PMR 3302
Sistemas Dinâmicos I

AULA 03: EQUAÇÕES DIFERENCIAIS

Larissa Driemeier
driemeie@usp.br

- Você precisa do Octave ou do MatLab para fazer essa aula (https://www.gnu.org/software/octave/download.html)
- Faça os exercícios, e olhe no arquivo disponibilizado no stoa toda vez que você não concordar ou não entender o gabarito mostrado no slide.
- Pause a aula e rode os programas sugeridos.

Listas de exercícios, mudança da programação, comunicados gerais serão por meio do *stoa*. Por favor, verifique regularmente o site.

NOSSA AGENDA

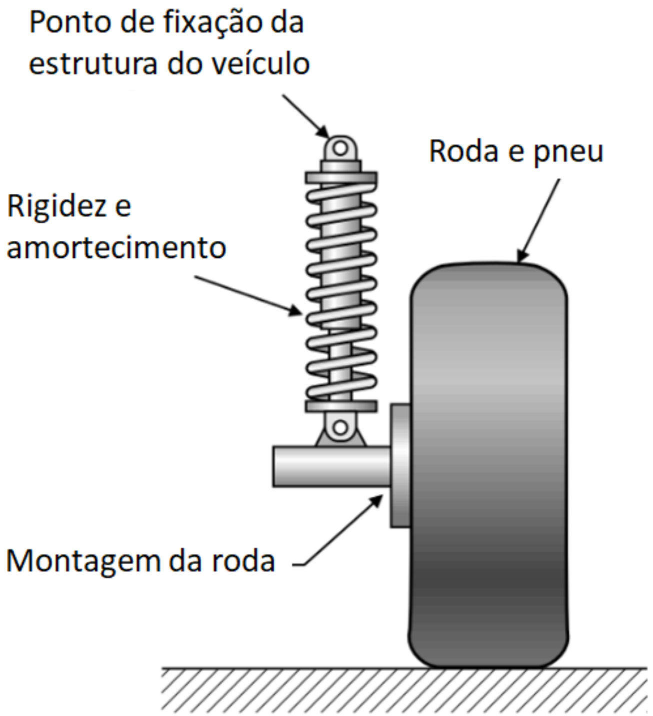
#	Data	Tópico
1	21/02	Introdução ao modelamento e uso do software
2	06/03	Introdução à programação em MatLab
3	20/03	Resolução de Equações Diferenciais - Sistemas Lineares e Não Lineares
4	03/04	Transformada de Laplace e Funções de Transferência
5	24/04	Projeto
6	15/05	Diagrama de Blocos e Simulink
7	29/05	Análise de Sistemas de Primeira Ordem
8	19/06	Análise de Sistemas de Segunda Ordem

