} & a_{22} \pm b_{22} & a_{23} \pm b_{23} \end{bmatrix}$$

- **ضرب ماتریسی:**

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k}\, b_{k,j}$$

لذا برای انجام ضرب دو ماتریس باید تعداد ستونهای ماتریس اول با تعداد سطرهای ماتریس دوم برابر باشد. لذا در حالت کلی ، ضرب ماتریس ها جابجایی پذیر نیست.

عملیات روی آرایه ها

❖ **ترانهاده :**

ترانهادهٔ یک ماتریس ، یک ماتریس جدید است به گونه ای که سطرهای ماتریس اول (اصلی) ، ستونهای ماتریس جدید را تشکیل می دهند. ترانهادهٔ ماتریس مفروض A را با A^T نمایش می دهند. در MATLAB ، ترانهادهٔ ماتریس A ، به صورت A' نشان داده می شود.

❖ **دترمینان :**

دترمینان برای ماتریس های مربعی محاسبه می شود. بعنوان مثال ، برای ماتریس ۲×۲ ،A، دترمینان برابر است با :

$$|A| = a_{11}a_{22} - a_{21}a_{12}$$

MATLAB دترمینان یک ماتریس را با استفاده از تابع det محاسبه می نماید. det(A) دترمینان ماتریس مربعیِ A را محاسبه می کند.

عملیات روی آرایه ها

❖ **مقادیر ویژه :**

✓ به معادلهٔ زیر توجه کنید :

$$AX = \lambda A$$

که در آن A یک ماتریس مربعیِ $n \times n$ ، X یک بردار ستونی با n سطر و λ یک اسکالر است . مقادیر λ برای هر X غیر صفر را **مقادیر ویژهٔ** ماتریس A ، و مقادیر متناظر X را ، **بردارهای ویژهٔ** ماتریس A می نامند.

✓ معادلهٔ زیر را « معادلهٔ مشخصهٔ » ماتریس A می نامند. حل این معادله، مقادیر ویژهٔ ماتریس A را می دهد.

$$|A - \lambda I| = 0$$

در MATLAB دستور eig() مقادیر ویژه ماتریس را می دهد.

توابع مربوط به آرایه ها

جدول ۱۹-۲: توابع آرایه‌ای MATLAB

تابع	توضیحات	مثال
mean(A)	اگر A یک بردار باشد، مقدار میانگین درایه‌ها را بر می‌گرداند.	>> A = [3 7 2 16]; >> mean (A) ans = 14
C=max(A)	اگر A یک بردار باشد، C بزرگترین درایهٔ A است.اگر A یک ماتریس باشد، C یک بردار سطری، حاوی بزرگترین درایهٔ هر ستون ماتریس A می‌باشد.	>> A = [3 7 2 16 9 5 18 13 04]; >> C = max (A) C = 18
[d,n]=max(A)	اگر A یک بردار باشد، *d* بزرگترین درایهٔ A و *n* موقعیت آن درایه در بردار می‌باشد (اگر بزرگترین درایه تکرار داشته باشد، اولین موقعیت حضور آن درایه را برمی‌گرداند)	>> [d, n] = max (A) d = 18 n=7
min(A)	مانند تابع max(A) عمل می‌کند، ولی برای کوچکترین درایه.	>> A = [3 7 2 16]; >> min (A) ans = 2
[d,n]=min(A)	مانند دستور [d,n]=max(A) عمل می‌کند، ولی برای کوچکترین درایه.	>> [d,n] = min (A) d = 16 n = 4
sum(A)	اگر A یک بردار باشد، مجموع درایه‌های بردار را برمی‌گرداند.	>> A = [3 7 2 16]; >> sum (A) ans = 28
sort(A)	اگر A یک بردار باشد، اجزاء (درایه‌های) بردار را به صورت صعودی مرتب می‌کند.	>> A = [3 7 2 16]; >> sort (A) ans = 2 3 7 16

توابع مربوط به آرایه ها

تابع	توضیحات	مثال
median(A)	اگر A یک بردار باشد، مقدار « میانهٔ» درایه‌های بردار را برمی‌گرداند.	>> A = [3 7 2 16]; >> median (A) ans = 5
std(A)	اگر A یک بردار باشد «انحراف معیارِ» درایه‌های بردار را بر می‌گرداند.	>> A = [3 7 2 16]; >> std (A) ans = 6.3770
det(A)	دترمینان ماتریس مربعیِ A را برمی‌گرداند.	>> A = [1 2; 3 4]; >> det (A) ans = – 2
dot(a,b)	حاصل ضربِ نقطه‌ای (اسکالر) دو بردار a و b را محاسبه می‌کند. بردار می‌تواند سطری یا ستونی باشد.	>> a = [5 6 7]; >> b = [4 3 2]; >> dot (a, b) ans = 52
cross(a,b)	حاصل ضربِ برداریِ دو بردار a و b (یعنی a×b) را محاسبه می‌کند. دو بردار باید ۳ درایه داشته باشند.	>> a = [5 6 7]; >> b = [4 3 2]; >> cross (a, b) ans = – 9 18 – 9
inv(A)	ماتریس معکوسِ، ماتریس مربعی A را برمی‌گرداند.	>> a = [1 2 3; 4 6 8; – 1 2 3]; >> inv (A) ans = – 0.5000 0.0000 – 0.5000 – 5.0000 1.5000 1.0000 3.5000 – 1.0000 – 0.5000

WWW.MOHANDES.ORG

توابع مربوط به آرایه ها

تمرین - دو ماتریس زیر را در نظر بگیرید :

$$B = \begin{bmatrix} 7 & 4 & 2 \\ 3 & 5 & 6 \\ -1 & 2 & 1 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 4 \\ -1 & 6 & 7 \end{bmatrix}$$

با استفاده از MATLAB مقادیر زیر را محاسبه نمایید.

(a) $A + B$
(b) AB
(c) A^2
(d) A^T
(e) B^{-1}
(f) $B^T A^T$
(g) $A^2 + B^2 - AB$
(h) $|A|\ ,\ |B|\ ,\ |AB|$

عملیات محاسباتی درایه به درایه

- ✓ محاسباتِ درایه به درایه را فقط می توان با آرایه ها و ماتریس های هم اندازه انجام داد. ضرب ، تقسیم ، و به توان رساندنِ درایه به درایه ای دو بردار یا ماتریس در MATLAB ، با تایپ یک نقطه (.) در مقابل عملگر محاسباتی ، انجام می گیرد.

جدول ۱۸-۲: عملیات محاسباتیِ درایه به درایه

عملگرهای ماتریسی		عملگرهای درایه به درایه (آرایه‌ای)	
جمع	+	جمع	+
تفریق	-	تفریق	-
ضرب	*	ضرب درایه به درایه	.*
توان	^	توان درایه به درایه	.^
تقسیم چپ	/	تقسیم چپ درایه به درایه	./
تقسیم راست	\	تقسیم راست درایه به درایه	.\

تولید اعداد تصادفی

- ✓ فرآیندهای فیزیکی و کاربردهای مهندسی فراوانی وجود دارد که در آنها ، به استفاده از اعداد تصادفی برای پیشرفت در حل یک مسئله نیاز است.
- ✓ MATLAB دو دستورِ *rand* و *randn* دارد که می توان از آنها برای تخصیص اعداد تصادفی به متغیرها استفاده کرد.

جدول ۲۰-۲: دستور *rand*

دستور	توضیحات	مثال
rand	یک عدد تصادفی بین ۰ و ۱ ایجاد می‌کند.	>> rand ans = 0.9501
rand(1,n)	یک بردار سطری n عضوی از اعداد تصادفی بین ۰ تا ۱ ایجاد می‌کند.	>> a = rand(1, 3) a= 0.4565 0.0185 0.8214

تولید اعداد تصادفی

دستور	توضیحات	مثال
rand(n)	یک ماتریس $n \times n$ با درایه‌های تصادفی بین ۰ و ۱ ایجاد می‌کند.	>> b = rand(3) b= 0.7382 0.9355 0.8936 0.4057 0.4103 0.3529 0.1763 0.9165 0.0579
rand(m,n)	یک ماتریس $m \times n$ با درایه‌های تصادفی بین ۰ و ۱ ایجاد می‌کند.	>> c = rand(2, 3) c= 0.1987 0.2722 0.0153 0.2028 0.6038 0.1988
randperm(n)	یک بردار سطری با n عضو از اعداد صحیح بین ۱ تا n که به صورت تصادفی قرار می‌گیرند.	>> randperm(7) ans = 5 2 4 7 1 6 3

دستور تولید اعداد تصادفیِ گوسی (نرمال)

- ✓ اگر یک توزیع نرمال مشخص شده باشد ، MATLAB می تواند مقادیر گوسی (نرمال) با میانگین صفر و واریانس ۱.۰ تولید نماید. این توابع عبارتند از :

$randn(n)$– این دستور یک ماتریس $n \times n$ حاوی اعداد تصادفی گوسی (یا نرمال) با میانگین صفر و واریانس ۱ تولید می کند.

$randn(m,n)$– این دستور یک ماتریس $m \times n$ حاوی اعداد تصادفی گوسی (نرمال) با میانگین صفر و واریانس ۱ تولید می کند.

نکته : برای مدل کردن نویز وارد بر سیستم های کنترل معمولا از سیگنال گوسی (اعداد تصادفی گوسی) استفاده می کنیم.

چند جمله ایها

- ✓ یک چند جمله ای ، تابعی از یک متغیر است که می توان آن را به صورت زیر بیان کرد :

$$f(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \ldots + a_{n-1} x^1 + a_n$$

که در آن x ، متغیر است و ضرایب چند جمله ای با مقادیر a_0 ، a_1 ، .... نشان داده می شوند.

- ✓ درجهٔ یک چند جمله ای برابر با بزرگترین توان متغیر می باشد.
- ✓ در MATLAB ، یک چند جمله ای ، یا بعبارت دیگر ضرایب چند جمله ای را با یک بردار نشان می دهیم.
- ✓ برای وارد کردن اطلاعات یک چند جمله ای در MATLAB ، ضرایب چند جمله ای را به صورت نزولی (از لحاظ توان متغیر) در یک بردار وارد می کنیم

چند جمله ایها

مثال :

$$۵s^۵ + ۷s^۴ + ۲s^۲ - ۶s + ۱۰$$

```
>> x = [۵  ۷  ۰  ۲  −۶  ۱۰]
```

توجه داشته باشید وارد کردن تمامی ضرایب لازم است . اگر یکی از ضرایب چند جمله ای صفر بود (یعنی وجود نداشت) ، در بردار ضرایب به جای آن صفر می گذاریم .

توابع چند جمله ایها

- ***conv (a,b)***- این دستور یک بردار ضرایب ایجاد می کند که حاوی ضرایب حاصل ضربِ چند جمله ایهایی است که با a وb مشخص کرده ایم. بردارهاَی a و b لازمَ نیست هم اندازه باشند.

```
>>x = [۲ ۵];
>>y = [۱ ۳ ۷];
>>z = conv(x, y)
z =
        ۲  ۱۱  ۲۹  ۳۵
```

- ***[q,r]=deconv(n,d)*** – این دستور دوبردار را برمی گرداند، اولین بردارحاوی ضرایب خارج قسمت و دومین بردار حاوی ضرایب چند جمله ای باقیمانده می باشد.

توابع چند جمله ایها

z = [۲ ۱۱ ۲۹ ۳۵]; x = [۲ ۵]

>> [g, t] = deconv (z, x)

g = ۱ ۳ ۷

t = ۰ ۰ ۰ ۰

❖ ***roots (a)***– این تابع ریشه های چند جمله ای مشخص شده توسط بردار ضرایب a را محاسبه می کند. تعداد ریشه های یک چندجمله ای برابر درجهٔ آن چند جمله ای می باشد.

>> x = [۵ ۷ ۰ ۲ − ۶ ۱۰]

>> r = roots(x)

r=

− ۱.۸۶۵۲

− ۰.۴۶۴۱ + ۱.۰۸۳۲i

− ۰.۴۶۴۱ − ۱.۰۸۳۲i

۰.۶۹۶۷ + ۰.۵۳۵۵i

۰.۶۹۶۷ − ۰.۵۳۵۵i

توابع چند جمله ایها

❖ ***poly (r)*** – این دستور ضرایب چند جمله ای را که ریشه هایش در بردار r داده شده است ، مشخص می نماید.خروجیِ این تابع یک بردار سطری حاوی ضرایب چند جمله ای می باشد.

مقدار یک چند جمله ای به ازاء یک مقدار خاص را نیز می توان با استفاده از تابع *polyval* محاسبه کرد.

❖ ***polyval (a,x)***– این دستور چند جمله ای با ضرایب a را برای مقدار (یا مقادیر) x محاسبه می کند. نتیجه ماتریسی به اندازهٔ x خواهد بود.

