Workshop – MATLAB & SIMULINK

Teoria de Controlo

Licenciatura em Engenharia Física

Segunda, 12 de Fevereiro, 2024

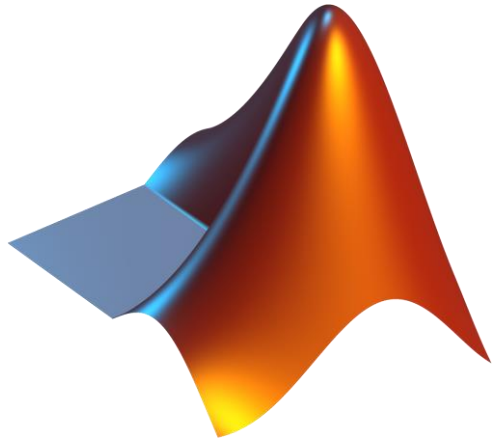
Vinícius Silva | Automação Controlo e Robótica | D7662@dei.uminho.pt

Conteúdo da Apresentação

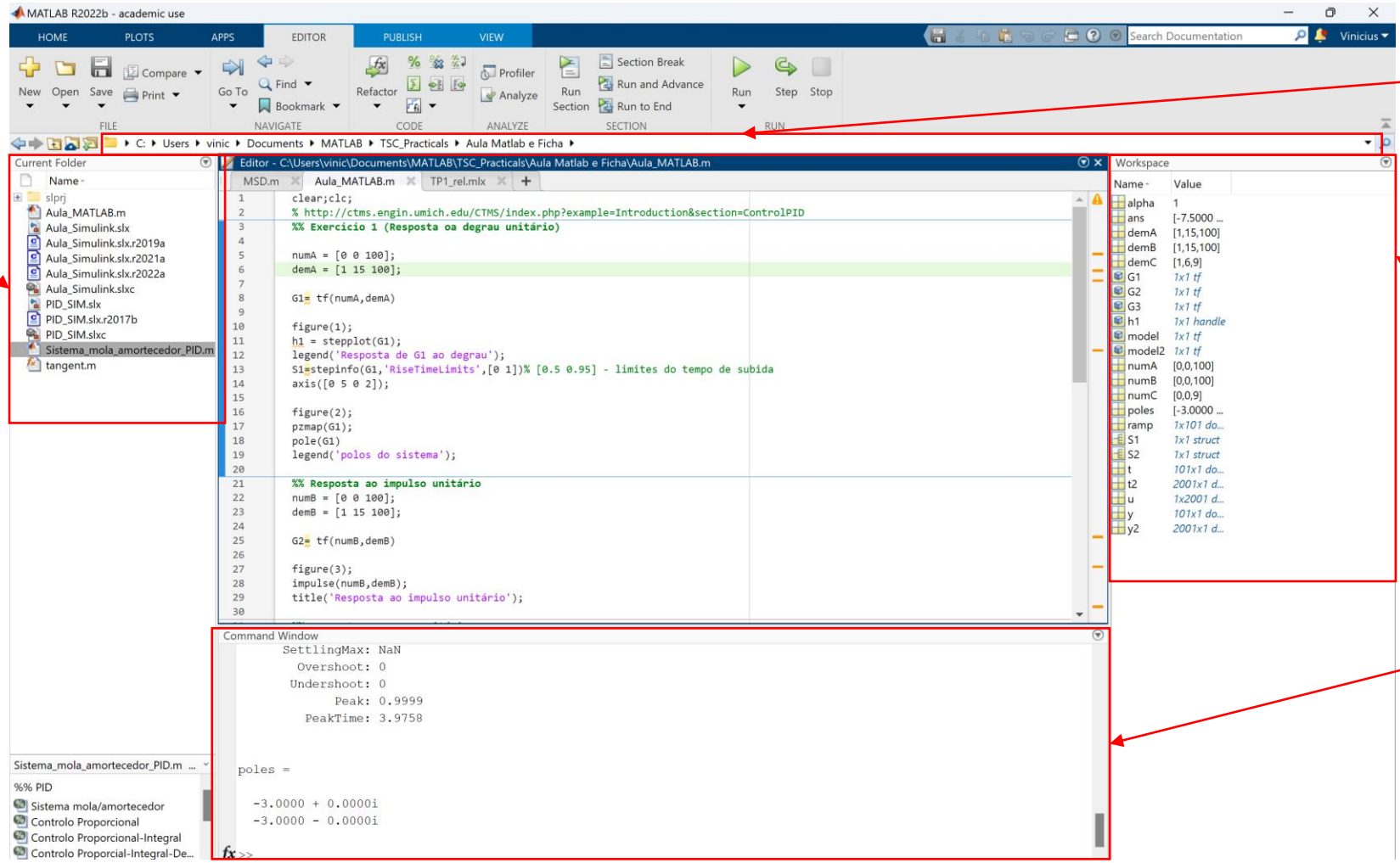
1. Overview
    1. Variables
    2. Matrix Index
    3. Matrix Operations
    4. Flow Control
2. Simulink
3. Matlab + Arduino – TCLab

