



کاربرد نرم افزارهای تحلیلی در مهندسی خودرو

محمدحسن آقانوری

تهیه کننده



نقشه راه درس کاربرد نرم افزار در مهندسی خودرو



مبانی و مفاهیم پایه

- ✓ مفاهیم اولیه برنامه نویسی
- ✓ معرفی نرم افزار
- ✓ توابع و ماتریس ها

رسم نمودار

- ✓ رسم دو بعدی
- ✓ رسم سه بعدی
- ✓ پروژه ۱

معادلات دیفرانسیل

- ✓ محاسبات عددی
- ✓ پروژه ۲

سیمولینک و بلوکها

- ✓ مبانی سیمولینک
- ✓ آشنایی با ابزارهای تحلیل خودرو
- ✓ پروژه ۳

روش سنجش و ارزشیابی

✓ پرسشهای شفاهی و کار کلاسی

✓ آزمون های کتبی، عملی با محوریت حل مسئله

✓ انجام کار در محیطهای شبیه سازی شده، تولید نمونه کار

✓ حضور و مشاهده رفتار مسئولیت پذیری، رعایت اخلاق

حرفه ای.

4	تمرینات کلاسی
3	پروژه ۱
4	پروژه ۲
3	امتحان کتبی
7	امتحان عملی
21	مجموع

مهارتهای عمومی و تخصصی مورد انتظار

❖ انتظار میرود بتواند معادلات مرتبط با مهندسی خودرو را برنامه نویسی کرده و در حد معمول با نرم افزار MATLAB کار کند .



روش ارائه

بر مبنای آموزش تکنیک های کاربردی و توانایی انجام پروژه های کاربردی

- ارائه نظری همراه تکنیک های کاربردی
- تمرین و تکرار و روحیه حل مسئله
- کارگاه و آزمایشگاه و پروژه های پژوهشی و گروهی

فصل اول

مفاهیم اولیه برنامه

آشنایی با ساختار برنامه در MATLAB

تعاریف و اصطلاحات:

✓ تعریف برنامه نویسی: برنامه نویسی فرآیند ایجاد مجموعه ای از دستورالعمل ها است که به رایانه می گوید چگونه یک کار را انجام دهد.

✓ کاربردهای برنامه نویسی: تحلیلی، اپلیکیشن (وب، دسکتاپ، سخت افزار، بازی، موبایل)

✓ الزامات برنامه نویسی: زبان، شناخت الگوریتم، ریاضیات، جستجو، انگیزه

✓ آشنایی با زبان ها و نرم افزارهای موجود در برنامه نویسی

✓ کاربرد برنامه نویسی در مهندسی خودرو و نرم افزارهای مرتبط با آن

✓ نرم افزار MATLAB

✓ کاربردها، ویژگی ها، نقاط قوت و ضعف

✓ تفاوت نسخه ها و ورژن ها a,b (Stability, Features)

✓ سیستم عامل مورد نیاز، ابزارهای کمکی

✓ طریقه نصب

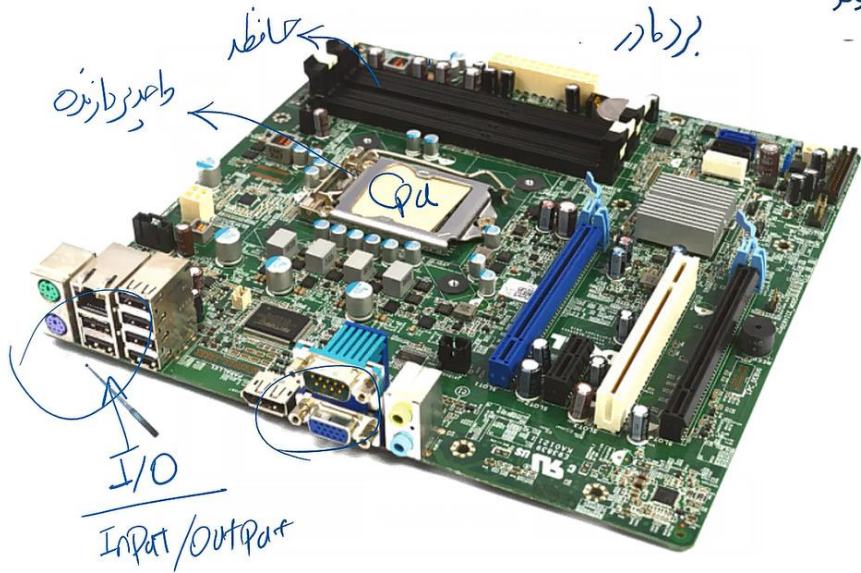
✓ پنجره های کاربردی Command window, Workspace, current folder

✓ Editor, Debugging, comments, M-file



دوره مبانی برنامه نویسی کامپیوتر

معماری کامپیوتر



How important is MATLAB Simulink in automobile industries?



MATLAB Simulink is a widely used software tool in the automobile industry for the development and testing of control systems, powertrain systems, and other automotive systems. It allows engineers to model and simulate complex systems, test different control algorithms, and analyze the results in a virtual environment before implementing them on physical vehicles. This can help to reduce development time and costs, as well as improve the performance and safety of the final product. Additionally, MATLAB Simulink has a large library of pre-built blocks, which can be used to model common automotive systems, such as engines, transmissions, and electric powertrains. Overall, it is a valuable tool for engineers in the automobile industry.

Which programming language is more useful for automobile engineer? Matlab or Python?