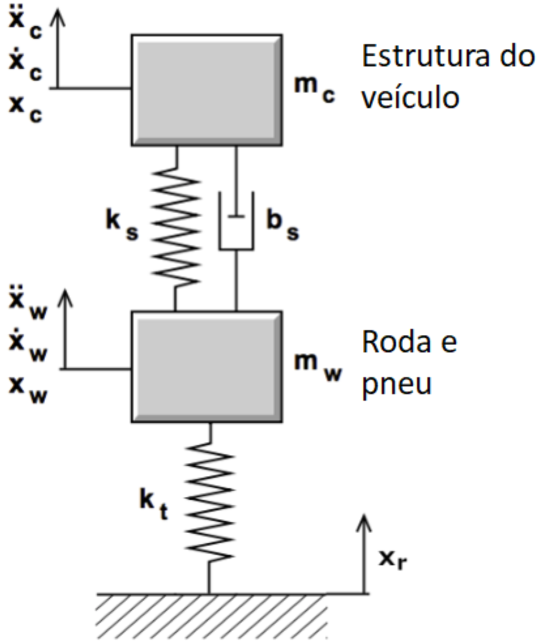
SISTEMAS

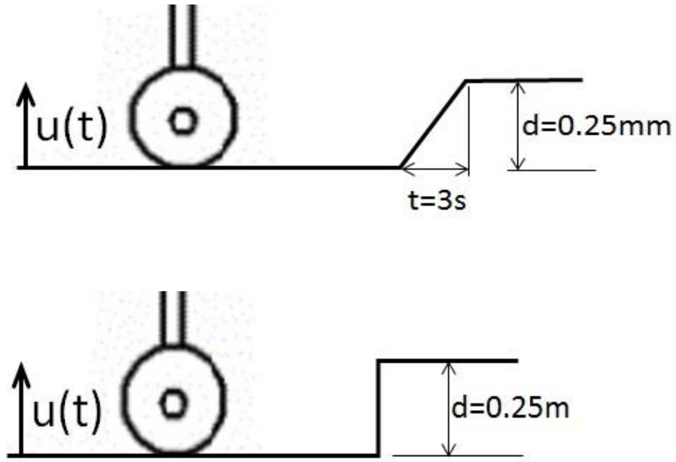
SLIT

SLIT
SISTEMA LINEAR INVARIANTE NO TEMPO

- **O problema modelado, deve ser agora analisado.**







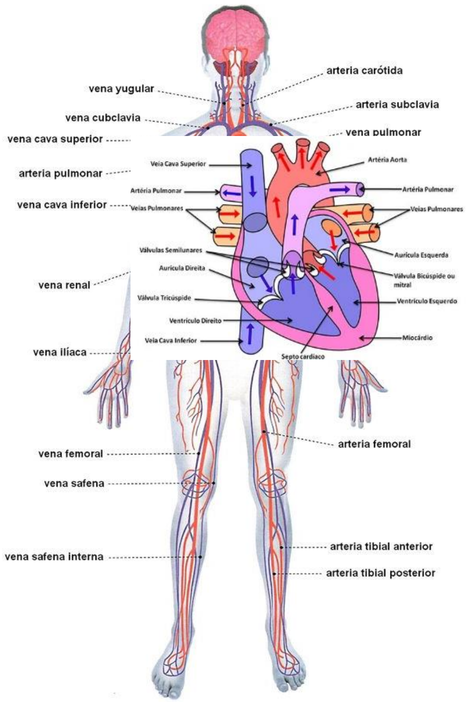
O QUE É UM SISTEMA?

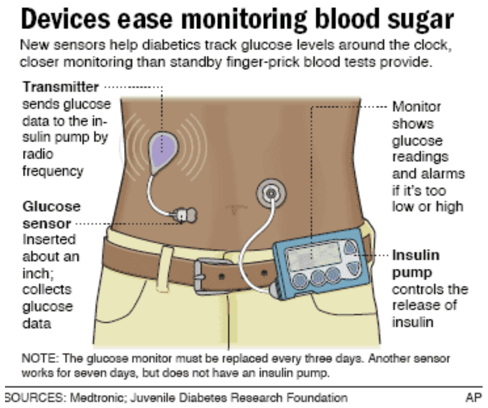
Sistema: Conjunto de componentes interconectados, que apresentam certas relações de causa e efeito e que atuam como um todo, com um determinado objetivo.

Sistema estático

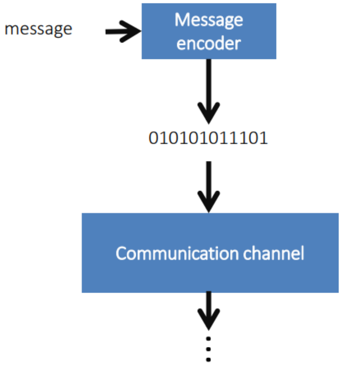
Sistema dinâmico

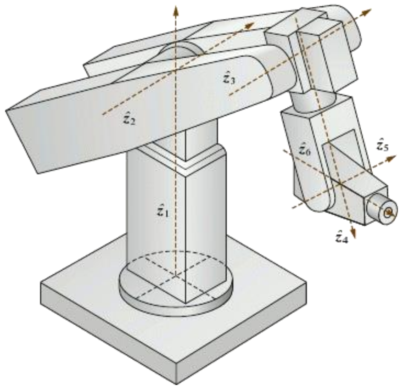
SISTEMA DINÂMICO





Um Sistema dinâmico é um conjunto de elementos conectados entre si que trocam, transformam e dissipam energia.

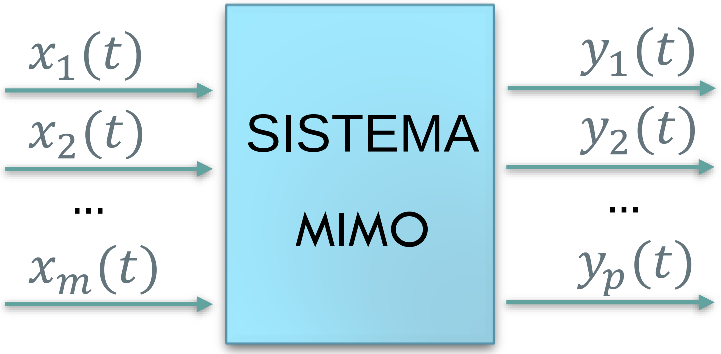




SINAIS E SISTEMAS

Os sinais representam:

- a ***entrada*** de um sistema (*input*): às vezes também é chamado de *controle* ou mesmo *a excitação* do sistema;
- ***saída*** do sistema (*output*): às vezes também é chamado de *resposta* ou *observação* do sistema.