مثال: مقدار چند جمله ای زیر در s = ۲ به صورت زیر بدست می آید :

>> x = polyval([۵ ۷ ۰ ۲ − ۶ ۱۰] , ۲)

x=

۲۷۸

M – فایل ها

- ✓ M- فایل ، یک سری دستورات و توابع است که پشت سر هم نوشته شده اند. دستوراتِ نوشته شده در یک M-فایل را می توان با تایپ نام فایل ، برای اجراء ، فراخوانی کرد. یک M-فایل همچنین می تواند M–فایل های دیگر را در داخل خود فراخوانی کند.

File ⇨ new ⇨ M-file ✓

- ✓ M- فایل ها دو نوع هستند:
- ✓ نوع اول یا همان اسکریپت فایل ها (Script files) ، دستوراتِ MATLAB هستند که پشت سر هم نوشته شده اند.

M – فایل ها

- ✓ دومین نوع M- فایل ها ، Function file ها (فایل های تابعی) هستند . Function file ، کاربر را قادر می سازد تا توابع کتابخانه ای اصلی را با افزودن توابع جدیدی که خود می سازد ، گسترش دهد . Function file ها می توانند یک یا چند مقدار را برگردانند.

Function file ها در MATLAB باید با دستوری به فرم زیر آغاز شوند:

function (name of result or results) = name (argument list)

function(نام متغیر های ورودی) نام تابع = (نام مقدار یا مقادیری که برگردانده می شوند)

برنامه نویسی در MATLAB

❖ عملگرهای مقایسه ای و منطقی :

یک عملگر مقایسه ای ، دو عدد را با هم مقایسه می کند.یک عملگر منطقی ، درست یا غلط بودنِ یک عبارات را بررسی کرده و نتیجهٔ درست (معادل ۱) و یا غلط (معادل ۰) را با توجه به نوع عملگر تولید می کند.

جدول ۲۱-۲: عملگرهای مقایسه‌ای

عملگر	نوع عملکرد
<	کوچکتر از
<=	کوچکتر مساوی با
>	بزرگتر از
>=	بزرگتر مساوی با
==	مساوی با
~=	نامساوی

برنامه نویسی در MATLAB

جدول ۲۲-۲: عملگرهای منطقی

عملگر منطقی	نام	توضیحات
& مثال:A&B	AND	روی دو عملوند (A و B) عمل می‌کند. اگر هردو درست باشند، نتیجه درست (معادل ۱) خواهد بود. در غیر این صورت نتیجه غلط (معادل ۰) خواهد بود.
| مثال:A|B	OR	روی دو عملوند (A و B) عمل می‌کند. اگر هردو و یا یکی از آن دو درست باشند، آنگاه نتیجه درست (۱) خواهد بود. در غیر این صورت (یعنی اگر هردو غلط باشند) نتیجه صفر است
~ مثال:~A	NOT	روی یک عملوند عمل می‌کند، و مقدار مخالف آن را می‌دهد. اگر عملوند نادرست باشد، حاصل درست (۱)، و اگر عملوند درست باشد، حاصل نادرست (۰) خواهد بود

برنامه نویسی در MATLAB

اولویت ها در MATLAB :

اولویت	عملیات
(بالاترین اولویت) ۱	پرانتز (اگر پرانتزهای تودرتو وجود داشته باشد، داخلی‌ترین پرانتز اولویت بالاتری دارد)
۲	توان
۳	نقیض منطقی NOT (~)
۴	ضرب و تقسیم
۵	جمع و تفریق
۶	عملگرهای مقایسه‌ای (<, <=, >, >=, ==, ~=).
۷	عملگر AND منطقی (&)
(پایین‌ترین اولویت) ۸	عملگر OR منطقی (|)

برنامه نویسی در MATLAB

❖ عبارات شرطی (if):

یک عبارت شرطی ، دستوری است که به MATLAB اجازهٔ تصمیم گیری می دهد که آیا ، یک گروه از دستورات را که پس از عبارت شرطی آمده اند اجراءکند ، یا آنها را اجراء نکرده و از روی آنها پرش نماید

فرم عمومی و کلی عبارت if به صورت زیر است :

```
if        عبارت منطقی
          دستورات
end
```

برنامه نویسی در MATLAB

	حالات مختلف استفاده از دستور if در زیر نمایش داده شده است:		
if , else,elseif	If متغیر ۱ دستورات (اگر متغیر ۱غیر صفر باشد این دستورات اجرا می‌شوند) elseif متغیر ۲ دستورات (اگر متغیر ۲ غیرصفر باشد، این دستورات اجرا می‌شوند) else دستورات (اگر هیچ کدام از دو متغیر درست نباشد این دستورات اجرا می‌شوند) end	If متغیر دستورات (اگر متغیر غیر صفر باشد این دستورات اجرا می‌شوند) else دستورات (اگر متغیر (غلط) یا صفر باشد، این دستورات اجرا می‌شوند) end	If متغیر دستورات (اگر متغیر غیر صفر باشد این دستورات اجرا می‌شوند) end

برنامه نویسی در MATLAB

❖ ساختار while در MATLAB :

در MATLAB ساختار و دستوری وجود دارد که حلقهٔ for را با ویژگیهایِ محدودکنندهٔ if ترکیب می کند. این ساختار حلقهٔ while نام دارد و فرم کلیِ آن به صورت زیر است :

while عبارت منطقی

دستورات

end

دستورات بین while و end تا زمانی که عبارتِ منطقی درست باشد (معادل ۱) ، به صورت مکرر اجراء می شوند. و یا اگر عبارت منطقی به جای یک متغیر اسکالر ساده ، یک ماتریس باشد ، تا زمانی که تمامی درایه های ماتریس غیر صفر باقی بمانند ، دستورات اجراء خواهند شد.

برنامه نویسی در MATLAB

❖ **دستورات خاتمهٔ برنامه:**

دستور	توضیحات
break	اجرای دستورات را در حلقه‌های for و while متوقف می‌کند. در حلقه‌های تودرتو، break فقط اجرای دستورات حلقه‌ای که در آن قرارگرفته است را متوقف می‌کند و به اجرای دستورات حلقه‌های بالاتر کاری ندارد.
return	این دستور عمدتاً در توابع MATLAB مورد استفاده قرار می‌گیرد. دستور return باعث می‌شود که اجرای برنامه از تابع خارج شده و به دستور بعد از تابع در برنامهٔ اصلی منتقل شود.
eror (' متن ')	اجرای برنامه را متوقف کرده و پیغامی که داخل پرانتز قرار گرفته است را روی صفحه نمایش می‌دهد. توجه داشته باشید که این پیغام باید بین ' ' قرار گیرد.

برنامه نویسی در MATLAB

❖ **حلقه** for **:**

✓ برای اجرای تعدادی دستور به تعداد مشخص

ساختار کلی :

for i= مقدار اولیه : گام : مقدار انتهایی

دستورات

end

گرافیک و رسم نمودار

✓ دستور اصلی و در عین حال ساده ترین دستور برای رسم نمودار ۲ بعدی به صورت زیر است :

plot (x مقادیر ، y مقادیر ، ' سبک های انتخابی ')

که در آن ، مقادیر x و مقادیر y ، بردارهایی حاوی مختصاتِ x,y نقاط مختلف شکل (گراف) می باشند. « سبک های انتخابی » علایم و نشانه هایی هستند که رنگ ، نوع خط و نحوهٔ نمایش نقاط خاص روی شکل را مشخص می کنند.

❖ **دستور hold**– با استفاده از دستور *hold on* ، تصاویر موجود در پنجره شکل نگه داشته می شوند و تمامی نمودارهای بعدی که با دستو *plot* رسم شوند به نمودارهای موجود افزوده خواهندشد. برای خروج از این حالت از دستور *hold off* استفاده می کنیم

گرافیک و رسم نمودار

تمرین:توابع زیر را با استفاده از دستور plot و دستور در یک شکل رسم کنید .

$y_1 = \sin t$,

$y_2 = t$,

$y_3 = t - (t^3/3!) + (t^5/5!) - (t^7/7!)$, $0 \le t \le 2\pi$

توابع زیر ا رسم کنید .

i) $f(t) = t \cos t$ $0 \le t \le 10\pi$,

ii) $y = 100 + e^{3t}$ $0 \le t \le 2\pi$

ریاضیات سمبولیک

- ✓ یک عبارت سمبولیک ، در MATLAB بعنوان یک رشتهٔ کاراکتری ذخیره می شود.برای تعریف یک عبات سمبولیک باید از ' ' استفاده شود. برای مثال 'sin(y/x)' یا 'x ^ ۴ + ۵*x^۳ + ۷*x^۲ − ۷'

- ✓ تعریف یک متغیر سمبولیک

syms نام متغیر

بحث بعد .... سیستمهای کنترل

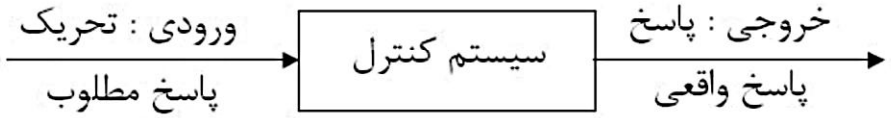
بخش دوم

مقدمه ای بر سیستمهای کنترل

سیستمهای کنترل

- ✓ ***کنترل*** یعنی تنظیم ، هدایت ، فرمان دادن ، یا اداره کردن.
- ✓ یک ***سیستم*** ، یک مجموعه ، یک گروه یا یک ترکیب از اجزاء مختلف (یا همان زیر سیستمها subsystems) می باشد.
- ✓ یک ***سیستم کنترل*** ارتباط و اتصال اجزایی است ، که پیکربندیِ یک سیستم خاص را جهت ایجاد یک پاسخ سیستمی مناسب ، تشکیل می دهند.
- ✓ به منظور شناسایی یا تعریف یک سیستم کنترل ، از دو عبارتِ : *ورودی* و *خروجی* بهره می بریم.

سیستمهای کنترل



- ✓ ***ورودی*** ، عبارت است از تحریک یا دستوری که به سیستم کنترل اعمال می شود. ورودی ها می توانند متغیرهای فیزیکی ، و یا سیگنالهایی نظیر سیگنال مرجع یا متغیر های دلخواهی باشند که می خواهیم در خروجیِ سیستم کنترل آن ها را داشته باشیم .
- ✓ ***خروجی*** پاسخ عملیِ حاصل از یک سیستم کنترل می باشد.خروجی می تواند برابر همان ورودی اعمال شده باشد و یا برابر آن نباشد.

سیستمهای کنترل

- ✓ سیستم های کنترل می توانند بیش از یک ورودی یا خروجی داشته باشند.
- ✓ اگر خروجی و ورودی داده شده باشد ، امکان شناسایی یا تعریفِ نوعِ اجزای بکار رفته در سیستم وجود دارد.
- ✓ به طور کلی ، ۳ نوع سیستم کنترلی وجود دارد :

الف- سیستم کنترلیِ ساخته شده توسط انسان مثل سوئیچ الکتریکی و ... .

ب- سیستم های کنترل طبیعی ، شامل سیستم های کنترل بیولوژیکی مثل اشاره کردن به یک شئ با انگشت .

ج- سیستم های کنترلی که هم از اجزاءطبیعی و هم از اجزاء ساخته شده توسط انسان ساخته می شوند مثل کنترلِ رانندگی یک اتومبیل توسط یک انسان.

مثال های بیشتری از سیستمهای کنترل

- ✓ سیستم های کنترل تهویه هوا و گرمایشی منازل مسکونی بوسیلهٔ ترموستات
- ✓ سیستم کنترل سرعت اتومبیل .
- ✓ کنترل های دستی از قبیل :

- بازو بسته کردن یک پنجره برای تنظم دمای هوا

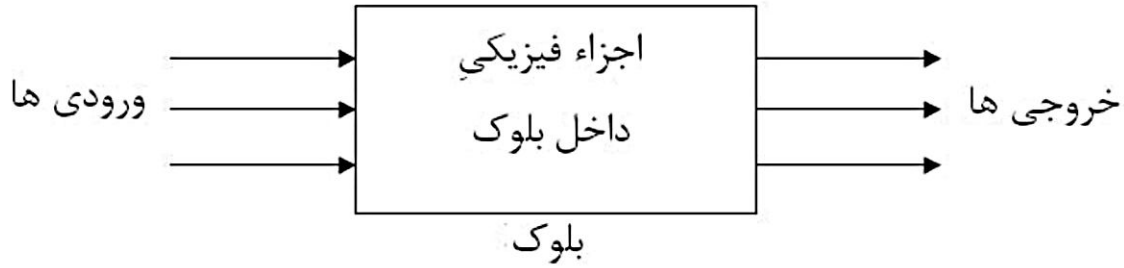
- بکارگیریِ یک سوئیچ برای تنظیم نور یک اتاق

- کنترلِ سرعت اتومبیل توسط راننده با تنظیم بنزین ورودی به موتور (با استفاده از پدال گاز)

- ✓ سیستم کنترلی که به صورت خودکار لامپ یک اتاق را در تاریکی روشن و در روشنایی روز خاموش می کند.
- ✓ گرم کننده های آبِ خودکار.