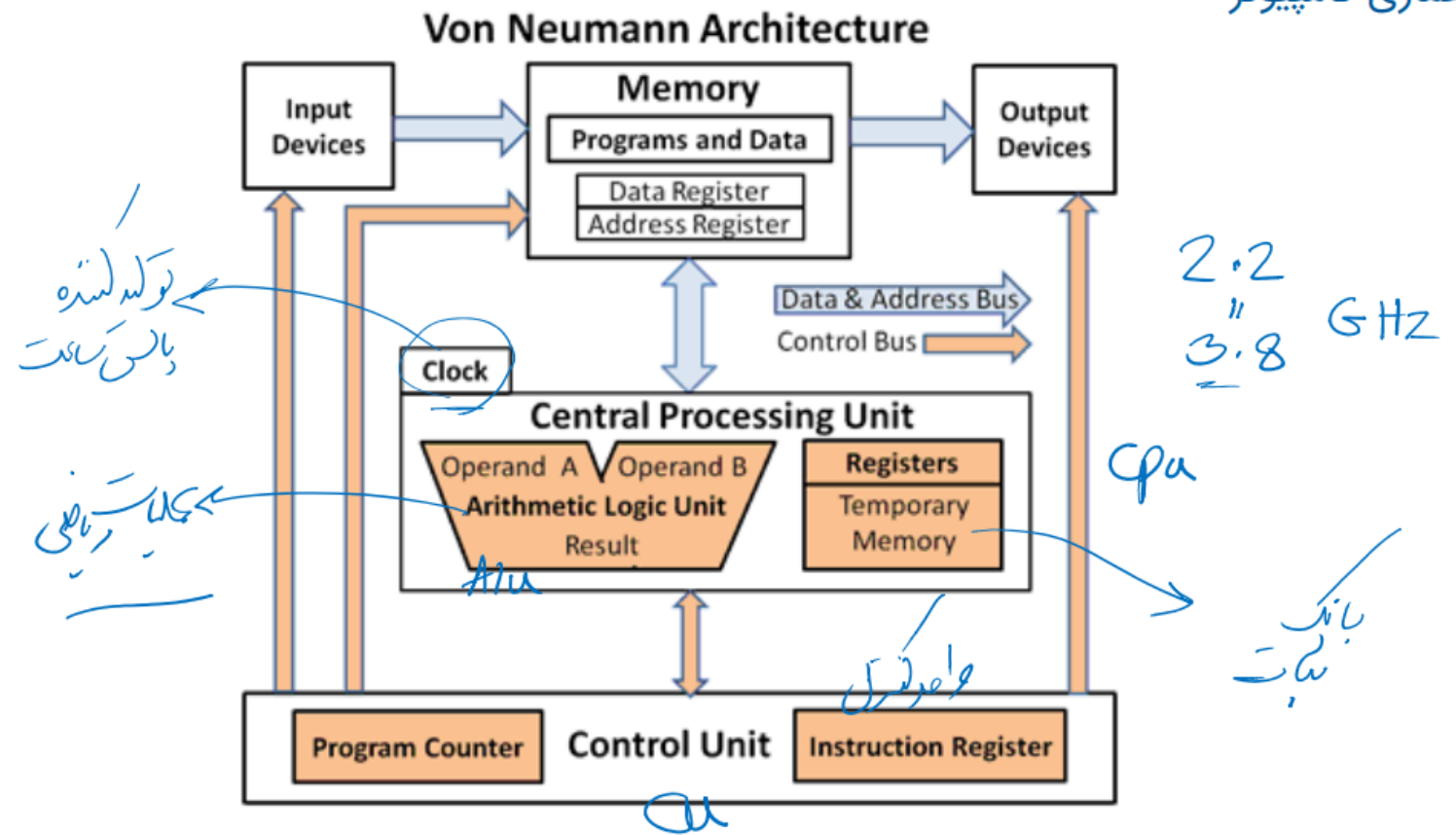
 ChatGPT

Both Matlab and Python are popular programming languages in the field of automobile engineering. Matlab is often used for control systems and signal processing, while Python is commonly used for data analysis and visualization. Both languages have libraries and toolboxes specifically designed for automobile engineering, such as the Vehicle Dynamics Blockset in Matlab, and the PyVehicle library in Python. Ultimately, the choice between Matlab and Python will depend on the specific needs of your project and your personal preferences as an engineer. Would you like me to look up more information on the two languages?

 Like

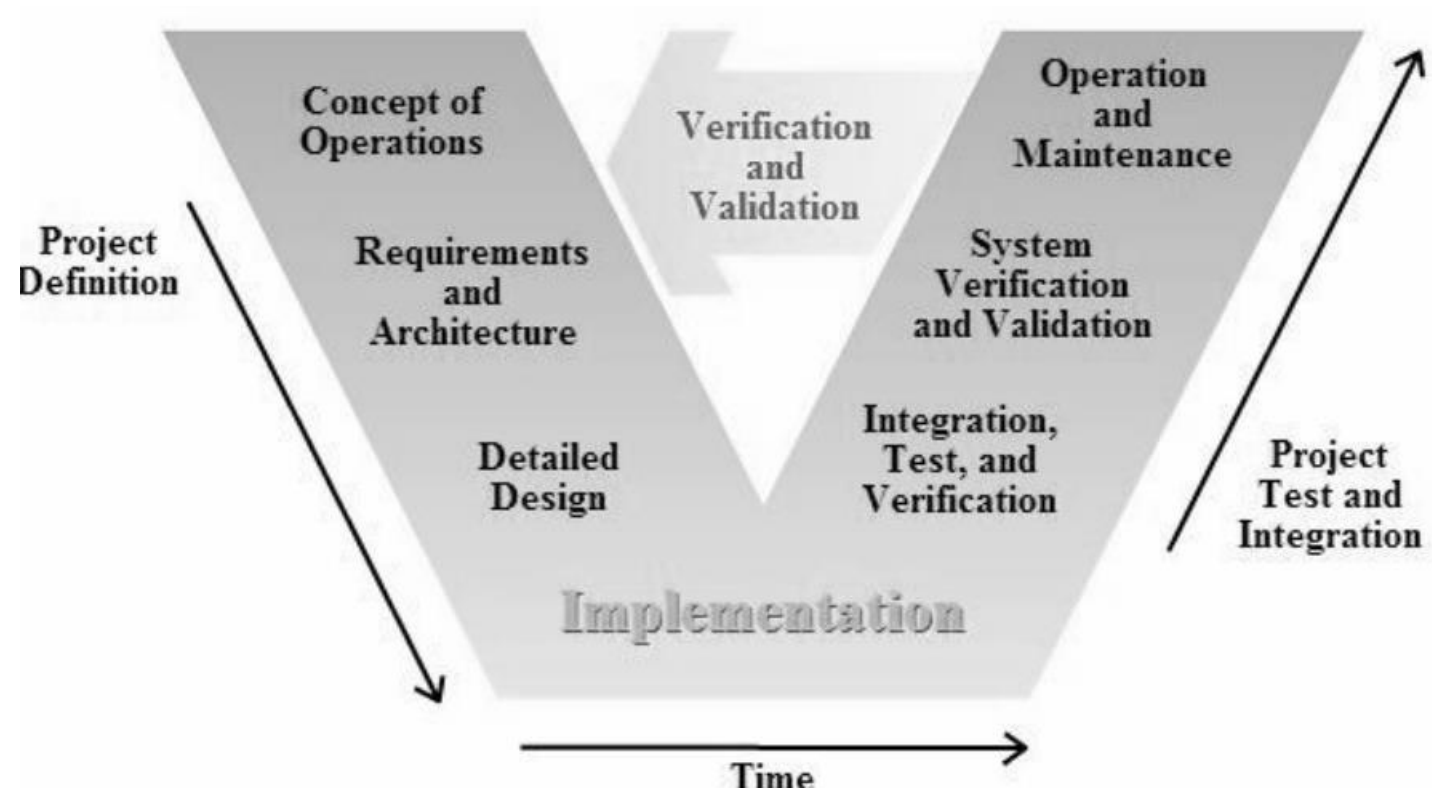
Dislike

معماری کامپیوتر



برگرفته از جزوه برنامه نویسی دکتر حامد شهبازی- استاد دانشگاه اصفهان

V-shaped development model



control logic development

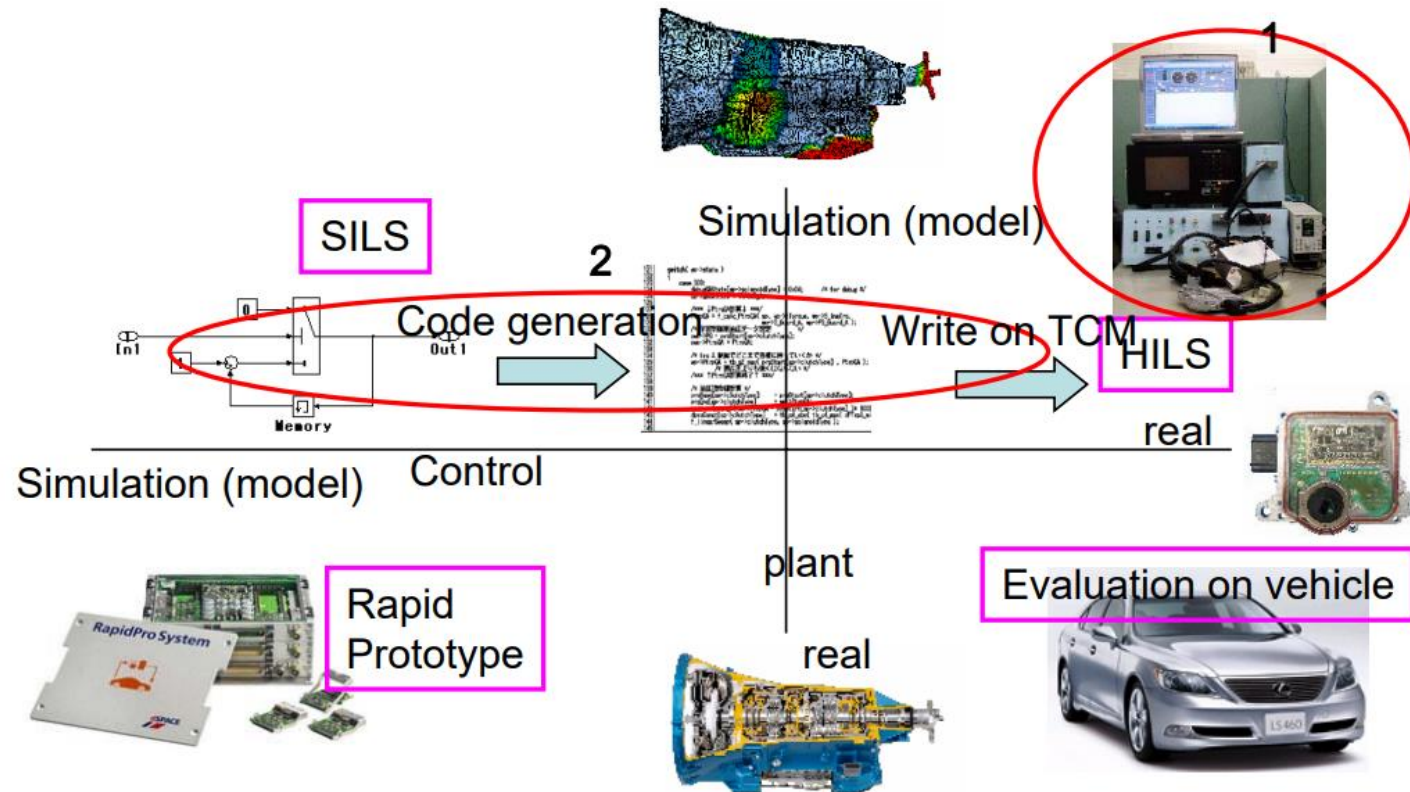


Table 1-1. Some Basic Commands Used in MATLAB

Function	Explanation	Example
help	Returns information about the specified command	>>help clc
demo	Shows the explanation of any subject in MATLAB	>>demo
save	Saves the workspace variables to the named file	>>save my_var
diary on	Starts recording the session	>>diary on
diary off	Stops recording the session and saves it to a diary file	>>diary off
exit	Terminates MATLAB	>>exit
quit	Terminates MATLAB	>>quit
clc	Clears the screen	>>clc
clear	Clears all variables or any specified variable from the workspace	>>clear all
who	Displays all the variables in the workspace	>>who
whos	Displays all the variables in the workspace with sizes and types	>>whos