SISTEMA CAUSAL E NÃO CAUSAL

Sistema causal: o/p do sistema não dependente de valores futuros do i/p.
Ex.:

i. $y(t) = x(t)$

ii. $y(t) = x(t) + x(t-1)$

Sistemas fisicamente factíveis são causais!!!!

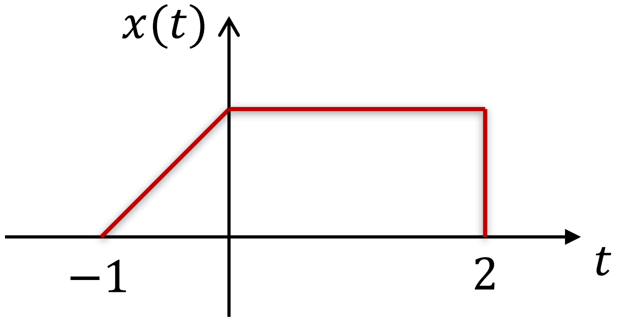
Sistema não causal: o/p do sistema dependente de valores futuros do i/p a qualquer instante de tempo.
Ex.:

Anti-causal: o/p depende APENAS valores futuros de i/p

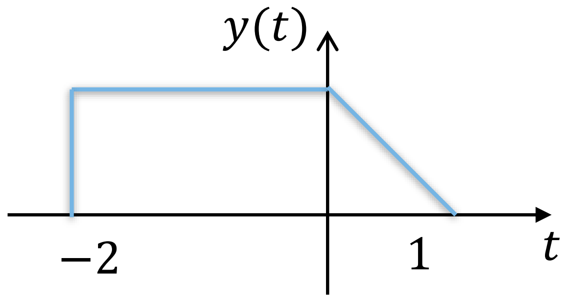
i. $y(t) = x(t+2)$

ii. $y(t) = x(t) + x(t-1) + x(t+1)$

EXEMPLO



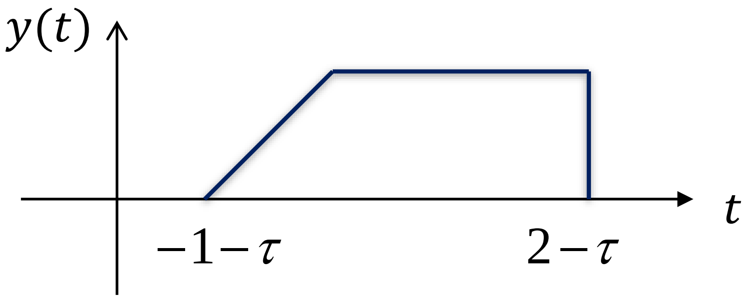
$$y(t) = x(-t)$$



$t \leq -1, x(t) = 0$, mas $y(t) \neq 0$

Sistema Não Causal

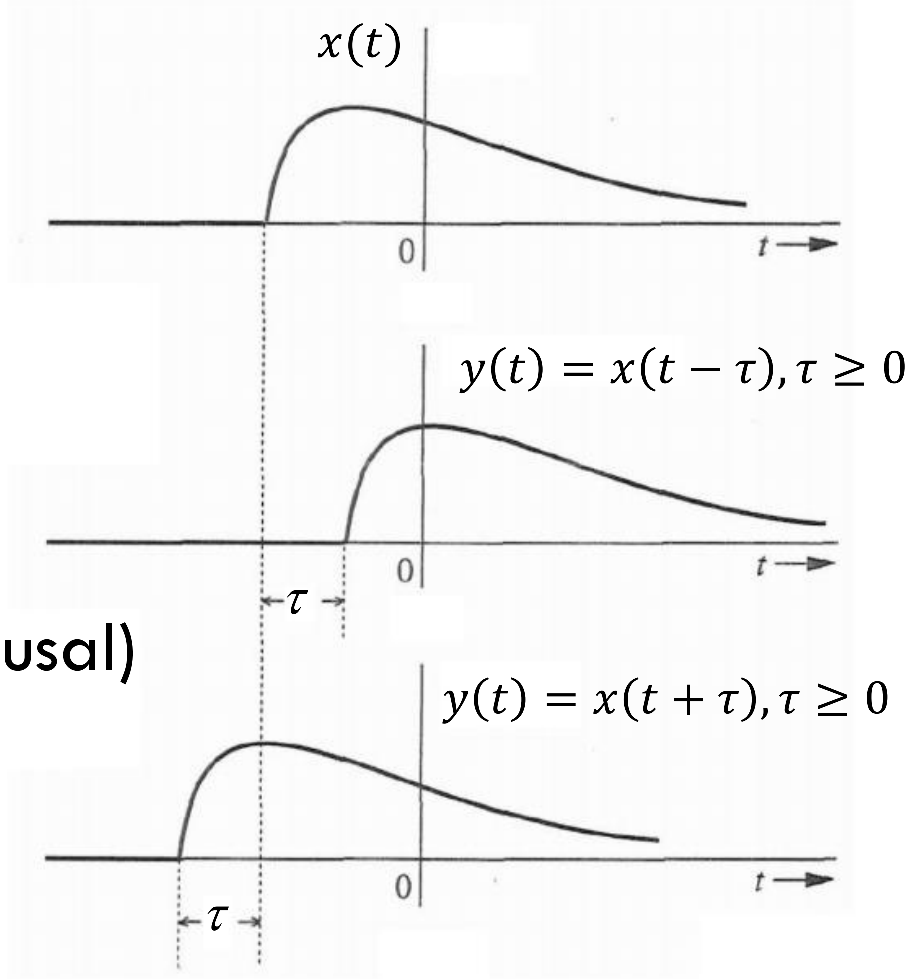
$$y(t) = x(t - \tau), \tau \geq 0$$



Sistema com delay...

Sistema Causal

SINAL COM ATRASO OU AVANÇO

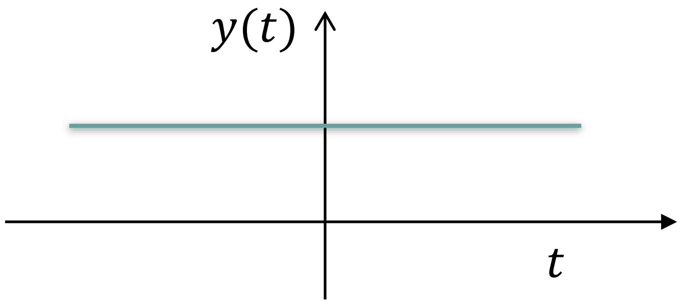


Atraso (causal)

Avanço (não causal)

MAIS EXEMPLOS...

$y(t) = x(t^2)$, onde $x(t) = u(t)$.