برخی تعاریف

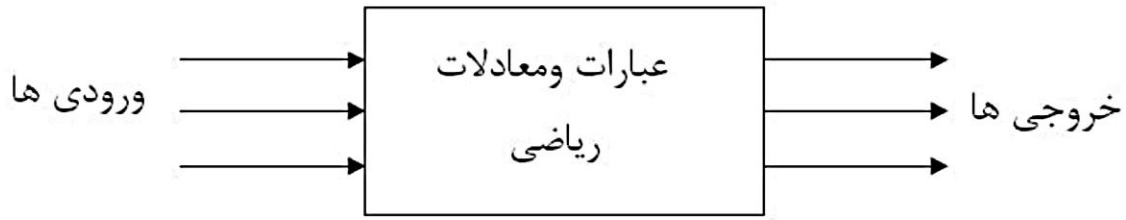
بلوک . یک بلوک ، مجموعه ای از عناصری است که می توانند با ویژگیهای مشترکی که توسط رابطهٔ بین ورودی / خروجی تشریح می شود ، با هم تشکیل یک گروه دهند .یک بلوک دیاگرام ، یک نمایش تصویریِ ساده از رابطهٔ علت و معلولی بین ورودیها و خروجی های یک سیستم فیزیکی است.



بلوک

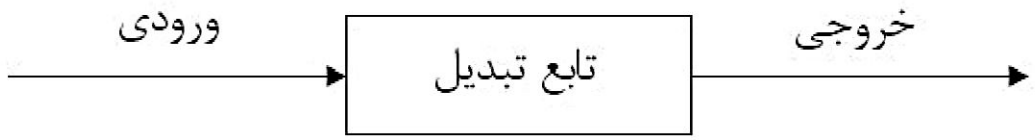
برخی تعاریف

خصوصیاتِ ورودی و خروجیِ تمامیِ اجزاء داخل بلوک را می توان، بوسیلهٔ معادلات و تعابیر ریاضیِ مناسب ، تشریح نمود .



برخی تعاریف

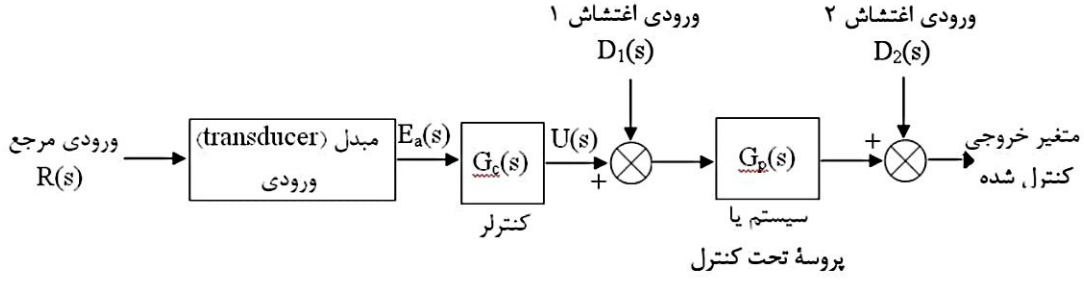
تابع تبدیل . تابع تبدیل ،خصوصیتی است که فقط به اجزاء سازنده سیستم بستگی دارد ، و به ورودی ها (تحریک) و شرایط اولیه وابسته نیست. تابع تبدیلِ یک سیستم (یا یک بلوک) که به صورت نسبت خروجی به ورودی تعریف می شود



$$\text{تابع تبدیل} = \frac{\text{خروجی}}{\text{ورودی}}$$

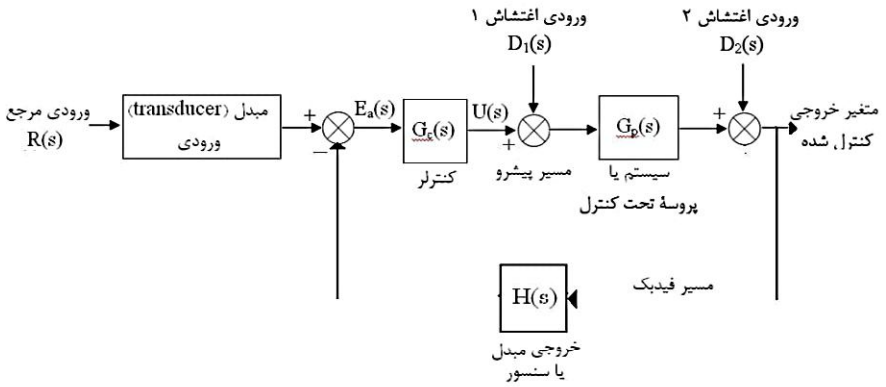
پیکربندی سیستمهای کنترل

سیستم های کنترل حلقه باز . سیستم های کنترل حلقه باز ، ساده ترین حالت ابزار کنترلی را نشان می دهند. در این پیکربندی خواسته های ما از سیستم کنترل به خوبی برآورده نمی شود و تنها به دلیل سادگی و اقتصادی بودن از آنها استفاده می کنیم.



پیکربندی سیستمهای کنترل

سیستم حلقه بسته (کنترل فیدبک).سیستم های کنترل حلقه بسته با استفاده از ساختار فیدبک ، خروجی خود را در جهت بازسازیِ دقیقِ سیگنال ورودی ، هدایت می کنند.یک سیستم با یک یا چند مسیر فیدبک را **سیستم کنترل حلقه بسته** می نامیم.



اصطلاحات سیستمهای کنترل

- ✓ **سیگنال خطا یا تحریک** : سیگنال خطا یا تحریک عبارت است از سیگنال ورودیِ مرجع به اضافه و یا منهای سیگنال فیدبک .
- ✓ **خروجیِ کنترل شدهٔ C(s)** : خروجی کنترل شدهٔ$C(s)$ متغیر خروجیِ قسمت تحت کنترل ، در سیستم کنترل می باشد.
- ✓ **کنترلر** : اجزاء یک سیستم کنترل حلقه باز را معمولاً می توان به دو بخش تقسیم کرد : کنترل کننده و فرآیند کنترل . کنترلر ، یک فرآیند یا سیستم را راه اندازی و هدایت می کند .
- ✓ **ورودیِ نویز یا اغتشاش** : ورودی نویز یا اغتشاش ، یک تحریک یا سیگنال ورودی نامطلوب است که بر خروجیِ کنترل شده تاثیر منفی دارد .

اصطلاحات سیستمهای کنترل

- ✓ **اجزاء (کنترلی) پیشرو** : اجزاء کنترل پیشرو ، اجزاء مسیر پیشرو هستند که سیگنال کنترلی اعمال شده به فرآیند یا سیستم را تولید می کند. اجزاء کنترلیِ پیشرو عبارتند از : کنترلر(ها) ، جبران ساز(ها) ، یا اجزاء متعادل ساز ، و تقویت کننده ها .
- ✓ **اجزاء فیدبک** : اجزاء فیدبک رابطهٔ اساسی بین خروجیِ کنترل شدهٔ $C(s)$ و سیگنال فیدبک اولیهٔ $B(s)$ را برقرار می کنند. این اجزاء عبارتند از : سنسورهای خروجیِ کنترل شده ، جبران سازها ، و اجزاءکنترلر .
- ✓ **مسیر فیدبک** : مسیر فیدبک ، مسیر انتقال برگشتی از خروجیِ کنترل شده به نقطهٔ جمع می باشد .
- ✓ **مسیر پیشرو** : مسیر پیشرو ،مسیر انتقال از نقطهٔ جمع به خروجی کنترل شده می باشد.

اصطلاحات سیستمهای کنترل

- ✓ **مبدل (ترنسدیوسر) ورودی** : مبدل ورودی ، ورودیها را به فرمی که توسط کنترلر مورد استفاده قرار می گیرد ، تبدیل می کند .
- ✓ **مسیر** : مسیر ، مجموعه ای از شاخه های متوالیِ پیوسته ای است که هم جهت هستند .
- ✓ **بهرهٔ مسیر** : حاصل ضربِ بهرهٔ شاخه هایی که در عبور از یک مسیر با آنها مواجه می شویم ، بهرهٔ مسیر نام دارد.
- ✓ **حلقه** : حلقه ، مسیری است که ابتدا و انتهایِ آن روی یک گره قرار داشته باشد ، و از هیچ کدام از دیگر گره های مسیر پیشرو بیش از یک بار عبور نشده باشد.
- ✓ **بهرهٔ حلقه** : بهره حلقه ، عبارت است از بهرهٔ مسیر حلقه .
- ✓ **فیدبک منفی** : اگر علامت فیدبک در نقطهٔ جمع (نقطه تجمیع) ، منفی (تفریق) باشد ، فیدبک منفی داریم.

اصطلاحات سیستمهای کنترل

- ✓ **دستگاه ، فرآیند یا سیستم کنترل شده $G_p(s)$**: دستگاه ، فرآیند یا سیستم کنترل شده عبارت است از ، سیستم ، زیر سیستم ، فرآیند یا یک شئ که توسط سیستم کنترل فیدبک کنترل می شود. برای مثال ، دستگاه می تواند سیستمِ یک کوره باشد که متغیر خروجی آن دما است .
- ✓ **فیدبک مثبت** : اگر علامت فیدبک درنقطهٔ جمع (نقطهٔ تجمیع) ، جمع باشد ، فیدبک مثبت داریم .
- ✓ **ورودیِ مرجع $R(s)$** : ورودی مرجع ، سیگنال خارجی اعمال شده به سیستم کنترل است که معمولاً در نقطهٔ تجمیع وارد شده ، و دستورلازم برای انجام یک عمل یا فرآیندِ تعیین شده را صادر می کند. ورودی مرجع ، معمولاً پاسخ مطلوب و ایده آلی است که انتظار داریم آن را در خروجی داشته باشیم.
- ✓ **پاسخ زمانی** : پاسخ زمانیِ یک سیستم عبارت است از خروجی به صورت یک تابع زمانی

سیستم های فیدبک

❖ *مهمترین ویژگی های سیستم های کنترلی حلقه بسته (فیدبک دار) :*

- ✓ افزایش دقت : توانایی سیستم برای باز تولیدِ یک ورودی ، در خروجی سیستم به صورت دقیق.
- ✓ کاهش حساسیتِ نسبتِ خروجی به ورودی ، در مقابل تغییر پارامترها و خصوصیات سیستم .
- ✓ کاهش اثرات غیرخطی بودن و اعوجاج .
- ✓ افزایش پهنای باند (پهنای باندِ یک سیستم محدوده ای از فرکانسهاست که سیستم برای آنها پاسخ مناسب خواهد داشت).
- ✓ کاهش نوسان یا ناپایداری .
- ✓ کاهش اثرات اغتشاشاتِ خارجی یا نویز .

تجزیه و تحلیل سیستم کنترل و طراحی اهداف

- ✓ مهندسی سیستم های کنترل عبارت است از *تحلیل* و *طراحیِ* پیکربندی های مختلف سیستم های کنترل.
- ✓ سیستم های کنترل دینامیک هستند ، یعنی پاسخ آنها به یک ورودی ، قبل از رسیدن به پاسخ حالتِ پایداری که متناسب با ورودی است ،تحت تاثیر پاسخ گذرا قرار می گیرد.
- ✓ ۳ هدف اصلی از تحلیل و طراحیِ سیستم های کنترل عبارتند از :

۱- ایجاد یک پاسخ قابل قبول در مقابلِ یک اغتشاش گذرا.

۲- کاهش خطای حالت ماندگار : در این مبحث ، دقتِ پاسخ حالت ماندگار مد نظر است.

۳- رسیدن به پایداری : سیستم های کنترل باید به گونه ای طراحی شوند که پایدار باشند. پاسخ طبیعی سیستم در زمان بینهایت یا باید به صفر میل کند و یا نوسانی باشد .

بخش سوم

تجزیه و تحلیل سیستم های کنترل بوسیله MATLAB

(قسمت اول)

تبدیل لاپلاس و معکوس آن

laplace(f)- این دستور از عبارت سمبولیک f لاپلاس می گیرد.

مثال – از تابع زیر تبدیل لاپلاس بگیرید.

$f(t) = ۷t^۳ cos(۵t + ۶۰°)$

حل –

```
>> syms t
>> f = ۷ * t^۳*cos(۵*t + (pi/۳));
>> laplace(f)
```

تمرین – تبدیل لاپلاس بگیرید.

$f(t) = -۷te^{-۵t}$

$f(t) = ۳ sin(۵t + ۴۵^o)$

تبدیل لاپلاس و معکوس آن

ilaplace(*f*)- از عبارت سمبولیک f معکوس لاپلاس می گیرد

مثال-

$$F(s) = \frac{1}{s^4 + 5s^3 + 7s^2}$$

حل –

```
>> syms s
>> f = ۱/(s^۴ + ۵*s^۳ + ۷*s^۲ );
 >> ilaplace(f)
```

تعریف تابع تبدیل

تابع تبدیل یک سیستم خطیِ تغییرناپذیر با زمان ، به صورت نسبت دو چندجمله ای به هم ، بیان می شود. تابع تبدیل برای یک سیستم یک ورودی و یک خروجی (SISO) به صورت زیر نوشته می شود :

$$H(s) = \frac{b_0 s^n + b_1 s^{n-1} + \ldots + b_{n-1} s + b_n}{a_0 s^m + a_1 s^{m-1} + \ldots + a_{m-1} s + a_m}$$

ضرایب صورت : $b_0 , b_1 , \ldots , b_n$

ضرایب مخرج : $a_0 , a_1 , \ldots , a_m$

تعریف تابع تبدیل

دستور tf(num,den) : برای تعریف تبع تبدیل از این دستور استفاده می شود.