Example 1-2. Find the result of $5 - \frac{8}{3} + \cos(\pi) - \frac{10}{e^2} + \sqrt{7}$.

Solution 1-2. The following code will find the solution.

```
> 5-8/3+cos(pi)-10/exp(2)+sqrt(7)
ans =
    2.6257
>
```

Table 1-8. Trigonometric Functions in Radians

Command	Definition
sin	Sine
cos	Cosine
tan	Tangent
cot	Cotangent
sec	Secant
csc	Cosecant

Table 1-9. Trigonometric Functions in Degrees

Command	Definition
sind	Sine in degrees
cosd	Cosine in degrees
tand	Tangent in degrees
cotd	Cotangent in degrees
secd	Secant in degrees
cscd	Cosecant in degrees

❖ دستورات اولیه:

- ✓ اعداد و توابع خاص
- ✓ عملگرهای اولیه و کاربرد آن
- ✓ عملگرهای شرطی
- ✓ اولویت بندی عملگرها

Command Window

```
>> ali = 20
```

```
ali =
```

```
20
```

- They should start with a letter.
- They can contain numbers and underscores.
- They can be a maximum of 63 characters long (the `namelengthmax` command can be used to check this).
- They should not be a keyword adopted in the MATLAB language.

Table 2-3. *Relational Operators*

Operator	Description	Example
>	Greater than	3 > 2
>=	Greater than or equal to	4 >= 3
<	Less than	2 < 3
<=	Less than or equal to	2 <= 3
==	Equal to	4 == 4
~=	Not equal to	2 ~= 3

❖ دستورات اولیه:

- ❖ اعداد و توابع خاص
- ❖ عملگرهای اولیه و کاربرد آن
- ❖ عملگرهای شرطی
- ❖ اولویت بندی عملگرها

Table 1-3. Special Matrix Functions in MATLAB

Function	Explanation	Example
eye	Creates an identity matrix	eye(5)
ones	Creates a matrix where all the elements are ones	ones(5)
zeros	Creates a matrix where all the elements are zeros	zeros(5)
diag	Extracts or displays the diagonal part of a matrix	diag(A)
sparse	Creates a matrix where all the elements are zeros	sparse(5,5)
spdiags	Extracts all diagonals from the matrix	sparse(A)
speye	Creates an identity sparse matrix	speye(5,5)
rand	Creates a randomly generated matrix with values between 0 and 1	rand(5)
magic	Creates magic matrices	magic(3)

Table 1-7. Additional Math Functions

Function	Explanation
fix	Rounds number toward zero
floor	Rounds number toward minus infinity
ceil	Rounds number toward plus infinity
round	Rounds number toward nearest integer
mod	Shows remainder after dividing
rem	Shows remainder division
sign	Returns -1,0, or 1 (Signum function)

Table 1-6. Some Math Functions

Function	Explanation
exp	Exponential function
log	Natural logarithm function
log10	Common logarithm function in base 10
reallog	Natural logarithm of a real number
sqrt	Square root of a number
nthroot	Real nth root of a real number

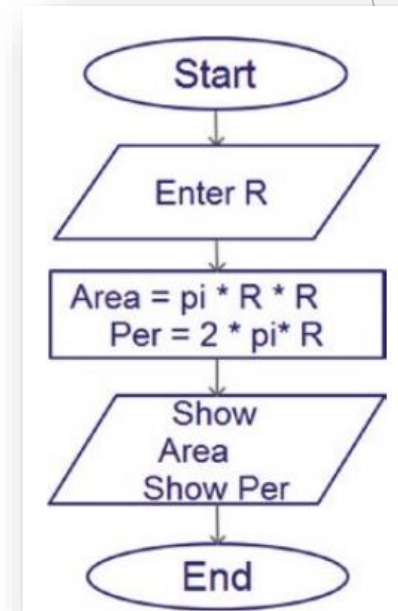
$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow 3 \times 2 \text{ matrix 6 elements}$$

- ✓ Det
- ✓ .*
- ✓ :
- ✓ linspace

Table 1-3. Special Matrix Functions in MATLAB

Function	Explanation	Example
eye	Creates an identity matrix	eye(5)
ones	Creates a matrix where all the elements are ones	ones(5)
zeros	Creates a matrix where all the elements are zeros	zeros(5)
diag	Extracts or displays the diagonal part of a matrix	diag(A)
sparse	Creates a matrix where all the elements are zeros	sparse(5,5)
spdiags	Extracts all diagonals from the matrix	sparse(A)
speye	Creates an identity sparse matrix	speye(5,5)
rand	Creates a randomly generated matrix with values between 0 and 1	rand(5)
magic	Creates magic matrices	magic(3)

❖ محاسبات سمبلیک



```

%this calculates the area and perimeter of a circle
Radius = input('Please enter the radius\n');
AreA = pi*(Radius^2);
Per= 2*pi*Radius;
disp(['Area of the circle is= ',num2str(AreA)]);
fprintf('Perimeter of the circle is = % d\n',Per);
  
```

برنامه نویسی:

Algorithm

الگوریتم ✓
دستورالعملها گفته می شود که مراحل انجام کاری را نشان می دهد. اگر این مراحل به ترتیب اجرا شوند کار به درستی انجام می شود

الگوریتم ← انباری

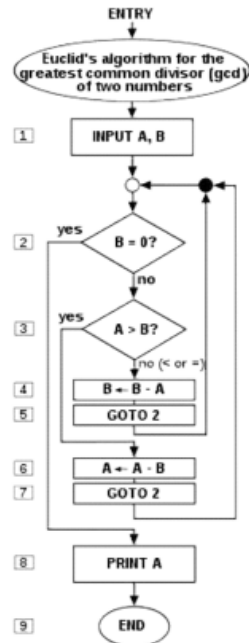




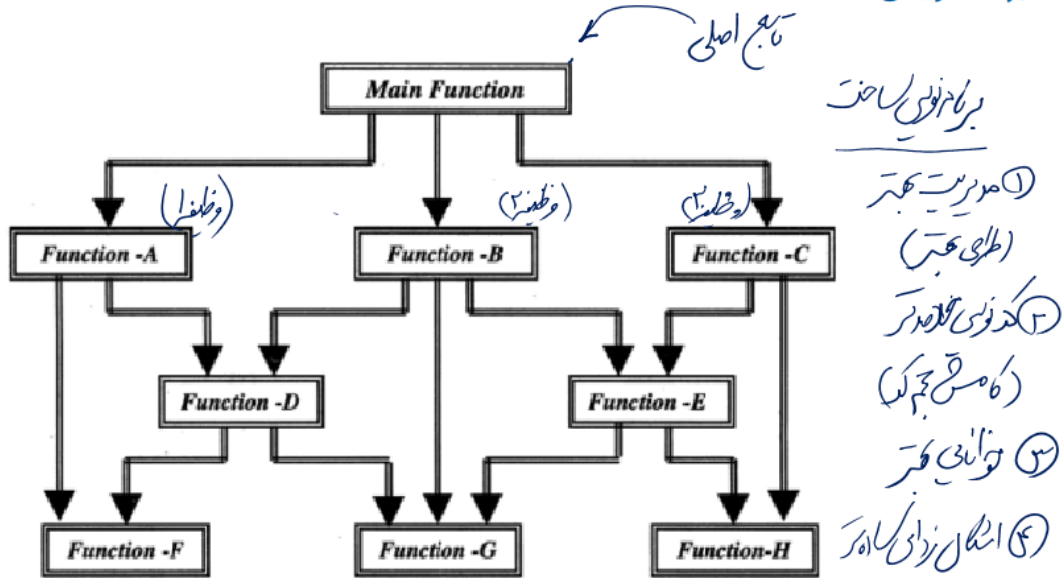


Table 2-1. Most Frequently Used Flowchart Symbols

	Used for initiating and finishing the program
	Used to indicate calculations, or assigning
	Used to indicate decision processes
	Used to indicate input or output processes