$t \leq 0, x(t) = 0$, mas $y(t) \neq 0$

Sistema Não Causal

$y(t) = \boxed{\sin(t+1)}\, x(t-1)$

Coeficiente...

Sistema Causal

EXERCÍCIOS

Verifique se cada sistema abaixo é causal (C) ou não causal (NC).

i. $y(t) = x(2t)$

ii. $y(t) = \begin{cases} x(2t) & t < 0 \\ x(t-1) & t \geq 0 \end{cases}$

iii. $y[n] = \left(\frac{1}{2}\right)^{n+1} x[n-1]$

iv. $y(t) = \sin t \, x(t)$

v. $y(t) = x(e^t)$

vi. $y(t) = x(\sin t)$

vii. $y(t) = x(t/4)$

viii. $y(t) = e^t x(t-1)$

ix. $y(t) = \int_{-\infty}^{t} x(t)\, dt$

SLIT - SISTEMAS LINEARES E INVARIANTES NO TEMPO

LTI SYSTEMS (LINEAR AND TIME INVARIANT SYSTEMS)

sistema *linear*

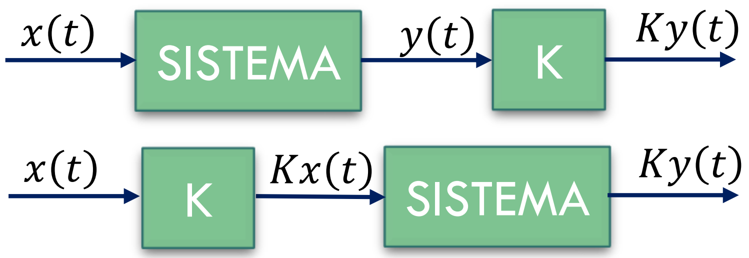
Sistemas contínuos são lineares se satisfazem duas propriedades: ***homogeneidade e aditividade*** (superposição)

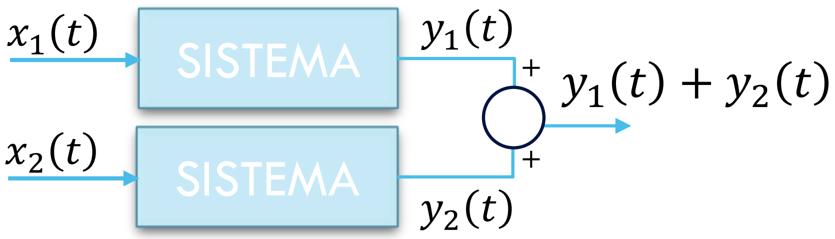
$$x(t) \rightarrow y(t) \ \rightarrow Kx(t) \rightarrow Ky(t)$$