مثال – سیستم LTI تک ورودی – تک خروجی زیر را در نظر بگیرید :

$$y'' + 6y' + 5y = 4u' + 3u$$

تا بع تبدیل این سیستم را در MATLAB تعریف نمایید.

تعریف تابع تبدیل

تابع تبدیل سیستم :

$$G(s)=\frac{Y(s)}{U(s)}=\frac{4s+3}{s^2+6s+5}$$

```
>> num=[۴ ۳];
>> den=[۱ ۶ ۵];
>> sys=tf(num,den)

Transfer function:
  ۴ s + ۳
-------------
s^۲ + ۶ s + ۵
```

تعریف تابع تبدیل

دستور **tfdata()** : برای بازیابیِ اطلاعات مربوط به ضرایب صورت و مخرج یک تابع تبدیل از این دستور استفاده می شود .

مثال –

```
>> [a,b]=tfdata(sys,'v')
a =
    ۰    ۴    ۳
b =
    ۱    ۶    ۵
```

نکته – پارامتر v باعث می شود ضرایب صورت و مخرج به صورت آرایه ای بازگردانده شود.

تعریف تابع تبدیل

❖ **تابع تبدیل به صورت ضرایب صفر و قطب ها**
(Zero-Pole-Gain)

$$H(s) = k\,\frac{(s-z_1)(s-z_2)\ldots(s-z_n)}{(s-p_1)(s-p_2)\ldots(s-p_m)}$$

صفرهای تابع تبدیل: $z_1 , z_2 , \ldots , z_n$

قطب های تابع تبدیل: $p_1 , p_2 , \ldots , p_m$

بهره تابع تبدیل : k

تعریف تابع تبدیل

دستور zpk(num,den,k) : این دستور تابع تبدیل را به شکل ضرایب صفر و قطب و بهره تعریف می کند. Zpkdata(sys) نیز عکس دستور فوق است

مثال –

```
>> a=[۱ ۲];
>> b=[۳ ۲ ۴];
>> sys=zpk(a,b,۵)

Zero/pole/gain:
  ۵ (s-۱) (s-۲)
----------------
(s-۳) (s-۴) (s-۲)
```

تعریف تابع تبدیل

❖ **دستورات مربوط به تبدیل دو فرم تابع تبدیل به یکدیگر :**

✓ **[z, p, k] = tf۲zp(num,den)**– این تابع ، تابع تبدیل چندجمله ای را به تابع تبدیل به فرم حاصل ضربِ صفرها و قطب ها تبدیل می کند .

✓ **[num, den] = zp۲tf(z’, p’, k)** تابع تبدیل به فرمت حاصل ضرب صفرها و قطب ها را به تابع تبدیل چند جمله ای تبدیل می کند .

WWW.MOHANDES.SITE

تعریف تابع تبدیل

تمرین- تابع اول را به فرمت چند جمله ای و تابع دوم را به فرم حاصل ضرب صفر و قطب نشان دهید

$$F(s) = \frac{8(s+1)(s+3)}{(s+2)(s+4)(s+6)^2}$$

$$G(s) = \frac{s^4 + 20s^3 + 27s^2 + 17s + 35}{s^5 + 8s^4 + 9s^3 + 20s^2 + 29s + 32}.$$

تجزیه به کسرهای جزئی

[r,p,k] = residue(B,A)- تابع تبدیل چند جمله ایِ را به کسرهای جزئی تشکیل دهنده اش تبدیل می کند:

$$H(s) = \frac{b_0 s^n + b_1 s^{n-1} + \ldots + b_{n-1} s + b_n}{a_0 s^m + a_1 s^{m-1} + \ldots + a_{m-1} s + a_m} \longrightarrow$$

$$\longrightarrow \quad H(s) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \ldots + \frac{r_n}{s - p_n} + k(s)$$

تجزیه به کسرهای جزئی

مثال - تابع F(s) را به کسرهای جزئی بسط دهید

```
>> b = [۰ ۰ ۰ ۰ ۱];
>> a = [۱ ۵ ۷ ۰ ۰];
>> [r, p, k] = residue (b, a)
r =
   ۰.۰۵۱۰ − ۰.۰۶۴۸i
   ۰.۰۵۱۰ + ۰.۰۶۴۸i
 − ۰.۱۰۲۰
   ۰.۱۴۲۹
p =
 − ۲.۵۰۰۰ + ۰.۸۶۶۰i
 − ۲.۵۰۰۰ − ۰.۸۶۶۰i
   ۰
   ۰
k = [ ]
```

تجزیه به کسرهای جزئی

تمرین – توابع زیر را به کسرهای جزئی بسط دهید و از پاسخ معکوس لاپلاس بگیرید.

$$F(s) = \frac{5s^2 + 3s + 6}{s^4 + 3s^3 + 7s^2 + 12}$$

$$F(s) = \frac{s^4 + 3s^3 + 5s^2 + 7s + 25}{s^4 + 5s^3 + 20s^2 + 40s + 45}$$

تعریف یک سیستم به صورت فضای حالت

✓ یکی از مفید ترین روشهای نمایش یک سیستم ، نمایش آن در ***فضای حالت*** است ، که آن را *نمایش داخلی سیستم* نیز می نامیم .

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

بردار حالت : x

بردارهای ورودی و خروجی : u و y

ماتریس های فضای حالت سیستم : A , B , C , D

تعریف یک سیستم به صورت فضای حالت

دستور ss(A,B,C,D) – این دستور سیستمی با ماتریس های فضای حالت داده شده، در MATLAB تعریف می کند.

مثال- سیستم LTI تک ورودی و تک خروجی زیر

$$y''+6y'+5y=4u'+3u$$

نمایش حالتی به صورت

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -5 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

دارد. این سیستم را در MATLAB تعریف نمایید

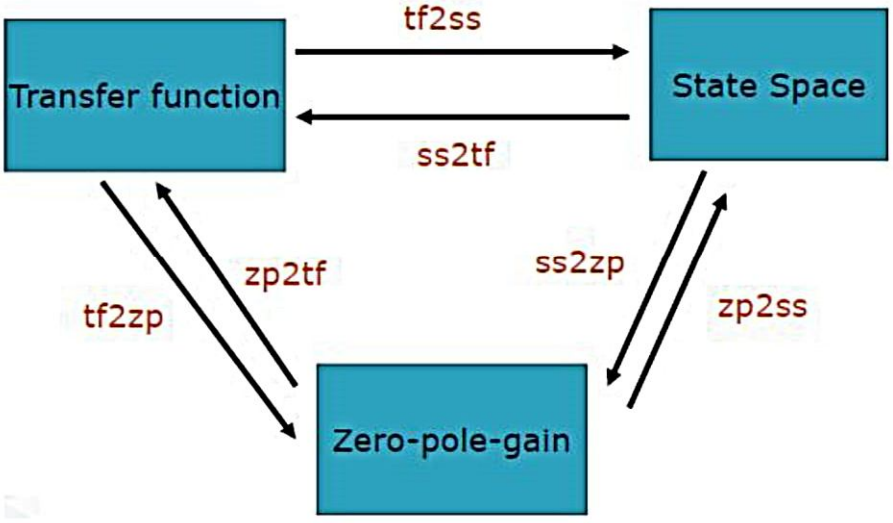
تعریف یک سیستم به صورت فضای حالت

```
>> sys=ss([۰ ۱;-۵ -۶],[۰;۱],[۳ ۴],۰)
a =
     x۱  x۲
  x۱   ۰   ۱
  x۲  -۵  -۶
b =
     u۱
  x۱   ۰
  x۲   ۱
c =
     x۱  x۲
  y۱   ۳   ۴
d =
     u۱
  y۱   ۰
```

تبدیل مدل ها به یکدیگر

- ✓ **[num, den] = ss۲tf(*A, B, C, D, iu*)**: فضای حالت به تابع تبدیل
- ✓ **[*z, p, k*] = ss۲zp(*A, B, C, D, iu*)**: فضای حالت به صفر و قطب
- ✓ **[*A,B,C,D*] = tf۲ss(num,den)** : تابع تبدیل به فضای حالت
- ✓ **[*z, p, k*] = tf۲zp(num,den)**: تابع تبدیل به صفرو قطب
- ✓ **[num, den] = zp۲tf(*z, p, k*)**: صفر و قطب به تابع تبدیل
- ✓ **[*A, B, C, D*] = zp۲ss(*z, p, k*)**: صفر و قطب به فضای حالت

تبدیل مدل ها به یکدیگر



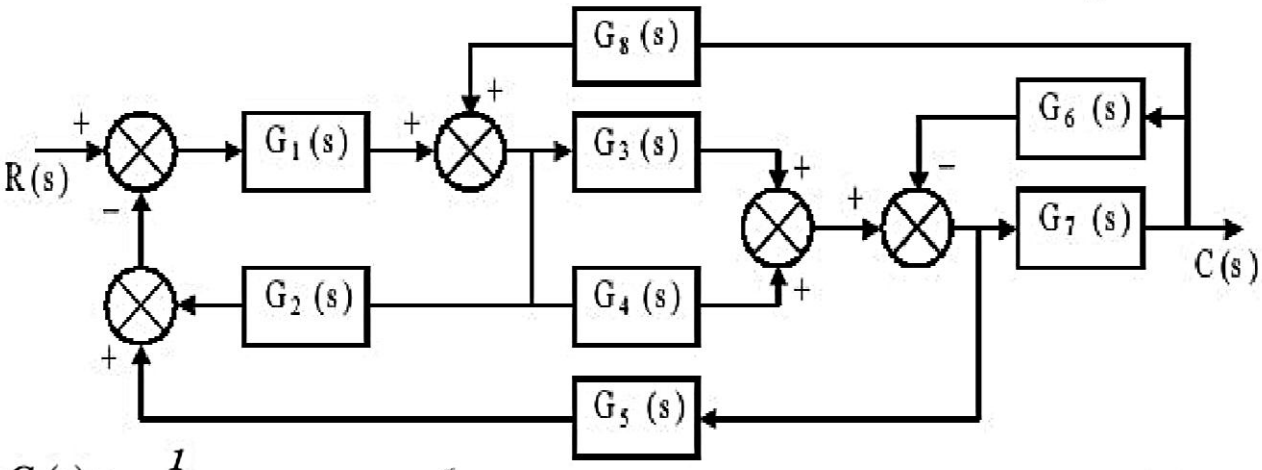
یافتن تابع تبدیل حلقه بسته یک دیاگرام بلوکی با استفاده از MATLAB

- ✓ روش معمول در یافتن تابع تبدیل یک دیاگرام بلوکی ، روش بهره میسون است.
- ✓ در MATLAB از مجموعه دستورات زیر برای یافتن تابع تبدیل استفاده می کنیم :

- **دستورِ *append()*** – با این دستور توابع تبدیل بلوک های مختلفِ موجود در دیاگرامِ یک سیستم کنترل را تعریف می کنیم .
- **دستورِ *connect()*** – با این دستور تابع تبدیل کلیِ سیستم کنترل را بدست می آوریم .

یافتن تابع تبدیل حلقه بسته یک دیاگرام بلوکی با استفاده از MATLAB

مثال -



$$G_1(s) = \frac{1}{(s+7)} \qquad G_4(s) = \frac{1}{s} \qquad G_5(s) = \frac{7}{(s+3)} \qquad G_7(s) = \frac{5}{(s+5)}$$

$$G_2(s) = \frac{1}{(s^2+6s+5)}$$

$$G_6(s) = \frac{1}{(s^2+7s+5)} \qquad G_8(s) = \frac{1}{(s+9)}$$

$$G_3(s) = \frac{1}{(s+8)}$$

یافتن تابع تبدیل حلقه بسته یک دیاگرام بلوکی با استفاده از MATLAB

حل – مرحله اول : تعریف توابع تبدیل هرکدام از بلوک ها

```
G۱ = tf ( [۰ ۰ ۱], [۰ ۱ ۷]);
G۲ = tf ( [۰ ۰ ۱], [۱ ۶ ۵]);
G۳ = tf ( [۰ ۰ ۱], [۰ ۱ ۸]);
G۴ = tf ( [۰ ۰ ۱], [۰ ۱ ۰]);
G۵ = tf ( [۰ ۰ ۷], [۰ ۱ ۳]);
G۶ = tf ( [۰ ۰ ۱], [۱ ۷ ۵]);
G۷ = tf ( [۰ ۰ ۵], [۰ ۱ ۵]);
G۸ = tf ( [۰ ۰ ۱], [۰ ۱ ۹]);
G۹ = tf ( [۰ ۰ ۱], [۰ ۰ ۱]);
```

نکته : تابع G۹ به عنوان بلوک ورودی در نظر گرفته می شود.