MATLAB

- Matrix and Laboratory:
  - Dynamically typed language
    - Variables require no declaration
    - Creation by initialization (x=10;)
  - All variables are treated as matrices
    - Scalar: 1×1 matrix; Vector: N×1 or 1×N matrix
    - Calculations are much faster
  - Advantages
    - Fast implementation and debugging
    - Natural matrix operation
    - Powerful image processing toolbox



MATLAB Main Window



Current Folder Contents

Current Folder Adress:
- View folders/files

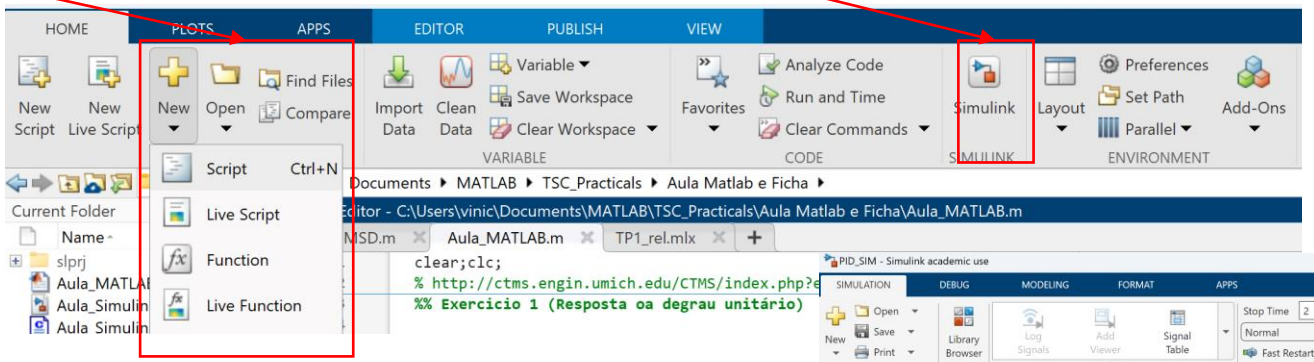
Workshop:
- View variables
- Inspect variables contents