برنامه نویسی:



```

+27 dayere_j3.m x dayere.m x main.m x prim.m x dayerehh.m x mainn.m x
1
2 rad = input("yek adad vared konid");
3 dayere_j3(rad);
  
```

```

+27 dayere_j3.m x dayere.m x main.m x prim.m x dayerehh.m x mainn.m x +
1 function [mohit, masaahat] = dayere_j3(shoa)
2 shoa = input('لطفا یک عدد وارد کنید \n ');
3 masaahat = pi * (shoa^2);
4 mohit = 2*pi*shoa;
5 % disp(['Area of this circle is =', num2str(masaahat)]);
6 fprintf('the primeter of this circle is = %d\n', mohit);
7 end
  
```

Loops and Conditional Statements

R2021a

Control flow and branching using keywords, such as `if`, `for`, and `while`

Within any program, you can define sections of code that either repeat in a loop or conditionally execute. Loops use a `for` or `while` keyword, and conditional statements use `if` or `switch`. Additional keywords provide finer control over the program flow.

MATLAB Language Syntax

<code>if, elseif, else</code>	Execute statements if condition is true
<code>switch, case, otherwise</code>	Execute one of several groups of statements
<code>for</code>	for loop to repeat specified number of times
<code>while</code>	while loop to repeat when condition is true
<code>try, catch</code>	Execute statements and catch resulting errors
<code>break</code>	Terminate execution of for or while loop
<code>return</code>	Return control to invoking script or function
<code>continue</code>	Pass control to next iteration of for or while loop
<code>pause</code>	Stop MATLAB execution temporarily
<code>parfor</code>	Parallel for loop
<code>end</code>	Terminate block of code or indicate last array index

❖ کنترل ها در برنامه نویسی

```
if comparison
    statement
elseif comparison
    statement
else
    statement
end
```

```
%Example2p5
% This program asks your age
Age = input('Please enter your age\n');
if Age < 6
    fprintf('Maybe NO School \n');
elseif Age <13
    fprintf('Middle School \n');
elseif Age <18
    fprintf('High School \n');
elseif Age <26
    fprintf('Maybe University \n');
else % for other possibilities
    fprintf('Professional \n')
end
```


❖ کنترل ها در برنامه نویسی

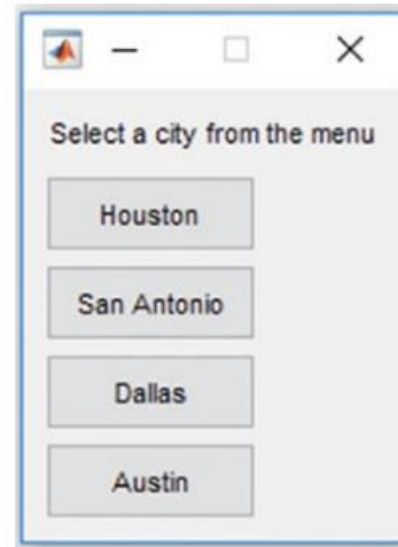
```
switch variable
    case option1
        In case of option1 do these
    case option2
        In case of option2 do these
    otherwise
        If none of the cases is applied, do this
end
```

```
%Example2p7
%This program has a switch-case illustration
Money=input('How much money you have? \n');
switch Money
    case 20
        fprintf('You can eat Chicken \n');
    case 30
        fprintf('You can eat Lobster \n');
    case 40
        fprintf('You can eat Beef \n');
    case 50
        fprintf('You can eat Fish \n');
    otherwise % for other possibilities
        fprintf('No match with the menu \n');
end
```

❖ کنترل ها در برنامه نویسی

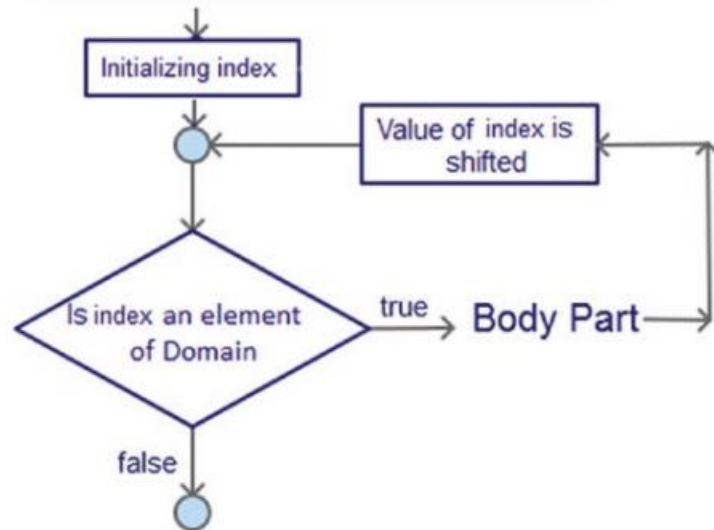
```
Name = menu('memutitle','option1','option2',...,'option')
```

```
%Example2p8
%This code uses menu function
city=menu('Select a city from the menu' ,...
    'Houston','San Antonio','Dallas','Austin');
switch city
    case 1
        fprintf('$450 \n');
    case 2
        fprintf('$550 \n');
    case 3
        fprintf('$650 \n');
    case 4
        fprintf('$750 \n');
end
```



❖ کنترل ها در برنامه نویسی

```
for starting_variable = Domain  
    codes to be executed  
end
```



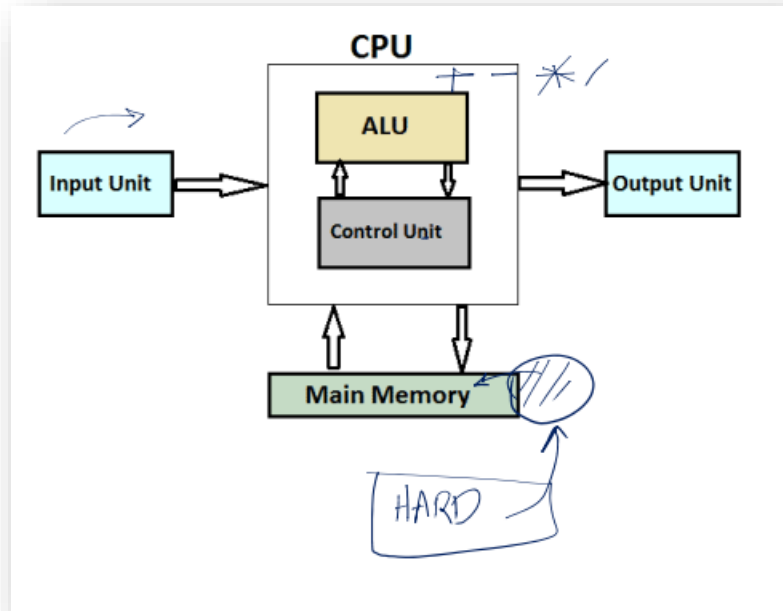
Flowchart of a for loop

$$\sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots = \ln 2 = 0.693147.$$

Example2p9.m

```
%Example2p9  
%This code calculates ln2  
Total=0;  
for i=1:1e+6 %1e+6=1000000  
    Number=(-1)^(i+1)/i;  
    Total = Total + Number;  
end  
fprintf('ln2= %d\n',Total);
```

❖ کنترل ها در برنامه نویسی



برنامه نویسی:

(رابطه ستور)
Taylor Series expansion for e^{-x} around 0 is given as

$$e^{-x} = 1 \ominus x \oplus \frac{x^2}{2} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} \dots$$

(+ - * /)
< رابطه ستور >
< امین یابویم سیده برگد ۴ ال اصلی >
< رهنی خدنی >
TII

❖ کنترل ها در برنامه نویسی

برنامه نویسی:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \quad \left(e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} \right)$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad \rightarrow \quad \text{(جواب عدد)}$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots \quad \text{for } |x| < 1$$