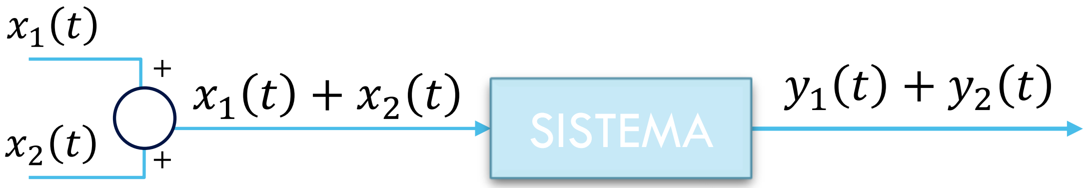
$$\begin{matrix} x_1(t) \rightarrow y_1(t), \\ x_2(t) \rightarrow y_2(t) \end{matrix} \quad \rightarrow x_1(t) + x_2(t) \rightarrow y_1(t) + y_2(t)$$

Isto é,

$$K_1x_1(t) + K_2x_2(t) \rightarrow K_1y_1(t) + K_2y_2(t)$$


$x(t)$
SISTEMA
$y(t)$
K
$Ky(t)$
$x(t)$
K
$Kx(t)$
SISTEMA
$Ky(t)$


$x_1(t)$
SISTEMA
$y_1(t)$
+
$y_1(t) + y_2(t)$
$x_2(t)$
SISTEMA
+
$y_2(t)$


$x_1(t)$
+
$x_1(t) + x_2(t)$
SISTEMA
$y_1(t) + y_2(t)$
$x_2(t)$
+


$K_1x_1(t) + K_2x_2(t)$
SISTEMA
$K_1y_1(t) + K_2y_2(t)$

EXEMPLOS...

i. $y(t) = x(\sin t)$

ii. $y(t) = x(t^2)$

iii. $y(t) = x(\log t)$

iv. $y(t) = [x(t)]^2$

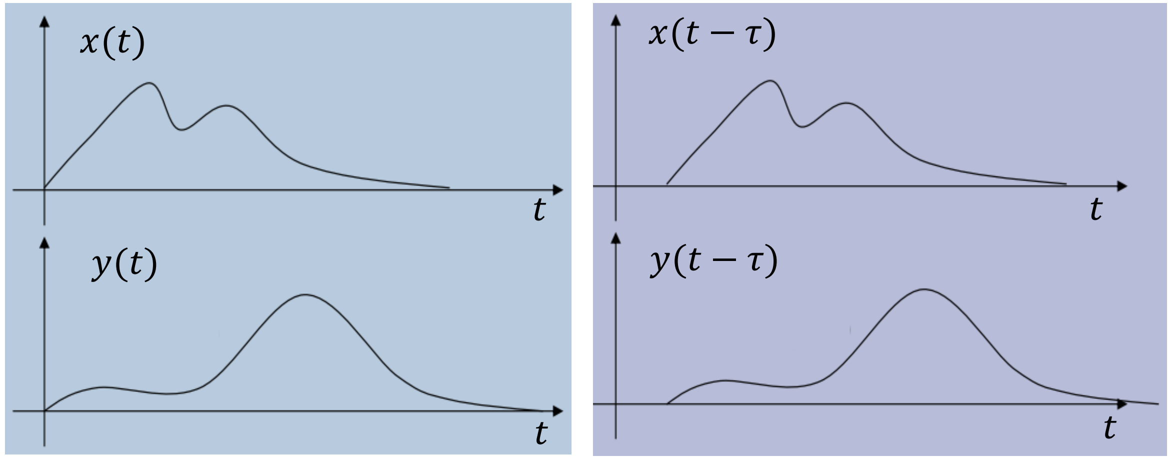
v. $y(t) = \sin t \, x(t)$

vi. $y(t) = e^2 \, x(t)$

SLIT - SISTEMA LINEAR E INVARIANTE NO TEMPO

LTI SYSTEMS (LINEAR AND TIME INVARIANT SYSTEMS)

sistema *invariante no tempo*



Um sistema é dito invariante no tempo se um deslocamento no tempo do sinal de entrada (retardo ou avanço) implicar em um deslocamento temporal idêntico no sinal de saída:

$$x(t) \rightarrow y(t) \Rightarrow x(t-\tau) \rightarrow y(t-\tau)$$

Sistemas invariante e variante no tempo, respectivamente:

$$\frac{d^2y}{dt^2} + 4\frac{dy}{dt} - y = \frac{dx}{dt} + 3x$$

$$\frac{d^2y}{dt^2} + 6t\frac{dy}{dt} + y = \frac{dx}{dt} - (t-4)x$$

é aquele cujos parâmetros não mudam com o tempo

QUAIS SISTEMAS ABAIXO SÃO SLITs?