Command Prompt/window:
- View past commands
- Execute small commands

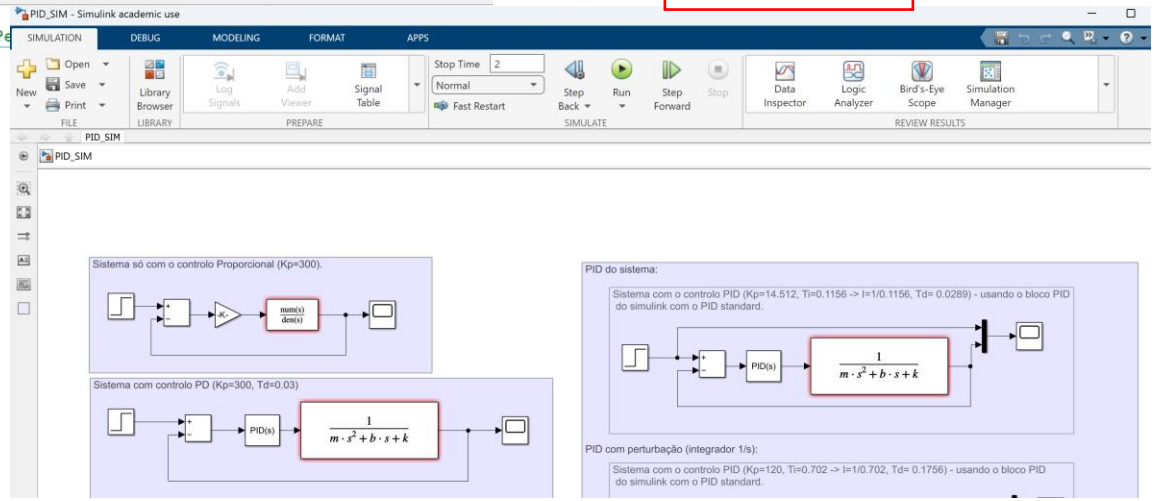
MATLAB Main Window

Create new MATLAB files:
- Script
- LiveScript
- Function

Simulink:
- Initiate Simulink



Simulink file .slx



Script file .m

```
clear;clc;
% http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID
%% Exercicio 1 (Resposta oa degrau unitário)

numA = [0 0 100];
demA = [1 15 100];

G1= tf(numA,demA)

figure(1);
h1 = stepplot(G1);
legend('Resposta de G1 ao degrau');
S1=stepinfo(G1,'RiseTimeLimits',[0 1])% [0.5 0.95] - limites do tempo de subida
axis([0 5 0 2]);

figure(2);
pzmap(G1);
pole(G1)
legend('polos do sistema');
```

Variables

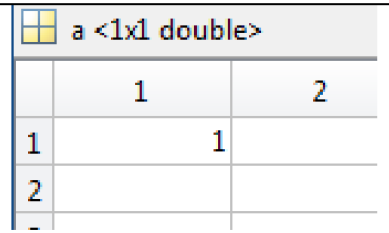
- Defining variables – variables are created when they are used:

```
int a;
a=1;
double b;
b=2+4;
```
C/C++

```
>>a=1;
>>b=2+4;
```
Matlab

```
>> a=1
a =
     1
>> b=2+4
b =
     6
```

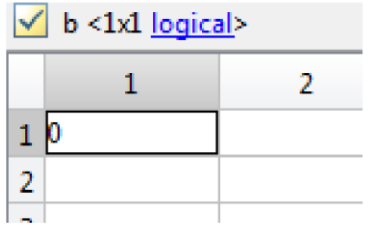
a = 1;




```
>> whos a
  Name        Size                Bytes  Class     Attributes

  a           1x1                     8  double
```

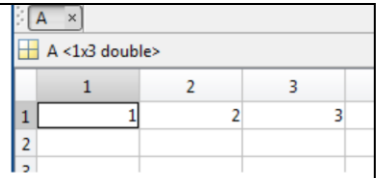
b = false;



Variables

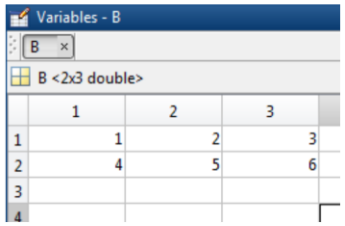
A = [1, 2, 3]

```
>> A=[1 2 3]
A =
     1     2     3
```



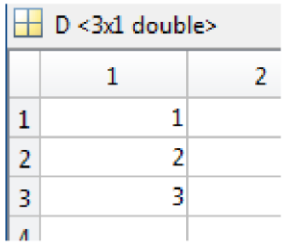
B = [1,2,3;4,5,6]

```
>> B=[1,2,3;4,5,6]
B =
     1     2     3
     4     5     6
```