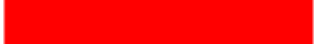







$$\tan^{-1}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots \quad \text{for } |x| < 1$$

عناوین
نشان





```
















1 function y = expon(x)
2   sum=0;
3   fact=1;
4   x_n=1;
5
6   for i = 1:20
7       sum = sum + x_n/fact;
8       fact = fact * i ;
9       x_n = x_n * x;
10
11
12   end
13   y = sum;
14   end
    
```


plot
 linspace
 figure window
 کتابخانه نمادها
 ترسیم چند نمودار
 ترسیم نمودار موتور
 ترسیم منحنی مشخصه های موتور
 ترسیم نمودار جابجایی، سرعت و شتاب پیشتون

Color Name	Short Name	RGB Triplet	Appearance
"red"	"r"	[1 0 0]	
"green"	"g"	[0 1 0]	
"blue"	"b"	[0 0 1]	
"cyan"	"c"	[0 1 1]	
"magenta"	"m"	[1 0 1]	
"yellow"	"y"	[1 1 0]	
"black"	"k"	[0 0 0]	
"white"	"w"	[1 1 1]	

❖ رسم نمودار دو بعدی

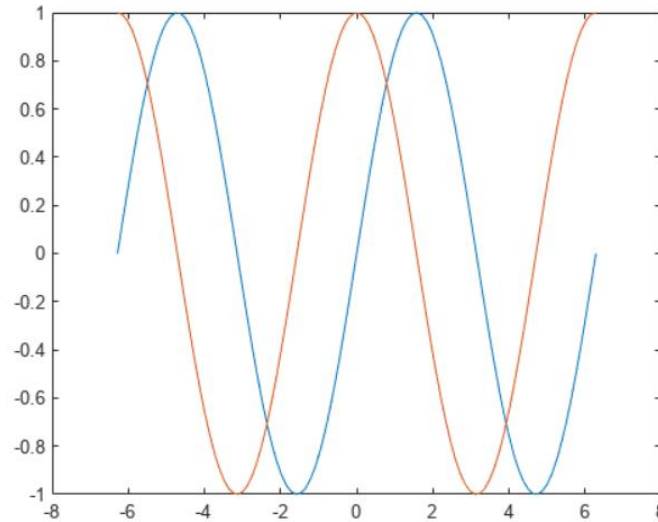
Line Style	Description	Resulting Line
"_"	Solid line	
"--"	Dashed line	
".:"	Dotted line	
"-."	Dash-dotted line	

Marker	Description	Resulting Marker
"o"	Circle	
"+"	Plus sign	
"*"	Asterisk	
"."	Point	
"x"	Cross	
"_"	Horizontal line	
" "	Vertical line	
"square"	Square	
"diamond"	Diamond	
"^"	Upward-pointing triangle	
"v"	Downward-pointing triangle	
">"	Right-pointing triangle	
"<"	Left-pointing triangle	
"pentagram"	Pentagram	
"hexagram"	Hexagram	

❖ رسم نمودار دو بعدی

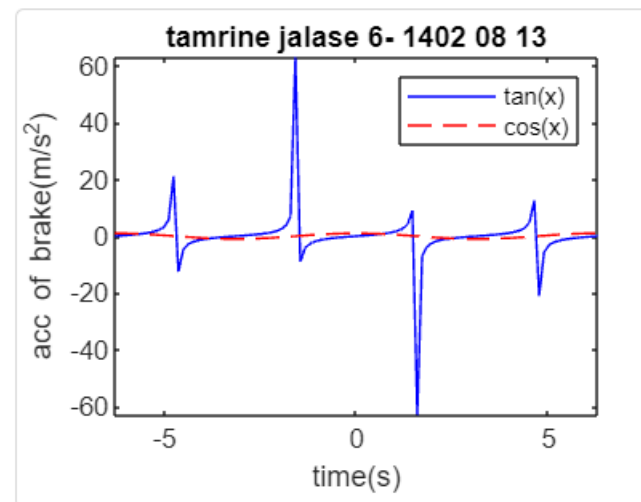
```
x = linspace(-2*pi,2*pi);  
y1 = sin(x);  
y2 = cos(x);
```

```
figure  
plot(x,y1,x,y2)
```



```
x = linspace(-2*pi,2*pi,100);  
y1 = tan(x);  
y2 = cos(x);
```

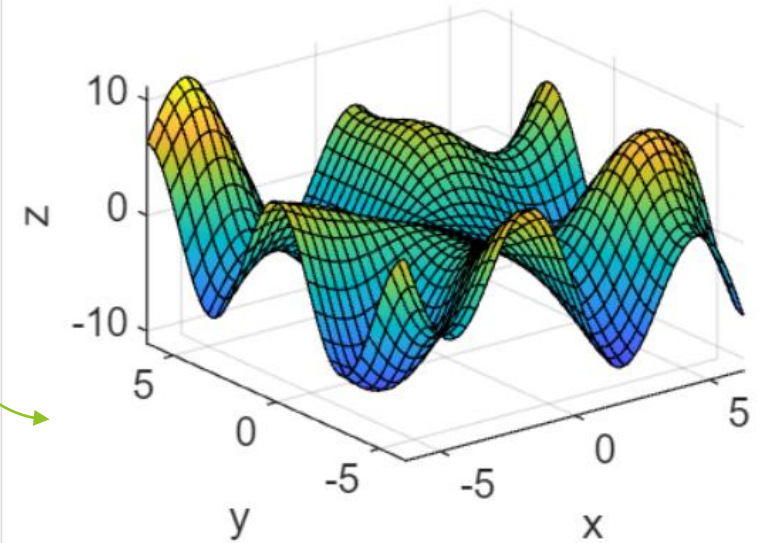
```
figure  
% plot(x,y1,'r')  
plot(x,y1,'-b',x,y2,'--r')  
xlabel('time(s)')  
ylabel('acc of brake(m/s^2)')  
title('tamrine jalase 6- 1402 08 13')  
legend('tan(x)', 'cos(x)')
```



❖ رسم نمودار سه بعدی

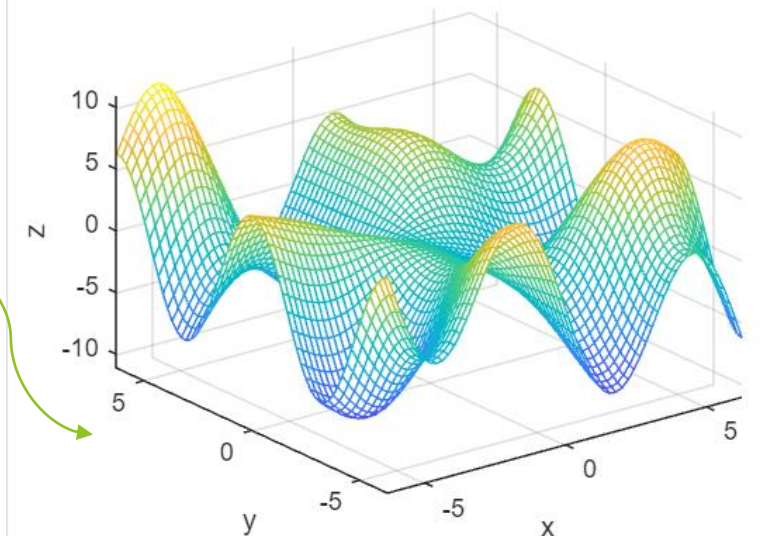
```
ezsurf(@(x,y) y.*sin(x)-x.*cos(y),[-2*pi 2*pi])  
title('ysin(x) - xcos(y) for x and y in [-2\pi,2\pi]')  
xlabel('x');  
ylabel('y');  
zlabel('z');
```

ysin(x) - xcos(y) for x and y in $[-2\pi, 2\pi]$



```
ezmesh(@(x,y) y.*sin(x)-x.*cos(y),[-2*pi 2*pi])  
title('ysin(x) - xcos(y) for x and y in [-2\pi,2\pi]')  
xlabel('x');  
ylabel('y');  
zlabel('z');
```

ysin(x) - xcos(y) for x and y in $[-2\pi, 2\pi]$



Roots of Quartic Polynomial

Solve the equation $x^4 - 1 = 0$.