یافتن تابع تبدیل حلقه بسته یک دیاگرام بلوکی با استفاده از MATLAB

مرحله دوم : با استفاده از دستور append یک بانک اطلاعاتی از بلوک های تبدیل می سازیم .

```
T۱ = append (G۱, G۲, G۳, G۴, G۵, G۶, G۷, G۸, G۹);
```

مرحله سوم : تعریف ماتریسی که نحوه اتصالات بلوک های مختلف به همدیگر را تشریح کند. درایهٔ اول هر کدام از سطرهای این ماتریس باید نشان دهندهٔ شمارهٔ بلوکی باشد که می خواهیم اتصالات آنرا شرح دهیم. درایه های بعدی هم نشان دهندهٔ بلوکهایِ ورودی به این بلوک است . دقت شود که بلوک های ورودی را باید با علامتشان در نظر بگیریم. برای نمونه در همین مثال سطر پنجم به صورت « ۶- ۴ ۳ ۵ » تعریف شده است ، که در آن عدد اول نشان دهندهٔ بلوک پنجم و سایر اعداد بلوک های ورودی به این بلوک هستند. علامت بلوک 6 منفی است

یافتن تابع تبدیل حلقه بسته یک دیاگرام بلوکی با استفاده از MATLAB

```
Q=[۱ −۲ −۵  ۹
   ۲  ۱  ۸  ۰
   ۳  ۱  ۸  ۰
   ۴  ۱  ۸  ۰
   ۵  ۳  ۴ −۶
   ۶  ۷  ۰  ۰
   ۷  ۳  ۴ −۶
   ۸  ۷  ۰  ۰];
```

مرحله چهارم : مشخص کردن شماره بلوک های ورودی و خروجی

```
Inputs = ۹;
Outputs = ۷;
```

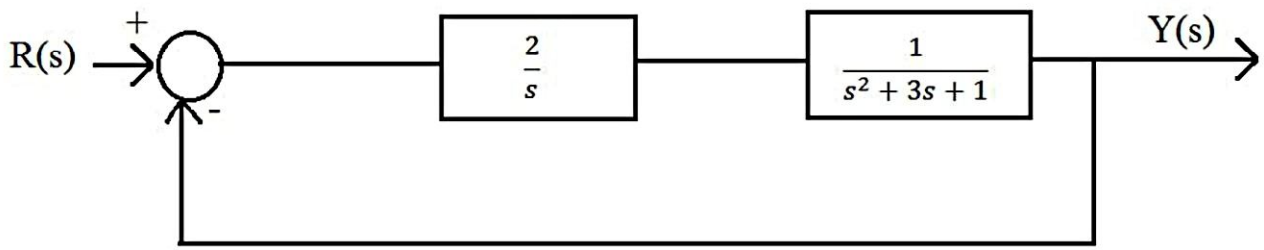
یافتن تابع تبدیل حلقه بسته یک دیاگرام بلوکی با استفاده از MATLAB

مرحله پنجم : استفاده از دستور connect()

```
Ts = connect (T۱, Q, Inputs, Outputs);
```

خروجی، تابع تبدیل حلقه بسته خواهد بود .

تمرین – تابع تبدیل حلقه بسته سیستم زیر را بیابید .



بخش سوم

تجزیه و تحلیل سیستم های کنترل بوسیله MATLAB

(قسمت دوم)

بدست آوردن نقشه صفر و قطبها

دستور pzmap(sys): صفرها و قطب های سیستم را مشخص میکند

```
[p,z]=pzmap(sys)
pzmap(sys)
```

مثال:

```
>> num=[۴ ۳];
>> den=[۱ ۶ ۵];
>> sys=tf(num,den)
>> [p,z]=pzmap(sys)
```

بدست آوردن پاسخ های زمانی سیستم های LTI

❖ **پاسخ ضربه :**

دستوراتِ زیر را می توان برای بدست آوردنِ پاسخ ضربهٔ واحد یک سیستم کنترل مورد استفاده قرار داد :

impulse (num, den)
impulse (*A, B, C, D*)
[*y, x, t*] = impulse (num, den)
[*y, x, t*] = impulse (*A, B, C, D*)
impulse (*sys*)

بدست آوردن پاسخ های زمانی سیستم های LTI

مثال: پاسخ ضربه سیستم زیر را بدست آورید.

$$T(s) = \frac{130}{s^2 + 15s + 130}$$

```
>>a=[۱۳۰];
>> b=[۱ ۱۵ ۱۳۰];
>> impulse(a,b)
```

تمرین : پاسخ ضربه سیستم زیر را بدست آورید.

$$T(s) = \frac{0.045}{s^2 + 0.025s + 0.045}$$

بدست آوردن پاسخ های زمانی سیستم های LTI

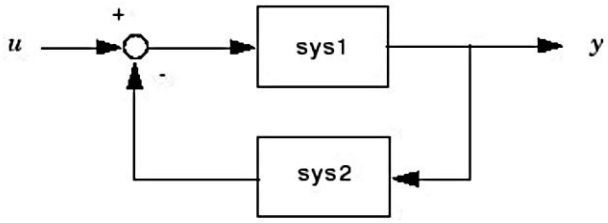
❖ **پاسخ پله :**

دستوراتِ زیر را می توان برای بدست آوردنِ پاسخ ضربهٔ واحد یک سیستم کنترل مورد استفاده قرار داد :

$$[y, x, t] = \textbf{step}\ [\text{num}, \text{den}]$$
$$[y, x, t] = \textbf{step}\ (A, B, C, D)$$
$$\textbf{step}\ (sys)$$

دستور فیدبک :

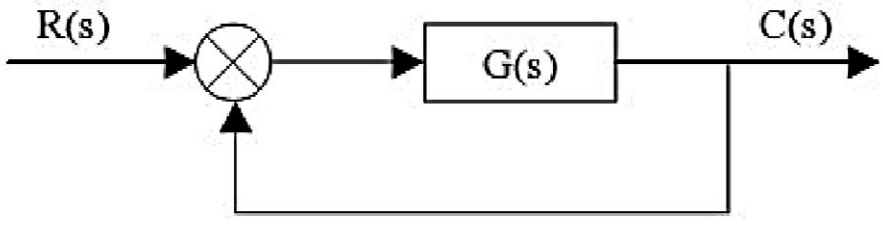
Feedback(sys۱,sys۲)



بدست آوردن پاسخ های زمانی سیستم های LTI

مثال : برای سیستم فیدبک واحد نشان داده شده پاسخ پله حلقه بستهٔ آن را تعیین کرده و رسم نمایید :

$$G(s) = \frac{30(s^2 - 5s + 3)}{(s + 1)(s + 2)(s + 4)(s + 5)}$$



بدست آوردن پاسخ های زمانی سیستم های LTI

حل –

```
>> numg = ۳۰*[۱ -۵  ۳]
>> deng = poly([-۱  -۲  -۴  -۵]);
>> G = tf(numg,deng);
>> T = feedback(G,۱)
>> step(T)
```

آیا این سیستم پایدار است ؟

بدست آوردن پاسخ های زمانی سیستم های LTI

تمرین –برای سیستم هایی با توابع تبدیل حلقه بازِ زیر :

$$G(S)=\frac{3s+2}{2s^3+4s^2+5s+1}$$

$$G(s)=\frac{50(s+1)}{s(s+3)(s+5)}$$

پاسخ پله حلقه بسته را رسم کنید.

بدست آوردن پاسخ های زمانی سیستم های LTI

❖ پاسخ به یک ورودیِ اختیاری :

پاسخ به یک ورودیِ اختیاری را می توان با استفاده از دستوراتِ زیر بدست آورد :

lsim (sys,u,t)

رسم مکان ریشه ها

✓ روش مکان ریشه ها ، تمامیِ قطب های حلقه بستهٔ سیستم را با استفاده از صفرها و قطب های حلقه بازِ سیستم ، به صورت ترسیمی نمایش می دهد.یک نمودار مکان ریشه از شاخه هایی تشکیل شده است که قطب ها را در برگرفته اند. بعبارت دیگر شاخه های مکان ریشه ها همان قطب های سیستم حلقه بسته هستند.

دستورات رسم مکان ریشه ها :

[r, k] = $r$locus (num, den)

[r, k] = $r$locus (A, B, C, D)

$r$locus (sys)

رسم مکان ریشه ها

مثال - *مکان ریشه های سیستم زیر را رسم کنید.*

$$G(s) = \frac{s+5}{s^2+7s+25}$$

```
>> num = [۱  ۵];
>> den = [۱  ۷  ۲۵];
>> rlocus(num, den);
```

تمرین -

$$G(s) = \frac{K(s+8)}{s^4+10s^3+48s^2+64s}$$

دیاگرام بود

دستورات رسم دیاگرام بود :

bode(num, den)

bode(*A, B, C, D*)

bode(sys)

مثال –دیاگرام بود تابع زیر را رسم نمایید

$$(a)\ G(s) = \frac{15}{s(s+3)(0.7s+5)}$$

دیاگرام بود

حل –

```
>> num = ۱۵;
>> den = conv([۱ ۰], conv([۱ ۳],[۰.۷ ۵]));
>> bode(num, den)
```

تمرین - دیاگرام بود سیستم زیر را رسم نمایید

$$G(s)=\frac{1}{s(s+1)}$$

$$(b)\ G(s) = \frac{(7s^3 + 15s^2 + 7s + 80)}{(s^4 + 8s^3 + 12s^2 + 70s + 110)}$$

نایکوئیست

دستورات رسم دیاگرام نایکوئیست :

nyquist (num, den)

nyquist (*A, B, C, D*)

nyquist (sys)

مثال- دیاگرام نایکوئیست سیستم زیر را رسم نمایید.

$$G(s) = \frac{1}{s^3 + 0.3s^2 + 5s + 1}$$

نایکوئیست

حل –

```
>> num = [۰ ۰ ۰ ۱];
>> den = [۱ ۰.۳ ۵ ۱];
>> nyquist(num,den)
```

تمرین – دیاگرام نایکوئیست سیستم زیر را رسم نمایید

$$G(s) = \frac{1280s + 640}{s^4 + 24.2s^3 + 1604.81s^2 + 320.24s + 16}$$

بخش سوم

تجزیه و تحلیل سیستم های کنترل بوسیله MATLAB

(قسمت سوم)

SIMULINK

ویژگی ها

۱- استفاده از دیاگرام بلوکی برای مدلسازی ، شبیه سازی و تحلیل سیستم های دینامیکی

۲- سادگی در استفاده

۳- دارای کتابخانه ِی وسیعی از بلوک های پرکاربرد در زمینه های مختلف و بلوک ها مورد نیاز در شبیه سازی سیستم های خطی ، غیر خطی و گسسته

بازکردن سیمولینک

محیط سیمولینک

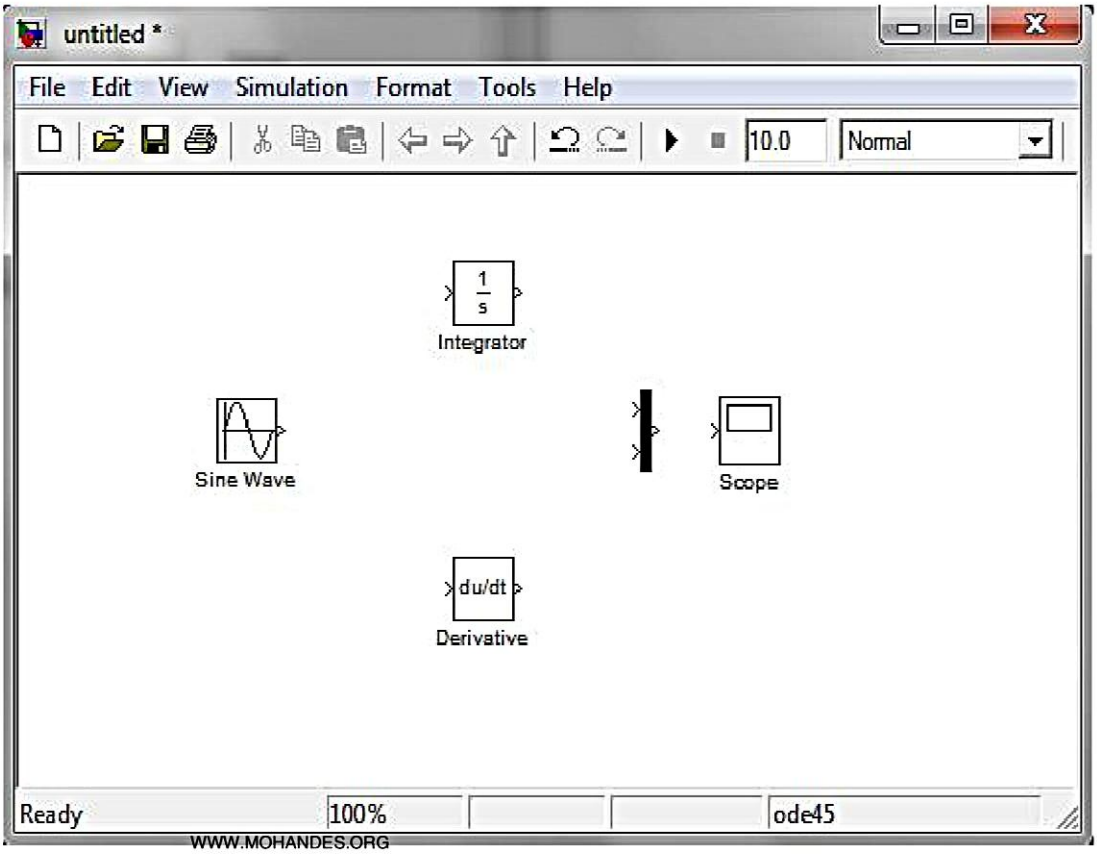
ایجاد یک مدل ساده

رسم نمودار یک موج سینوسی به همراه مشتق و انتگرال آن.