D=[1 ; 2 ; 3]

```
>> D= [1 ;2 ;3]
D =
     1
     2
     3
```



E=[1 2 3]'

```
>> E=[1 2 3]'
E =
     1
     2
     3
```

Variables

```
>> A=1:10
A =
     1     2     3     4     5     6     7     8     9    10


>> B= 0:2:10
B =
     0     2     4     6     8    10


>> 1:0.5:5
ans =
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
```

Extra commands:

- A = zeros(3);
- B = ones(5);
- C = rand(100,2);
- D = eye(20);

Matrix Index

- Matrix indices begin from 1
- Matrix indices must be positive integers

```
A =
     1     2     3
     4     5     6
     7     8     9
```

A(row, column)

```
>> A(1,3)
ans =
     3
```

```
>> A(1)
ans =
     1
```

```
>> A(2)
ans =
     4
```

Column-Major Order

E.g: A(3, 2) = ?

Matrix Index

```
A =
     1     2     3
     4     5     6
     7     8     9
```

```
>> A(2,2:3)

ans =

     5     6
```

```
>> A(2,1:end)

ans =

     4     5     6
```

```
>> A(2,:)

ans =

     4     5     6
```

```
>> A(2,[1 3])

ans =

     4     6
```

https://www.mathworks.com/company/technical-articles/matrix-indexing-in-matlab.html

Matrix Operations

- + addition
- - subtraction
- * multiplication

- ^ power
- ‘ complex conjugate transpose

Matrix Operations

Given A and B:

```
>> A = [1 2 3;4 5 6;7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9
```

```
>> B = [3 5 2; 5 2 8; 3 6 9]

B =

     3     5     2
     5     2     8
     3     6     9
```

Addition

```
>> X = A + B

X =

     4     7     5
     9     7    14
    10    14    18
```

Subtraction

```
>> Y = A - B

Y =

    -2    -3     1
    -1     3    -2
     4     2     0
```

Product

```
>> Z = A * B

Z =

    22    27    45
    55    66   102
    88   105   159
```

Transpose

```
>> T = A'

T =

     1     4     7
     2     5     8
     3     6     9
```

Matrix Operations

- .* element-wise multiplication
- ./ element-wise division
- .^ element-wise power

```
A = [1 2 3; 5 1 4; 3 2 1]
   A =
        1    2    3
        5    1    4
        3    2   -1
```

```
x = A(1,:)

x=
     1   2   3
```

```
y = A(3 ,:)

y=
     3  4  -1
```

```
b = x .* y

b=
     3  8 -3
```

```
c = x . / y

c=
   0.33   0.5   -3
```

```
d = x .^y

d=
        1    16    0.33
```

Flow control

- If, for, while…

```
if (a<3)
    Some Matlab Commands;
elseif (b~=5)
    Some Matlab Commands;
end
```

```
while  ((a>3) & (b==5))
    Some Matlab Commands;
end
```