i. $y(t) = x(t+1)$

ii. $y(t) = 1/x(t)$

iii. $y(t) = 3\ddot{x}(t) + \dot{x}(t) - x(t)$

iv. $y(t) = \sin t \, x(t)$

v. $y(t) = x(t) + 2$

vi. $y(t) = \cos 2\pi \, x(t)$

vii. $y(t) = (\log t + 3t^2)x(t)$

viii. $y(t) = u(t) t \, x(t)$

REPRESENTAÇÃO DE EQUAÇÕES DIFERENCIAIS NO ESPAÇO DE ESTADO

Define-se como estado $x_1(t), x_2(t), \dots, x_n(t)$, o menor conjunto de variáveis que determinam completamente o comportamento do sistema para qualquer instante t.

As variáveis de entrada $u_1(t), u_2(t), \dots, u_m(t)$ são geradas por agentes externos (fontes) que alteram as condições de energia do sistema.

As variáveis de saída $y_1(t), y_2(t), \dots, y_p(t)$ são medidas por sensores instalados no sistema, são as variáveis controladas.

FORMA DO ESPAÇO DE ESTADO

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \qquad \text{equação dos estados}$$
$$y(t) = C(t)x(t) + D(t)u(t) \qquad \text{equação de saída}$$

a primeira derivada das variáveis de estado sempre está presente nas equações dinâmicas

$x(t)$- vetor de variáveis de estados (dimensão nx1);

$u(t)$ - vetor de variáveis de entrada (dimensão mx1);

$y(t)$ - vetor de variáveis de saída (dimensão px1);

$A(t)$- matriz de transmissão dos estados (nxn);

$B(t)$ - matriz de coeficientes de entrada (nxm);

$C(t)$ - matriz de coeficientes de saída ou matriz dos sensores (pxn);

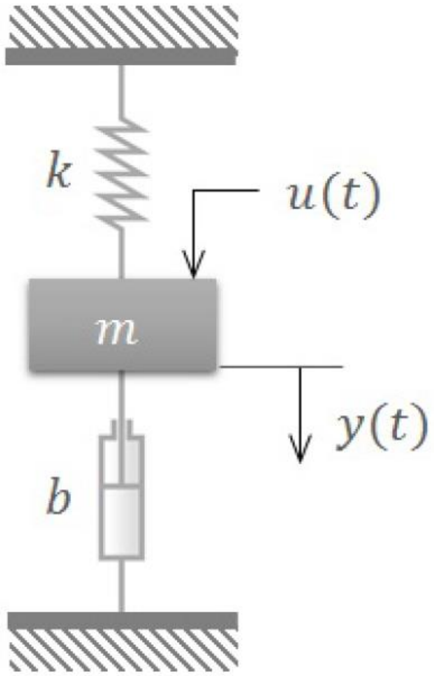
$D(t)$ - matriz de coeficientes de alimentação direta (pxm).

um sistema dinâmico de ordem n é representado por um conjunto de n equações diferenciais de primeira ordem.

EXEMPLO – SISTEMA MMA

ADAPTADO DE OGATA

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t)$$



$$x_1(t) = y(t)$$
$$x_2(t) = \dot{y}(t)$$

vetor de estados (n=2)

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

Estados estão associados com variáveis armazenadoras de energia no sistema.

entrada é um escalar (m = 1)

$$u_1(t) = u(t)$$

Entradas são variáveis que alteram as condições de energia do sistema.

saída é um escalar (p = 1)

$$y(t) = x_1(t)$$

Saídas são variáveis associadas com sensores, i.é, são variáveis medidas.

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m} & \frac{-b}{m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t)$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{m}(-kx_1 - bx_2) + \frac{1}{m}u_1(t)$$

$$y(t) = x_1(t)$$

$$A = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m} & \frac{-b}{m} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D = 0$$