بلوک های زیر را از مسیر های تعیین شده به مدل اضافه نمایید.

- Simulink >> Sources >> Sin Wave
- Simulink >> Continuous >> Integrator
- Simulink >> Continuous >> Derivative
- Simulink >> Signal Routing >> Mux
- Simulink << Sinks << Scope

ایجاد یک مدل ساده



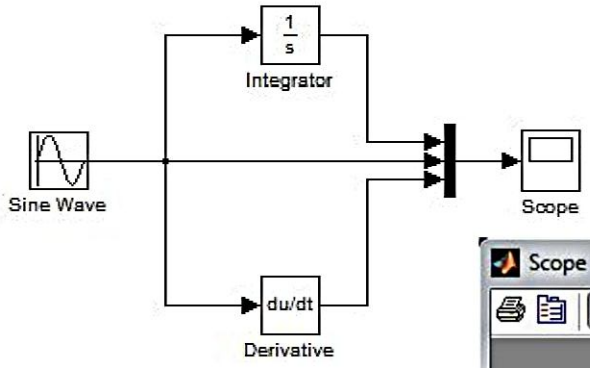
WWW.MOHANDES.ORG

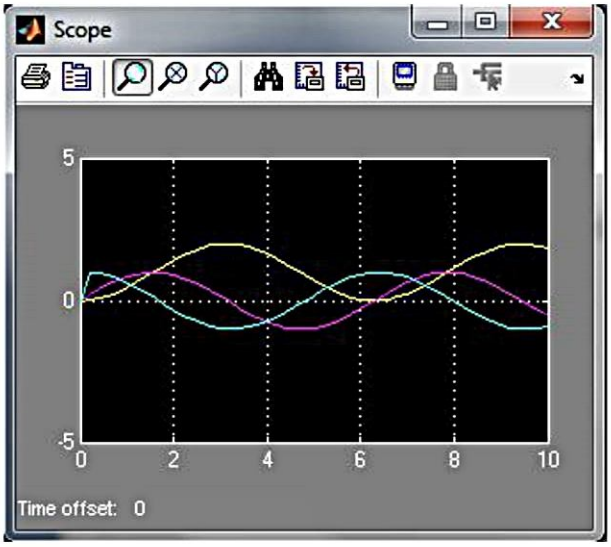
ایجاد یک مدل ساده

اتصال بلوک ها

اجرای مدل

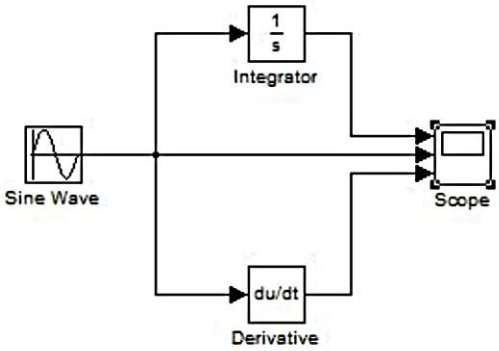
مشاهده خروجی

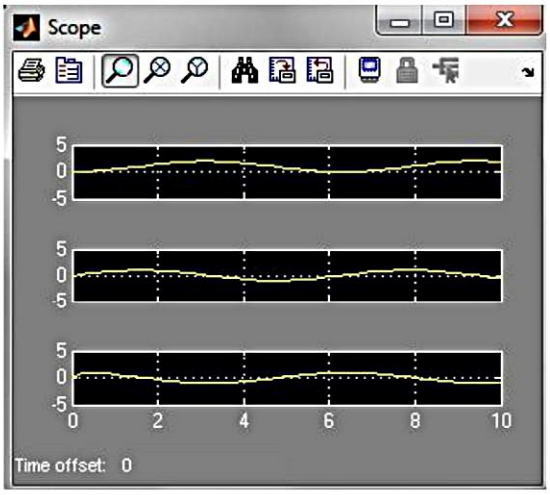




ایجاد یک مدل ساده

روش دیگر





رسم نمودار یک چند جمله ای

$$y = 2x^3 - x^2 + 3x - 1$$

بلوک های مورد نیاز را از مسیرهای زیر وارد کنید:

- Simulink >> Sources >> Clock
- Simulink >> Sources >> Constant
- Simulink >> Math Operations >> Sum
- Simulink >> Math Operations >> Gain
- Simulink >> Math Operations >> Math Function
- Simulink >> Sinks >> XY Graph

رسم نمودار یک چند جمله ای (تنظیمات)

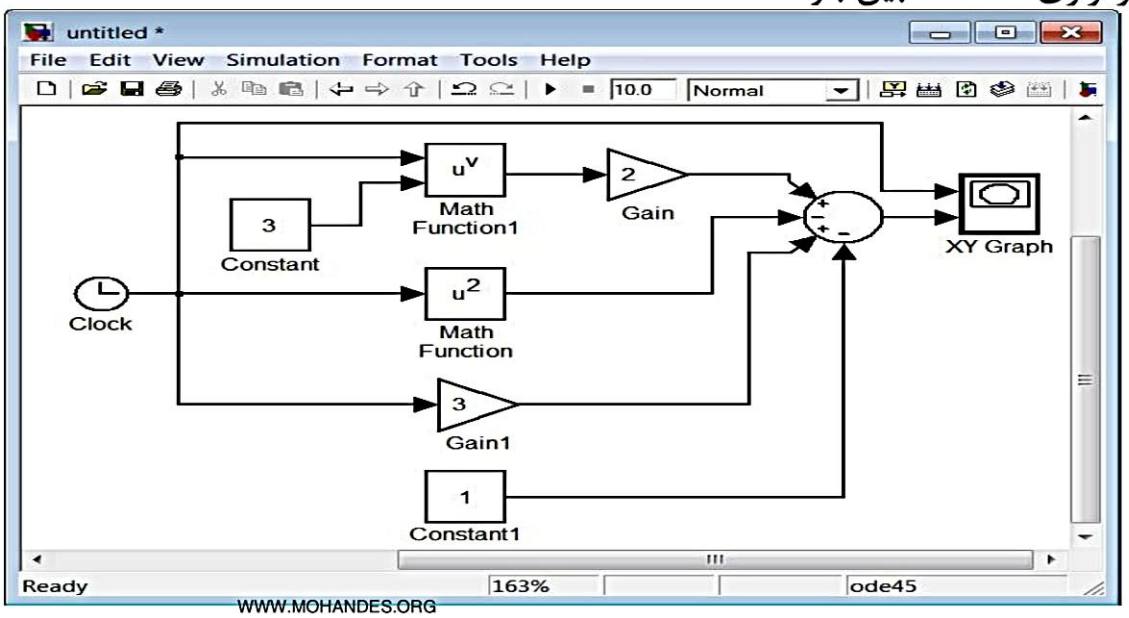
- روی بلوک Constant دوبار کلیک نموده و مقدار آن را ۳ قرار دهید
- روی بلوک Gain دوبار کلیک نموده و مقدار آن را ۲ قرار دهید
- روی بلوک Gain۱ دوبار کلیک نموده و مقدار آن را ۳ قرار دهید
- روی بلوک Math Function دوبار کلیک نموده و در پنجره تنظیمات باز شده نوع تابع را square انتخاب کنید
- روی بلوک Math Function۱ دوبار کلیک نموده و در پنجره تنظیمات باز شده نوع تابع را pow انتخاب کنید
- روی بلوک Sum دوبار کلیک نموده و در قسمت List of signs علامت‌های -+-+| را وارد کنید
- روی بلوک XY Graph دوبار کلیک نموده و حد پائین x را ۵- و حد بالای آن را ۵ و حد پائین y را ۱۵۰- و حد بالای آن را ۱۵۰ قرار دهید

رسم نمودار چند جمله ای

تنظیم زمان شبیه سازی : .

..Simulation >> Configuration Parameters

برقراری اتصالات بین بلوک ها :

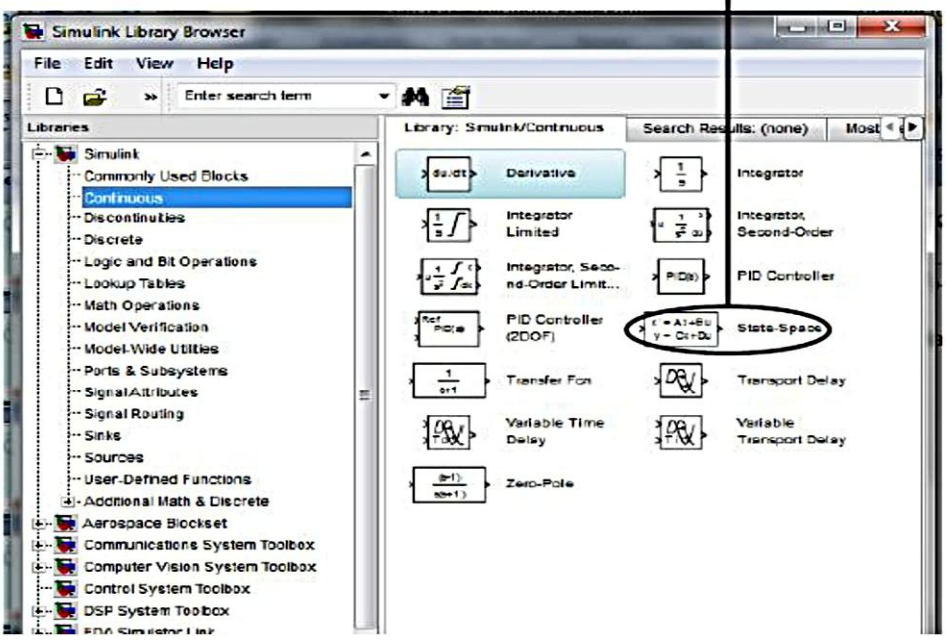


WWW.MOHANDES.ORG

برخی از بلوک های مورد استفاده در کنترل خطی

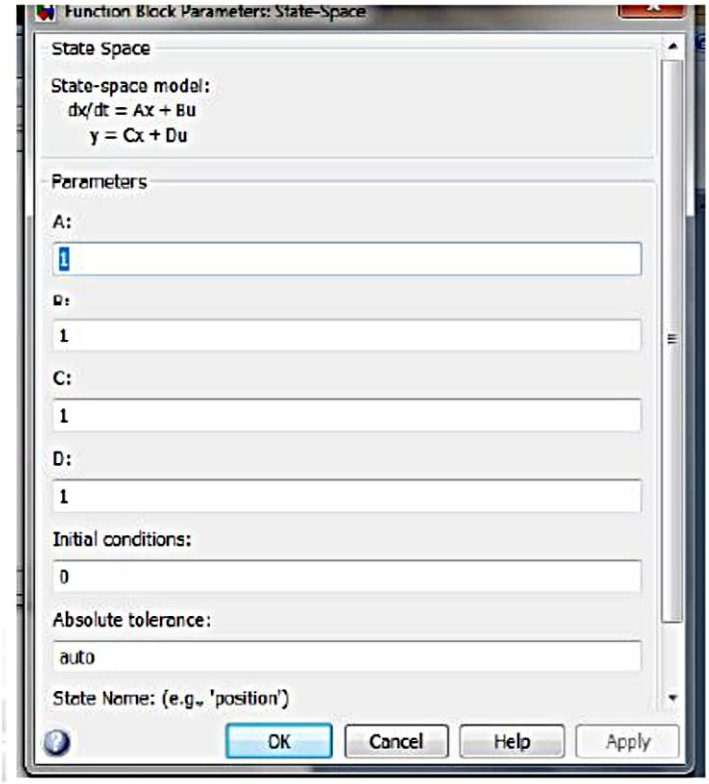
برای شبیه سازی یک سیستم توصیف شده با معادلات فضای حالت