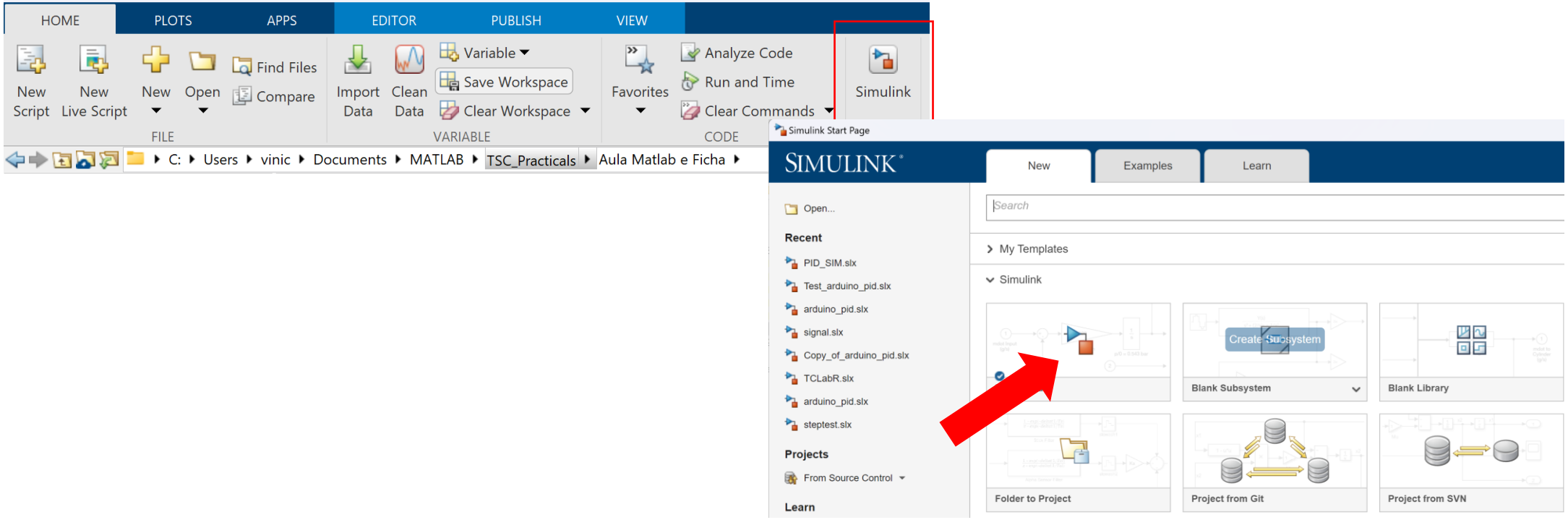
```
for ii=1:100
    Some Matlab Commands;
end
```

```
for j=1:3:200
    Some Matlab Commands;
end
```

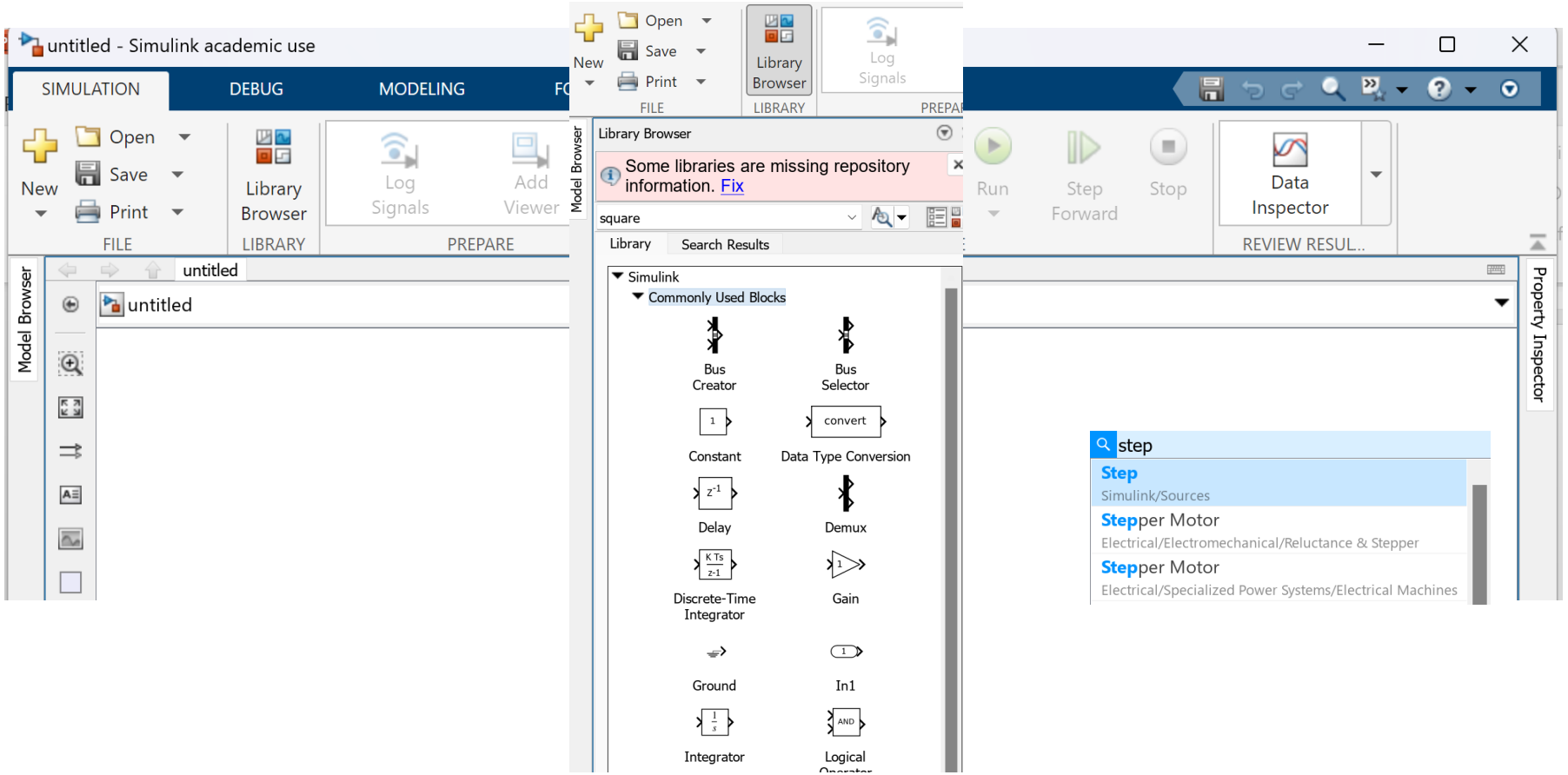
```
for k=[0.1 0.3 -13 12 7 -9.3]
    Some Matlab Commands;
end
```