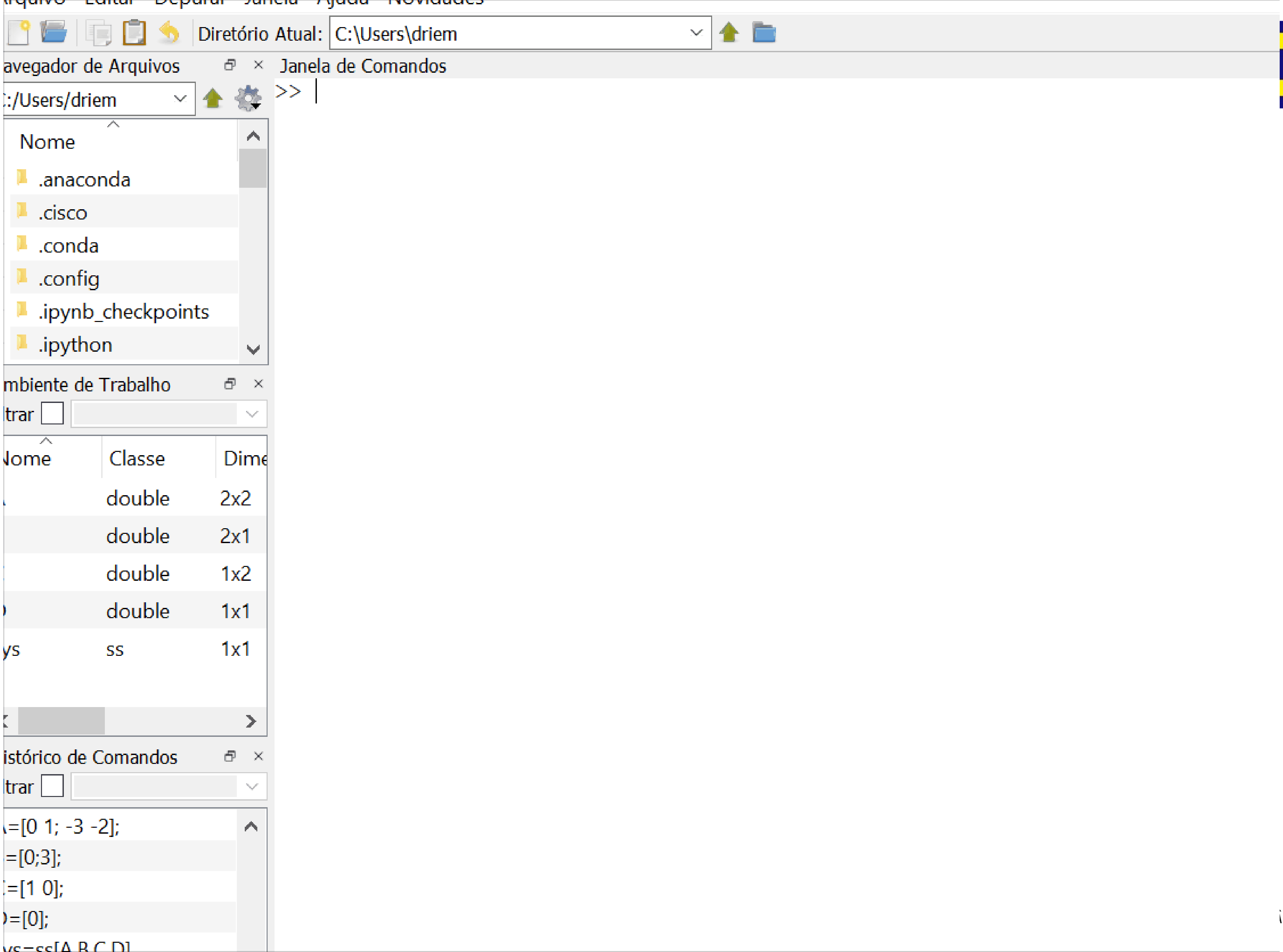
Existe uma classe própria no MATLAB para sistema lineares descritos na forma do espaço de estado criada pela função ss (que representa, em inglês, *state space*) e definida pelas matrizes A, B, C e D.