Create a vector to represent the polynomial, then find the roots.

```
p = [1 0 0 0 -1];  
r = roots(p)
```

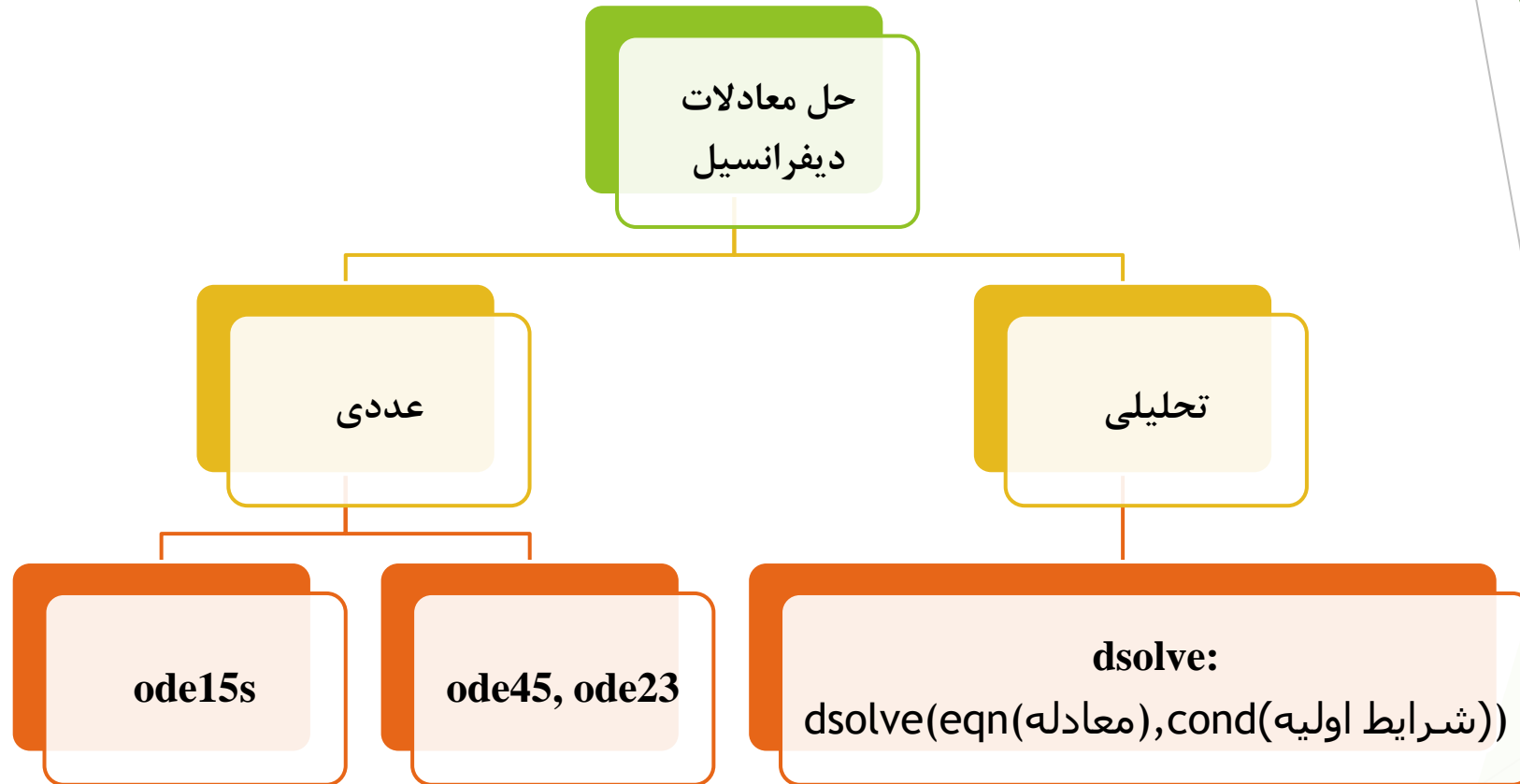
```
r = 4x1 complex  
-1.0000 + 0.0000i  
0.0000 + 1.0000i  
0.0000 - 1.0000i  
1.0000 + 0.0000i
```

- ### فهرست محتوای این جلسه
- ایجاد چند جمله ای
 - ریشه یابی _roots_
 - مشتق _diff_
 - انتگرال _int_integral_
 - معرفی انواع معادله دیفرانسیل

❖ تحلیل معادلات دیفرانسیل

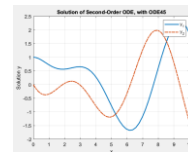
فهرست محتوای این جلسه

- dsolve مرتبه اول و دوم
- ode45
- شرایط مرزی
- حل معادلات خودرویی



❖ تحلیل معادلات دیفرانسیل

دستور	نوع مسئله	دقت	کاربرد
Ode45	غیر سخت nonStiff	متوسط	اکثر موارد (سعی شود جهت حل معادله از ابتدا از این دستور استفاده شود)
Ode23		پایین	حل مسائل دارای خطای خام (crude error)، حل مسائل تقریباً سخت
Ode113		پایین تا بالا	حل مسئله دارای خطای دقیق، معادلات مربوط به محاسبات عددی زمان بر
Ode15s	سخت stiff	پایین تا متوسط	هنگامی که حل معادله با دستور ode45 بسیار کند پیش رود.
Ode23s		پایین	مسائل دارای خطای خام (crude error)، هنگامی که ماتریس جرم (mass matrix)، ثابت باشد
Ode23t		پایین	حل معادلات بدون میرایی عددی (numerical damping)
Ode23tb		پایین	حل مسئله با خطای خام و معادله سخت
Ode15i	کاملاً ضمنی Fully implicit	پایین	برای مسئله های کاملاً ضمنی استفاده میشود و معادلات جبری دیفرانسیل



معادله دیفرانسیل

MATLAB® دستورات در

$$\frac{dy}{dt} + 4y(t) = e^{-t},$$

$$y(0) = 1.$$

```
syms y(t)
ode = diff(y)+4*y == exp(-t);
cond = y(0) == 1;
ySol(t) = dsolve(ode,cond)
```

```
ySol(t) =
exp(-t)/3 + (2*exp(-4*t))/3
```

$$2x^2 \frac{d^2 y}{dx^2} + 3x \frac{dy}{dx} - y = 0.$$

```
syms y(x)
ode = 2*x^2*diff(y,x,2)+3*x*diff(y,x)-y == 0;
ySol(x) = dsolve(ode)
```

```
ySol(x) =
C2/(3*x) + C3*x^(1/2)
```

معادله هوایی Airy

$$\frac{d^2 y}{dx^2} = xy(x).$$

```
syms y(x)
ode = diff(y,x,2) == x*y;
ySol(x) = dsolve(ode)
```

```
ySol(x) =
C1*airy(0,x) + C2*airy(2,x)
```

❖ سیمولینک و بلوکهای آن

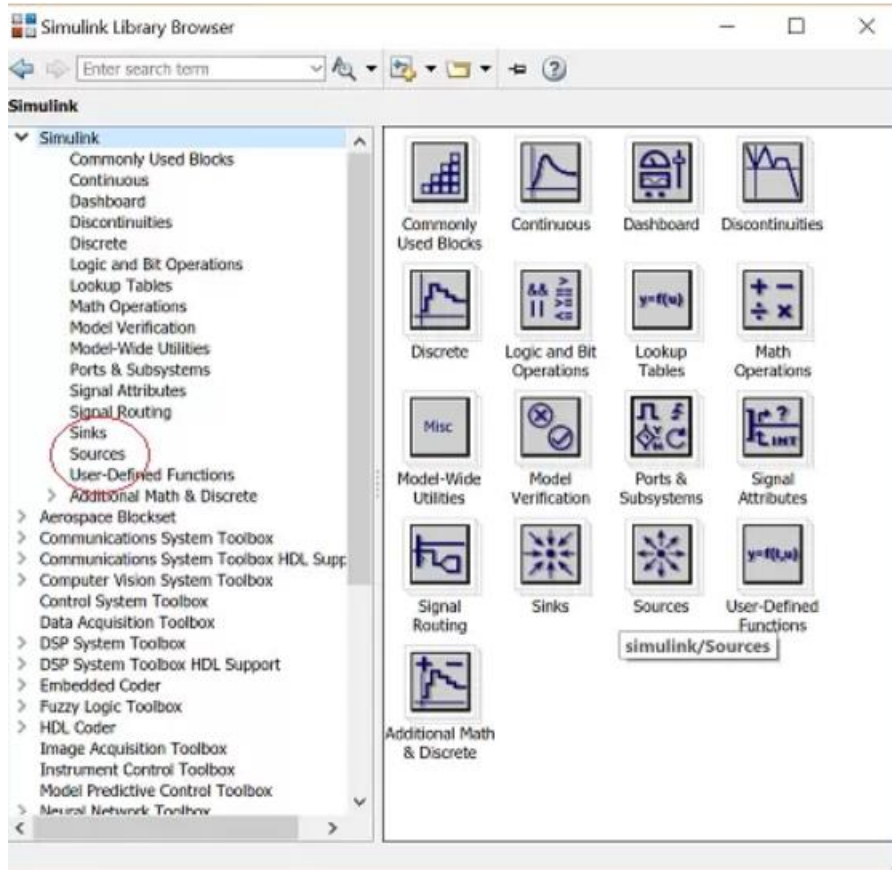
بلوک ها



چندین گروه کلی از بلوک ها در یک کتابخانه ی سیمولینک وجود دارد:

- **منابع:** برای ایجاد سیگنال های مختلف استفاده می شوند.
- **سینک ها:** برای نمایش یا خروجی سیگنال ها به کار می روند.
- **مستمر:** عناصر سیستم مستمر زمانی (توابع انتقال ، مدل های فضایی ، کنترل کننده ها ی PID)
- **مجزا:** عناصر سیستم مجزای زمانی، خطی
- **عملیات ریاضی:** شامل بسیاری عناصر معمول ریاضی است.
- **خروجی ها و سیستم های فرعی:** شامل بلوک های مفید برای ساخت سیستم است.

بلوک های دارای چندین خروجی صفر تا چند تایی است و همچنین خروجی های چند گانه پایانه های داده های ورودی که استفاده نشده اند توسط مثلث باز کوچک نشان داده می شود. پایانه های استفاده نشده توسط نقاط مثلثی نمایش داده می شوند . بلوک نشان داده شده دارای یک پایانه داده های ورودی سمت چپ و یک پایانه داده های استفاده نشده خروجی در سمت راست است.



❖ سیمولینک و بلوکهای آن

Simulink >> Sources >> Clock Simulink

>> Sources >> Constant

Simulink >> Math Operations >> Sum

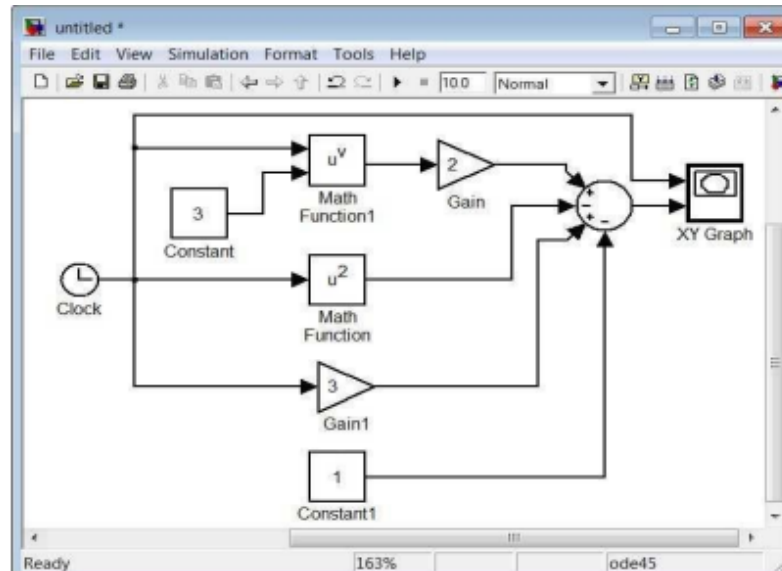
Simulink >> Math Operations >> Gain

Simulink >> Math Operations >> Math Function

Simulink >> Sinks >> XY Graph

مثال اول را می‌خواهیم با رسم نمودار یک تابع آغاز کنیم ضابطه این تابع به صورت زیر است:

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



❖ سیمولینک و بلوکهای آن

Simulink >> Sources >> Constant Simulink

>> Sinks >> Display

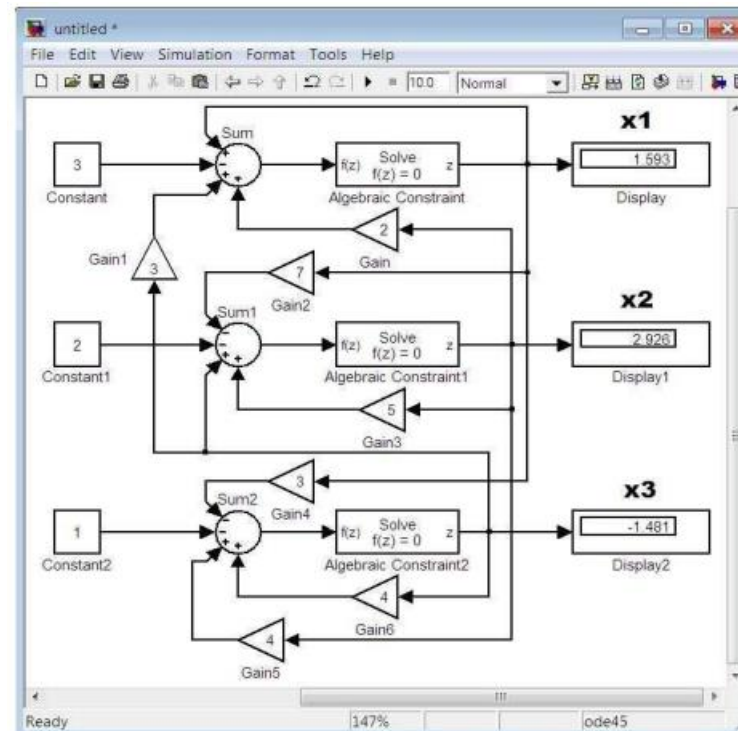
Simulink >> Math Operations >> Sum

Simulink >> Math Operations >> Gain

Simulink >> Math Operations >> Algebraic Constraint

برای مثال آخر در قسمت دوم آموزش میخواهیم دستگاه معادلات زیر را حل کنیم:

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 3 \\ -7x_1 + 5x_2 + x_3 = 2 \\ -3x_1 + 4x_2 + 4x_3 = 1 \end{cases}$$

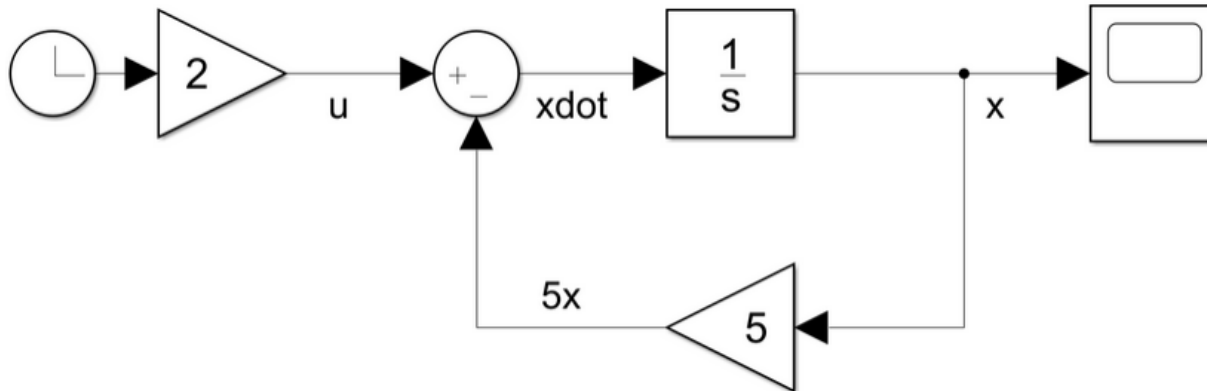


❖ سیمولینک و بلوکهای آن

می خواهیم معادله زیر را حل کنیم.