Simulink

- Create a new model:

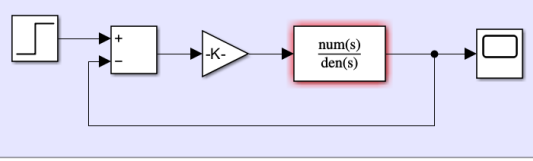


Simulink

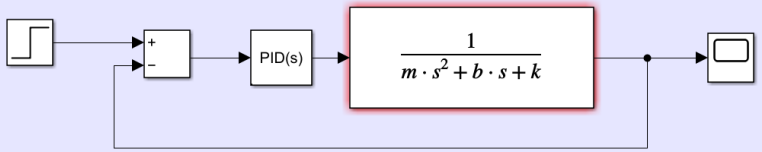

untitled - Simulink academic use
SIMULATION
DEBUG
MODELING
New
Open
Save
Print
FILE
Library Browser
LIBRARY
Log Signals
Add Viewer
PREPARE
Run
Step Forward
Stop
Data Inspector
REVIEW RESUL...
Model Browser
untitled
Property Inspector
Library Browser
Some libraries are missing repository information. Fix
square
Library
Search Results
Simulink
Commonly Used Blocks
Bus Creator
Bus Selector
Constant
Data Type Conversion
Delay
Demux
Discrete-Time Integrator
Gain
Ground
In1
Integrator
Logical
step
Step
Simulink/Sources
Stepper Motor
Electrical/Electromechanical/Reluctance & Stepper
Stepper Motor
Electrical/Specialized Power Systems/Electrical Machines

Simulink

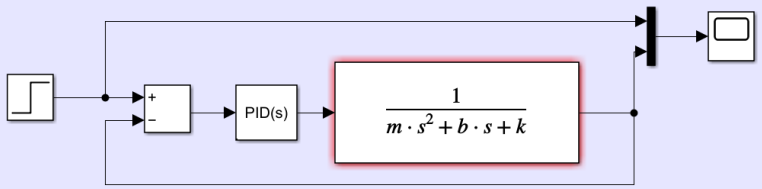
Sistema só com o controlo Proporcional (Kp=300).