Continues >> State-space



برخی از بلوک های مورد استفاده در کنترل خطی

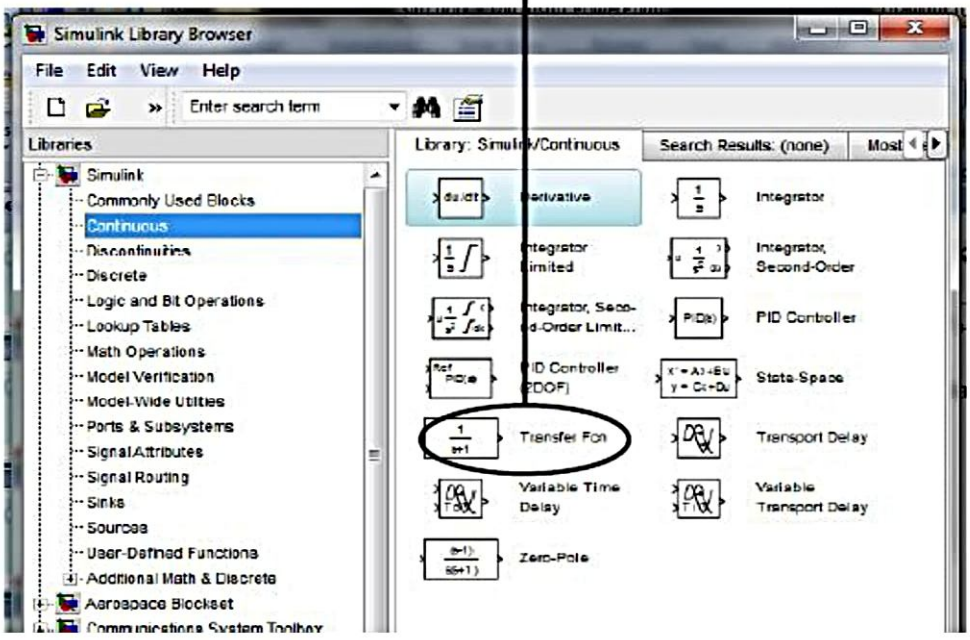
تنظیمات



برخی از بلوک های مورد استفاده در کنترل خطی

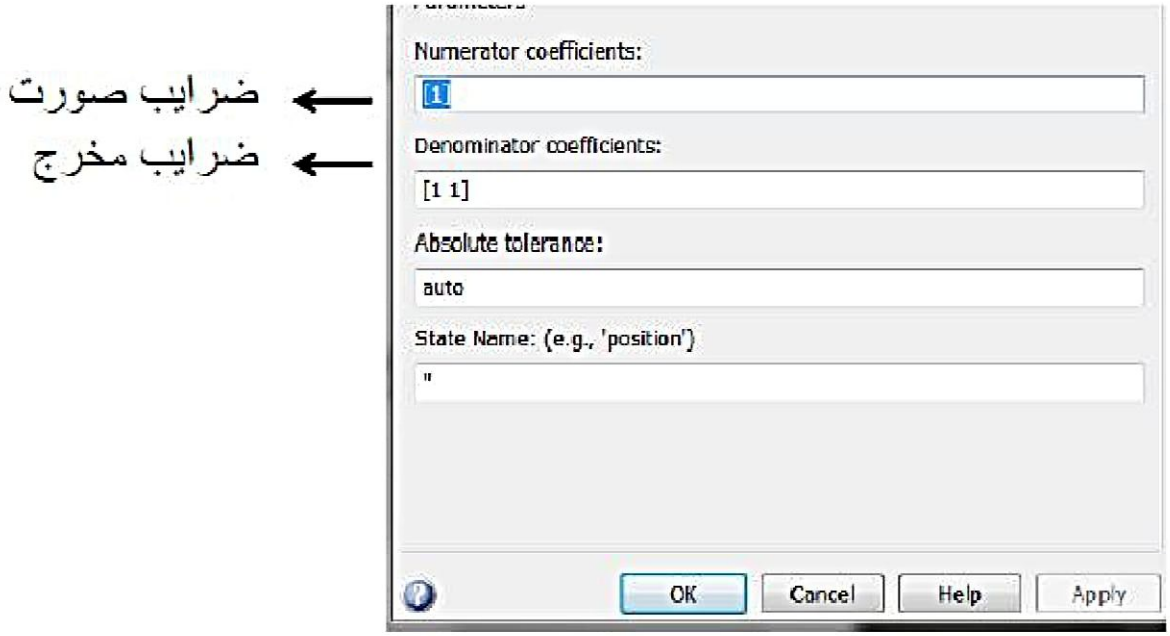
برای شبیه سازی یک سیستم توصیف شده با تابع تبدیل

Continues >> Transfer fcn



برخی از بلوک های مورد استفاده در کنترل خطی

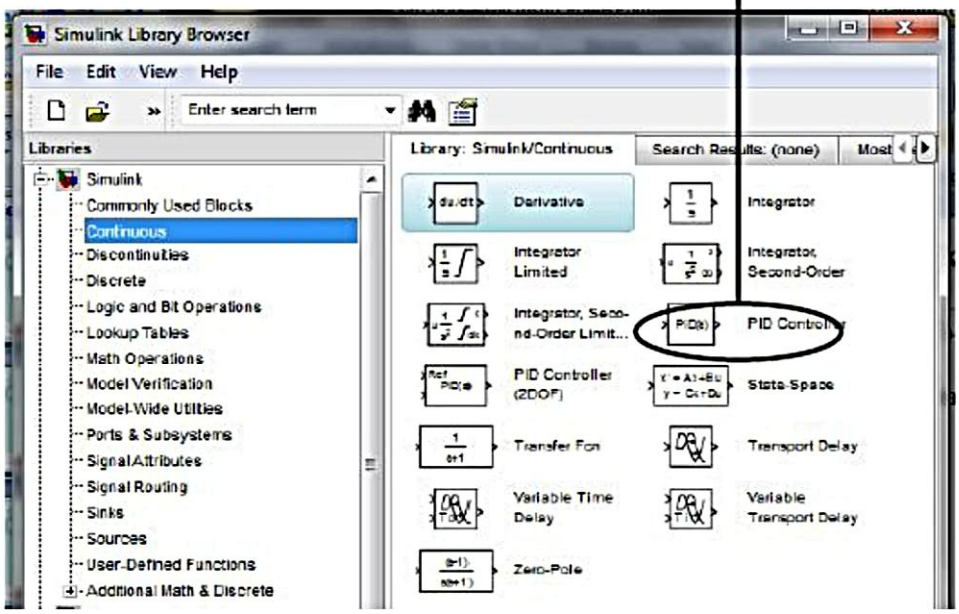
تنظیمات



برخی از بلوک های مورد استفاده در کنترل خطی

برای شبیه سازی کنترل کننده PID از بلوک زیر استفاده می کنیم :

Continues >> PID Controller

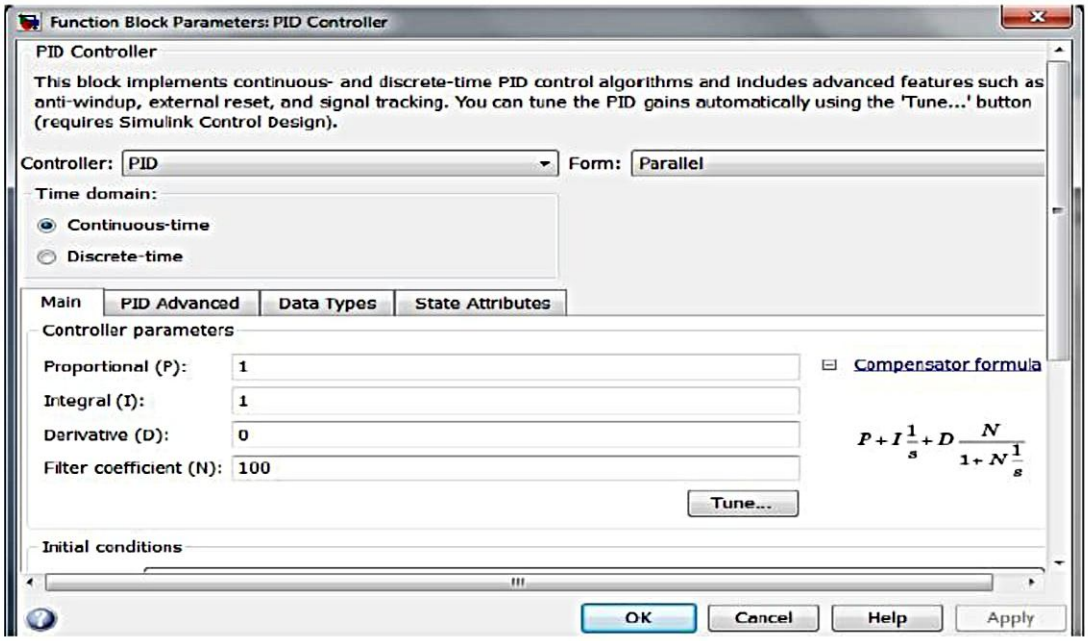


WWW.MOHANDES.ORG

برخی از بلوک های مورد استفاده در کنترل خطی

تنظیمات

$$P+I\frac{1}{s}+D\frac{N}{1+N\frac{1}{s}}$$



بدست آوردن پاسخ پله و ضربه یک سیستم

می‌خواهیم پاسخ پله و پاسخ ضربه سیستم توصیف شده با تابع تبدیل زیر را بدست آوریم :

$$G(s)=\frac{1}{2s^3+3s^2+s+1}$$

بلوک های مورد نیاز :

- Step << Sources << Simulink
- Scope << Sinks << Simulink
- Derivative << Continuous << Simulink
- Transfer Fcn << Continuous << Simulink

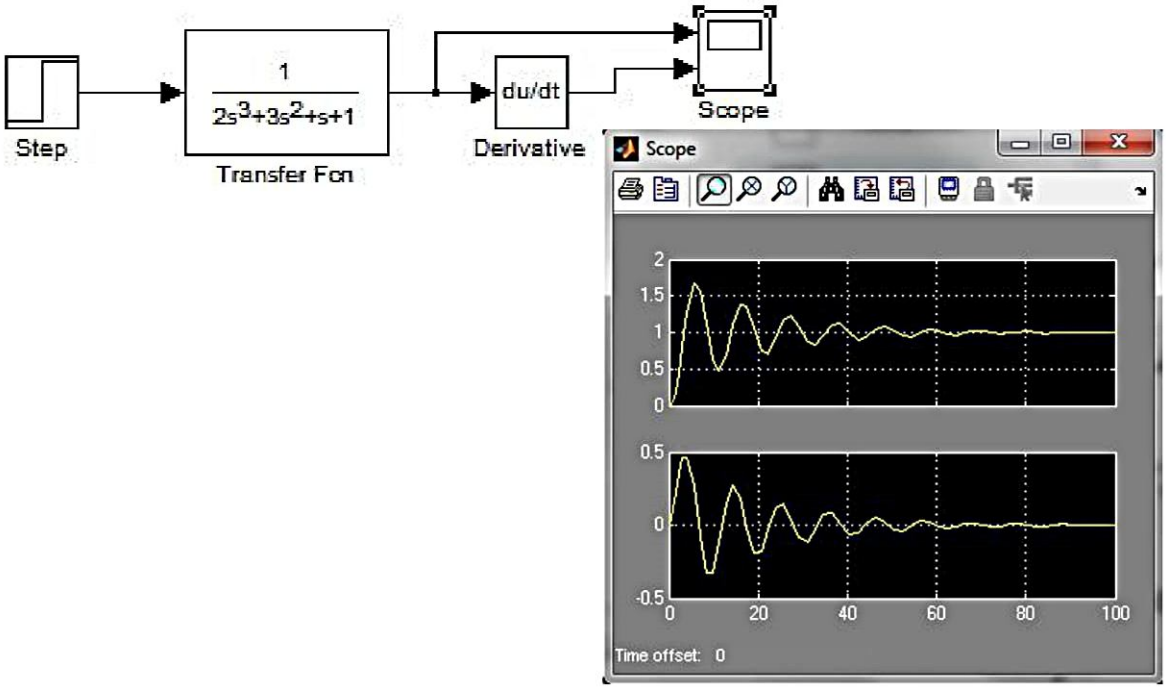
بدست آوردن پاسخ پله و ضربه یک سیستم

تنظیمات :

- روی بلوک Step دوبار کلیک نموده و مقدار Step time را ۰ قرار دهید
- روی بلوک Transfer Fcn دوبار کلیک نموده و مقدار Denominator coefficients را [۱ ۱ ۳ ۲] قرار دهید
- روی بلوکScope دوبار کلیک نموده و سپس به منویParameters رفته و تعداد محورها را برابر ۲ قرار دهید

بدست آوردن پاسخ پله و ضربه یک سیستم

اتصالات و پاسخ (زمان شبیه سازی را ۱۰۰ ثانیه بگیرید)



بدست آوردن پاسخ پله و ضربه یک سیستم

تمرین : پاسخ حلقه بسته (با فیدبک واحد منفی) تابع تبدیل بیان شده در مثال قبل را بدست آورید

$$G(s)=\frac{1}{2s^3+3s^2+s+1}$$

شبیه سازی معادلات دینامیکی (دیفرانسیلی)

فرض کنید معادله دینامیکی سیستمی به صورت زیر است :

y“+۰.۵y’+y=۳u

نکته : برای شبیه سازی چنین سیستمهایی به صورت معکوس عمل می کنیم و به جای استفاده از مشتق گیر از انتگرال گیر استفاده می کنیم .