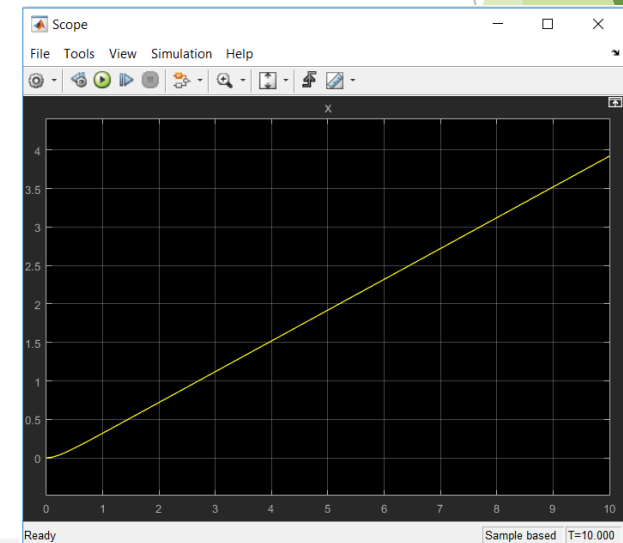
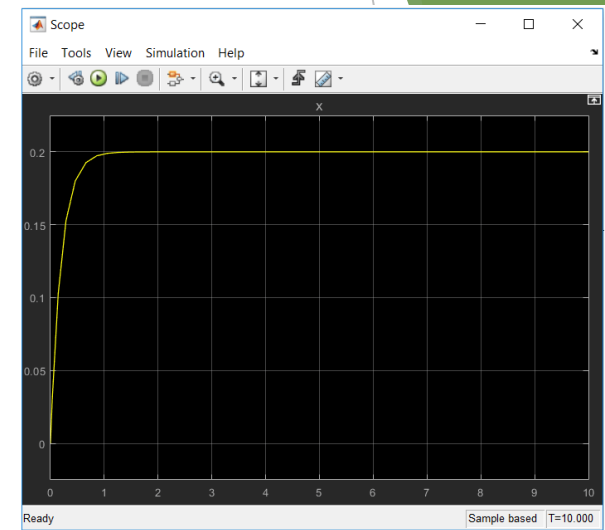
$$\dot{x} + 5x = u$$

در این حالت فرض می کنیم مقدار u برابر $2t$ باشد و با زمان تغییر کند. در این حالت کافیسیت که فقط به جای بلوک constant از بلوک clock که به ما مقدار t را در هر لحظه به ما می دهد استفاده کنیم. بعد از بلوک clock به یک بلوک gain هم نیاز داریم تا ضریب ۲ را برای $u=2t$ ایجاد کنیم.



حالا پروژه آماده است. پس از run گرفتن، با دو بار کلیک روی بلوک scope می توانیم نمودار زیر را مشاهده کنیم.

مرحله: ویکی مکانیک



❖ سیمولینک و بلوکهای آن

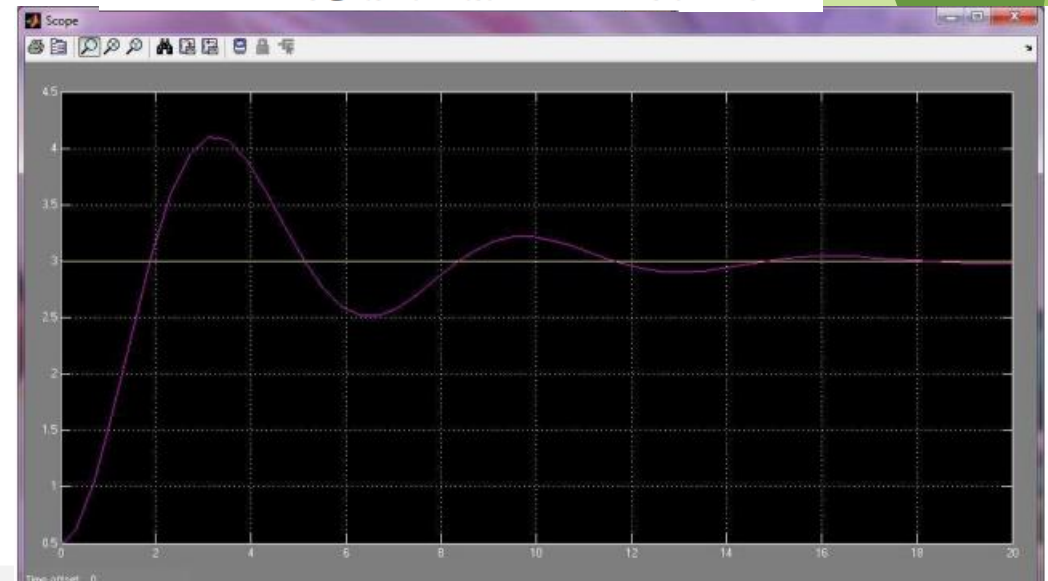
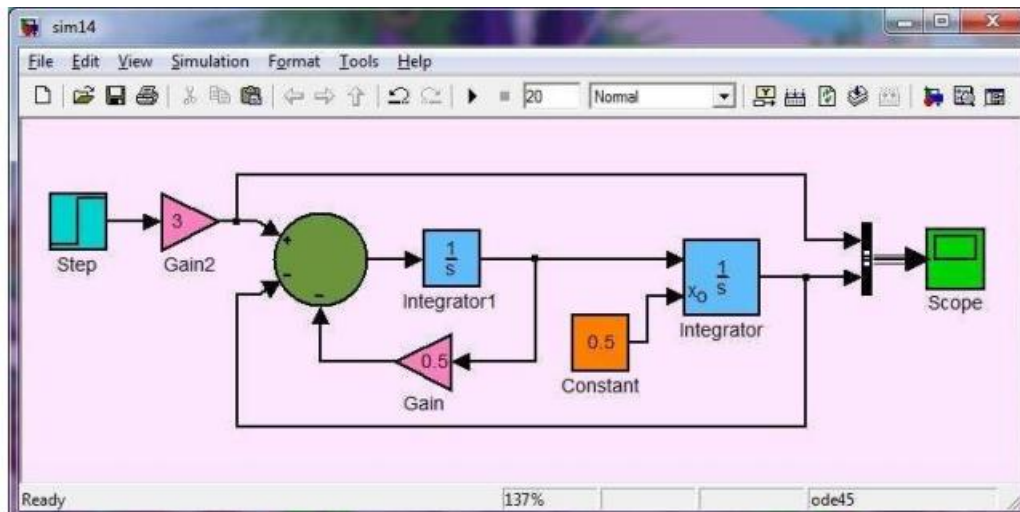
$$\frac{d^2y}{dt^2} + 0.5 \frac{dy}{dt} + y(t) = 3u(t)$$
$$y'(0) = 0, \quad y(0) = 0.5$$

Simulink >> Sources >> Step Simulink
>> Sources >> Constant
Simulink >> Math Operations >> Sum
Simulink >> Math Operations >> Gain
Simulink >> Continuous >> Integrator
Simulink >> Signal Routing >> Bus Creator Simulink
>> Sinks >> Scope

- روی بلوک Step دوبار کلیک نموده و مقدار Step time را ۰ قرار دهید.
- روی بلوک Constant دوبار کلیک نموده و مقدار آن را ۰.۵ قرار دهید.
- روی بلوک Gain دوبار کلیک نموده و مقدار آن را ۰.۵ قرار دهید.
- روی بلوک Gain2 دوبار کلیک نموده و مقدار آن را ۳ قرار دهید.
- روی بلوک Sum دوبار کلیک نموده و در قسمت List of signs علامت های + و - را وارد کنید. • روی بلوک Integrator دوبار کلیک نموده و در قسمت Initial condition source حالت external را برگزینید.

- مقدار Stop time را ۰.۲ ثانیه قرار دهید.

حال مدل را Run کنید تا ورودی و خروجی را مشاهده کنید:



پروژه نهایی و امتحان پایانی	
برنامه ای بنویسید که دارای یک منو به صورت گرافیکی به شرح زیر باشد:	
مقایسه حداقل 10 خودرو برتر خارجی بر اساس 4 معیار استاندارد نظیر قیمت، ایمنی و سایر امکانات خودرو با قابلیت سورت و طبقه بندی	قسمت اول
بررسی مشخصات حداقل 3 خودرو داخلی شامل منحنی مشخصه، مشخصات نمایشگر پشت فرمان و مشخصات خاص آن ها	
حل یک معادله دیفرانسیل خودرو به صورت دلخواه و رسم پاسخ آن بر اساس شرایط اولیه قابل تنظیم	قسمت دوم
ارائه نهایی می تواند به زبان برنامه نویسی به غیر از زبان متلب باشد...	
بایستی هر قسمت یک توضیح علمی از نحوه عملکرد داشته باشد.	



Full screen image with caption lorem ipsum dolor sit amet



Thanks

Mh Aghananouri



@matlab_1402



Mha.matlab1402@gmail.com



صلوات