Sistema com controlo PD (Kp=300, Td=0.03)

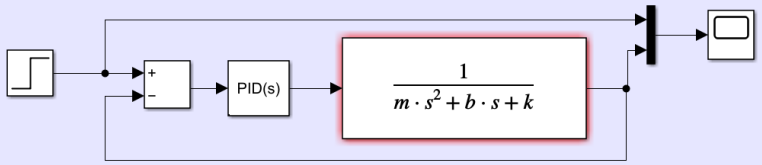


Sistema com o controlo PI (Kp=30, Ti=0.43 -> I=1/0.43) - usando o bloco PID do simulink com o PID standard.



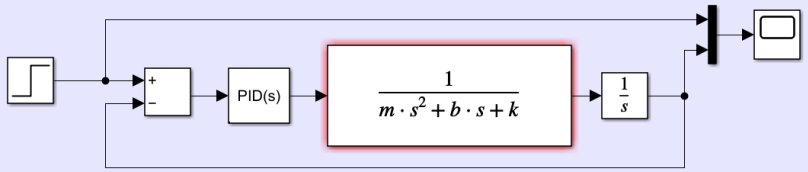
PID do sistema:

Sistema com o controlo PID (Kp=14.512, Ti=0.1156 -> I=1/0.1156, Td= 0.0289) - usando o bloco PID do simulink com o PID standard.



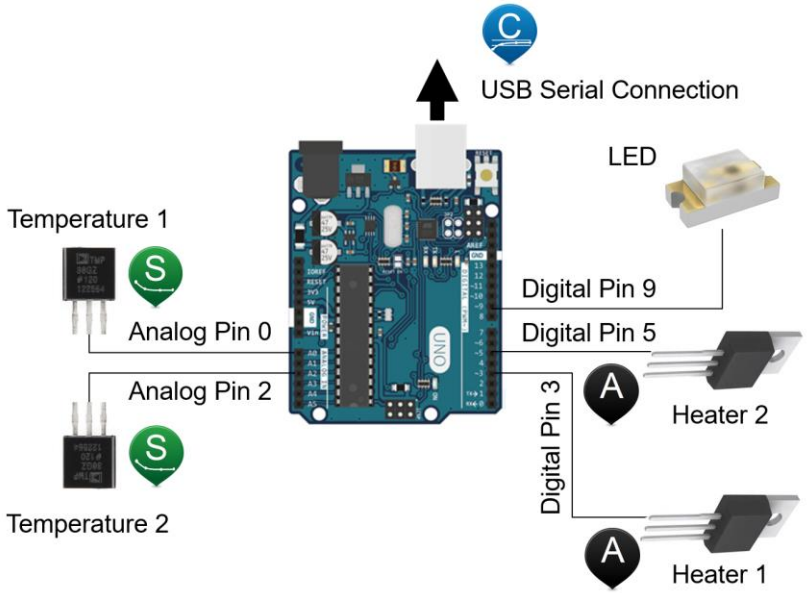
PID com perturbação (integrador 1/s):

Sistema com o controlo PID (Kp=120, Ti=0.702 -> I=1/0.702, Td= 0.1756) - usando o bloco PID do simulink com o PID standard.



Matlab + Arduino – TCLab

- [Temperature Control Lab]



Extra Resources:

- Matlab “help” function:

```
>> help step
 step  Step response of dynamic systems.

    [Y,T] = step(SYS) computes the step response Y of the dynamic system SYS.
    The time vector T is expressed in the time units of SYS and the time
    step and final time are chosen automatically. For multi-input systems,
    independent step commands are applied to each input channel. If SYS has
    NY outputs and NU inputs, Y is an array of size [LENGTH(T) NY NU] where
    Y(:,:,j) contains the step response of the j-th input channel.
```

- https://ctms.engin.umich.edu/CTMS/index.php?aux=Home
- Matlab website