بلوک های مورد نیاز:

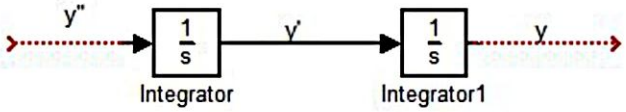
- Simulink >> Math Operations >> Sum
- Simulink >> Math Operations >> Gain
- Simulink >> Continuous >> Integrator

شبیه سازی معادلات دینامیکی (دیفرانسیلی)

ابتدا معادله را به صورت زیر باز نویسی می کنیم :

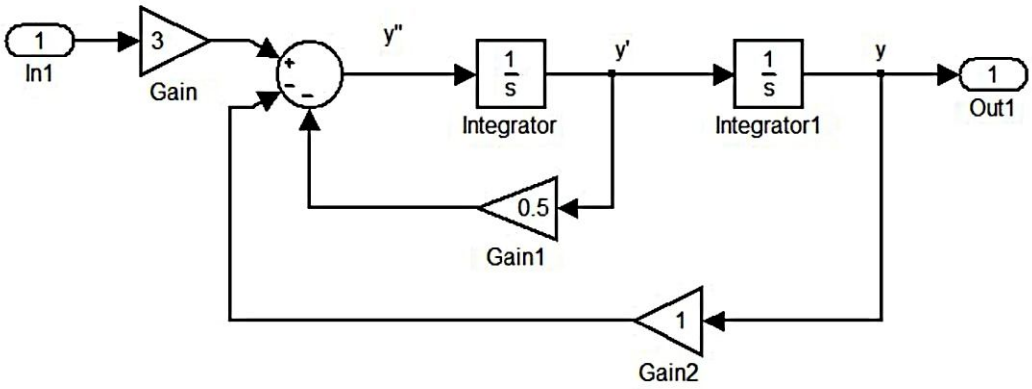
$$y'' = -0.5y' - y + 3u$$

دو انتگرال گیر را پشت سر هم قرار می دهیم . بدین ترتیب y" ، y' و y را خواهیم داشت :

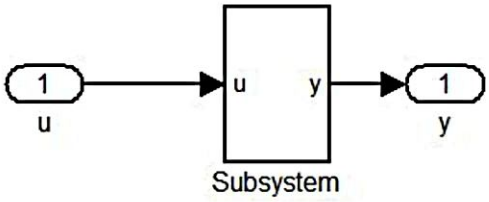


حال معادله بالا را می سازیم

شبیه سازی معادلات دینامیکی (دیفرانسیلی)



تولید Subsystem

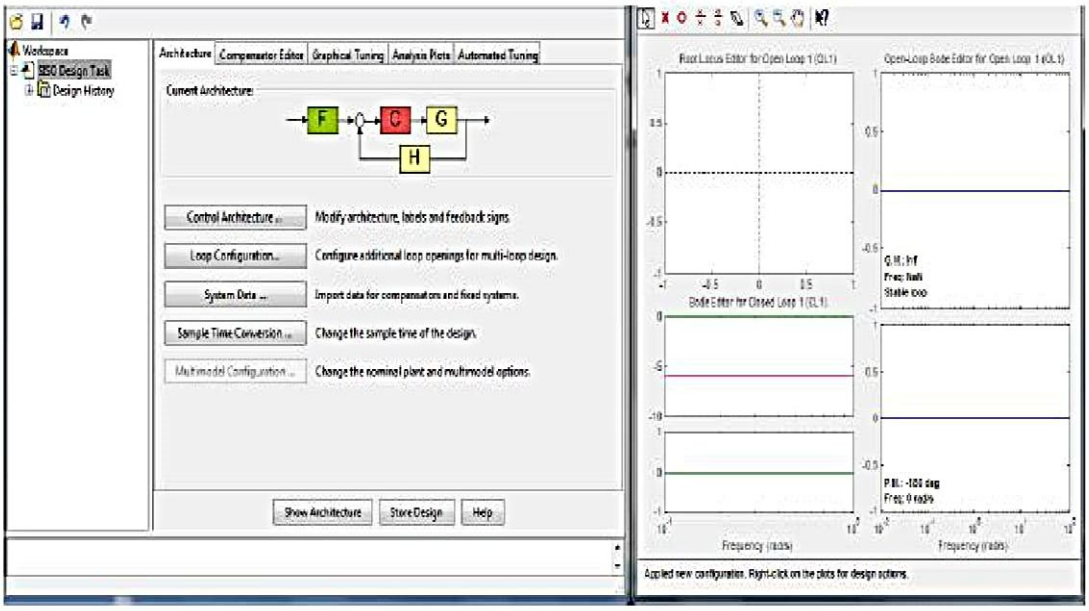


شبیه سازی معادلات دینامیکی (دیفرانسیلی)

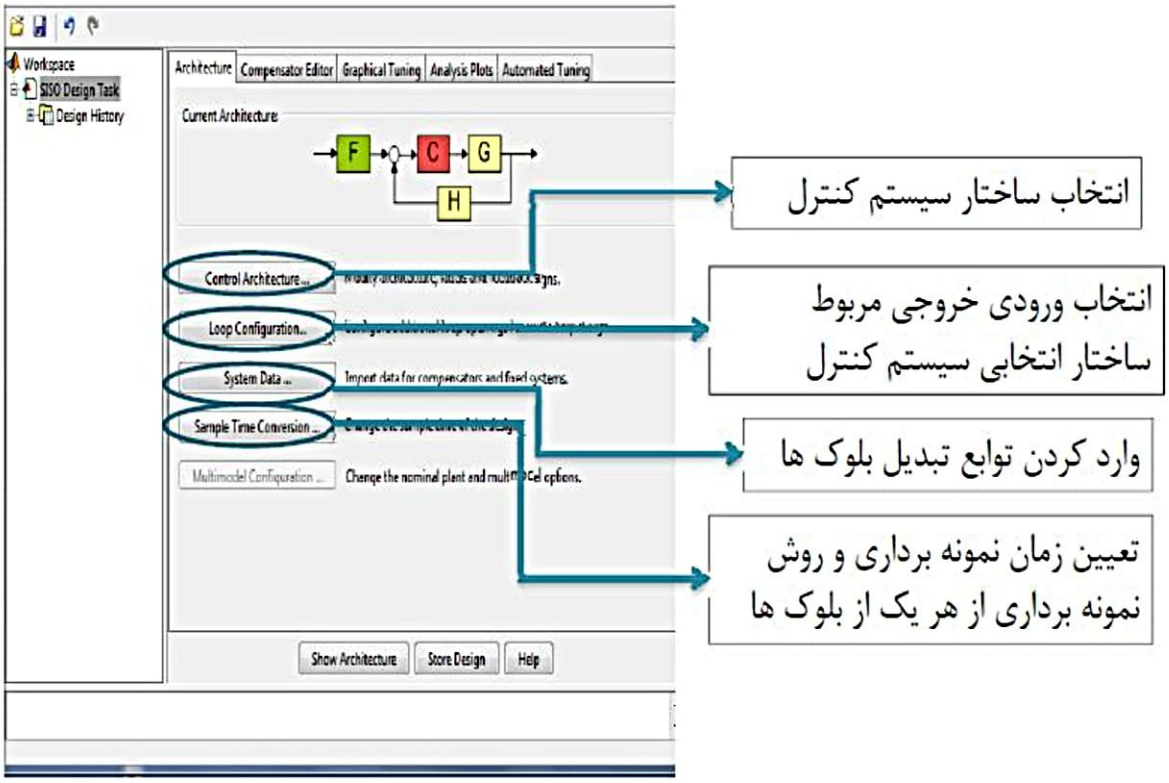
تمرین : پاسخ پله حلقه باز و حلقه بسته (با فیدبک واحد منفی)
سیستم قبل را بدست آورید

SISOTOOL

- با وارد کردن دستور sisotool در command window منتظر بمانید تا GUI مربوط به جعبه ابزار کنترل باز شود:

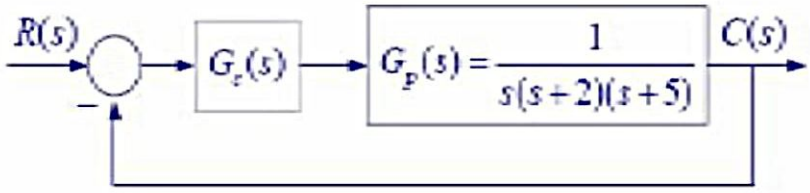


SISOTOOL



SISOTOOL

مثال – می خواهیم سیستم زیر را در sisotool شبیه سازی کرده و برای آن کنترلر طراحی نماییم .



❑ حل –

۱- ابتدا سیستم اصلی را در command window تعریف می کنیم:

```
>> Gp=tf(1,[1 7 10 0])
Transfer function:
        1
------------------
s^3 + 7 s^2 + 10 s
```

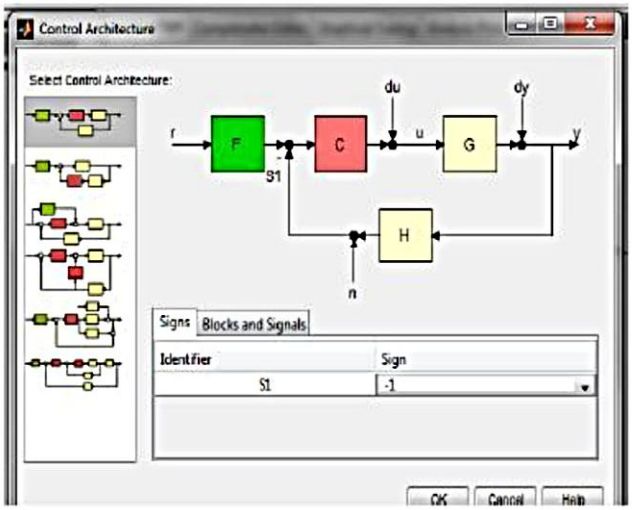
۲- sisotool را باز کنید:

```
>> sisotool
```

SISOTOOL

۳- ساختار سیستم کنترلی را در پنجره ی Control and Estimation Tools Manager به طریق زیر انتخاب کنید:

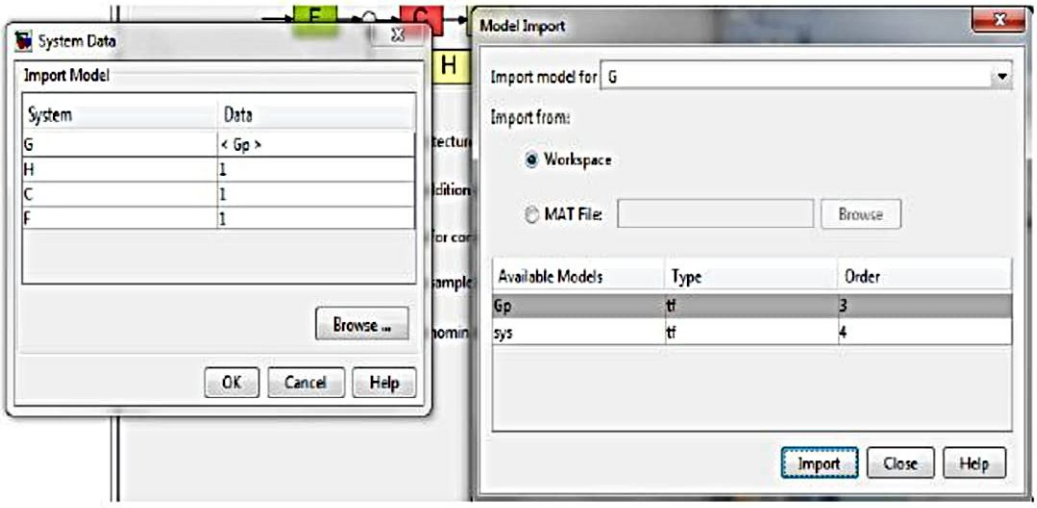
Architecture → Control Architecture



SISOTOOL

۴- به صورت زیر سیستم تعریف شده در command window را وارد ساختار انتخابی برای سیستم کنید:

System Data → Browse



- حال صفحه ی SISO Design for SISO Design Task را نگاه کنید. چه می بینید؟!!

SISOTOOL

۵- برای افزودن کنترل کننده به جای بلوک C در ساختار کنترلی انتخابی به صورت زیر عمل کنید:

Compensator Editor →

در صفحه یPole/Zeros راست کلیک کنید

→Add pole/zero →

گزینه ی مورد نظر خود را انتخاب کنید.

تغییرات حاصل از انتخاب شما بلافاصله در صفحه ی SISO Design for SISO Design Task نشان داده می شود.