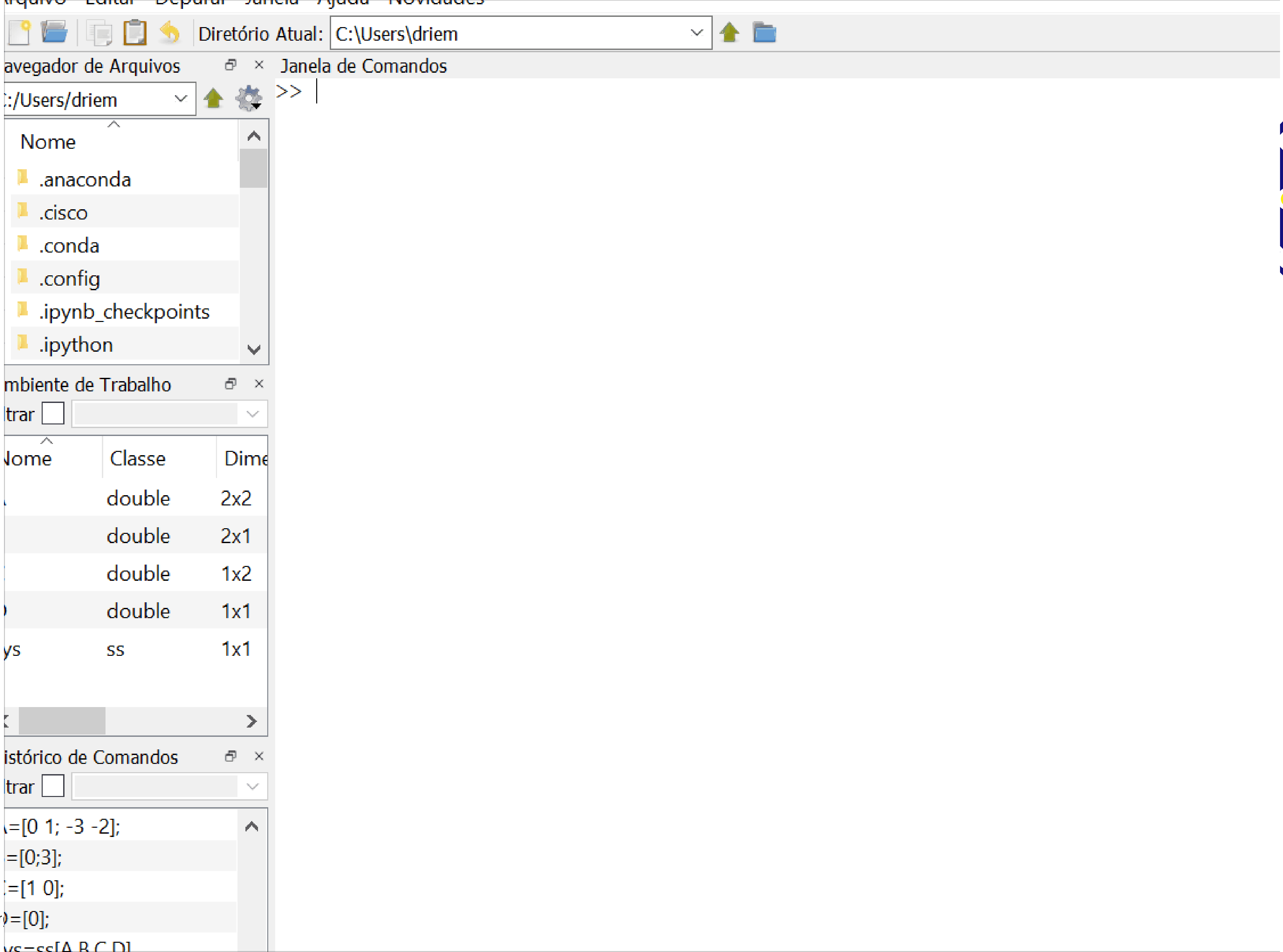
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -3 & -2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

```
» A = [0 1; -3 -2]
» B = [0; 3]
» C = [1 0]
» D = [0]
» sys = ss(A,B,C,D)
```

https://octave.sourceforge.io/control/function/ss.html


Diretório Atual: C:\Users\driem
avegador de Arquivos
Janela de Comandos
>>
:/Users/driem
Nome
.anaconda
.cisco
.conda
.config
.ipynb_checkpoints
.ipython
mbiente de Trabalho
ltrar
Iome
Classe
Dime
double 2x2
double 2x1
double 1x2
double 1x1
ys ss 1x1
istórico de Comandos
ltrar
=[0 1; -3 -2];
=[0;3];
=[1 0];
=[0];


Diretório Atual: C:\Users\driem
avegador de Arquivos
Janela de Commandos
>>
:/Users/driem
Nome
.anaconda
.cisco
.conda
.config
.ipynb_checkpoints
.ipython
mbiente de Trabalho
ltrar
lome Classe Dime
double 2x2
double 2x1
double 1x2
double 1x1
ys ss 1x1
istórico de Comandos
ltrar
=[0 1; -3 -2];
=[0;3];
=[1 0];
=[0];

SIMULAÇÃO DE SISTEMAS DINÂMICOS LINEARES

$$\delta(t) = \begin{cases} 0, & t \neq 0 \\ 1, & t = 0 \end{cases}$$

```
» impulse(sys);
```

simular a resposta a um impulso unitário

```
» impulse(sys,10);
```

valor em segundos para o tempo final de simulação

```
» [y t] = impulse(sys,10);
```

armazenar os vetores do tempo de simulação (criado automaticamente pelo MATLAB) e da resposta do sistema,

https://octave.sourceforge.io/control/function/impulse.html

Diretório Atual: C:\Users\driem

avegador de Arquivos

:/Users/driem

Nome
.anaconda
.cisco
.conda
.config
.ipynb_checkpoints
.ipython

mbiente de Trabalho

ltrar

Nome	Classe	Dime
	double	2x2
	double	2x1
	double	1x2
	double	1x1
ys	ss	1x1

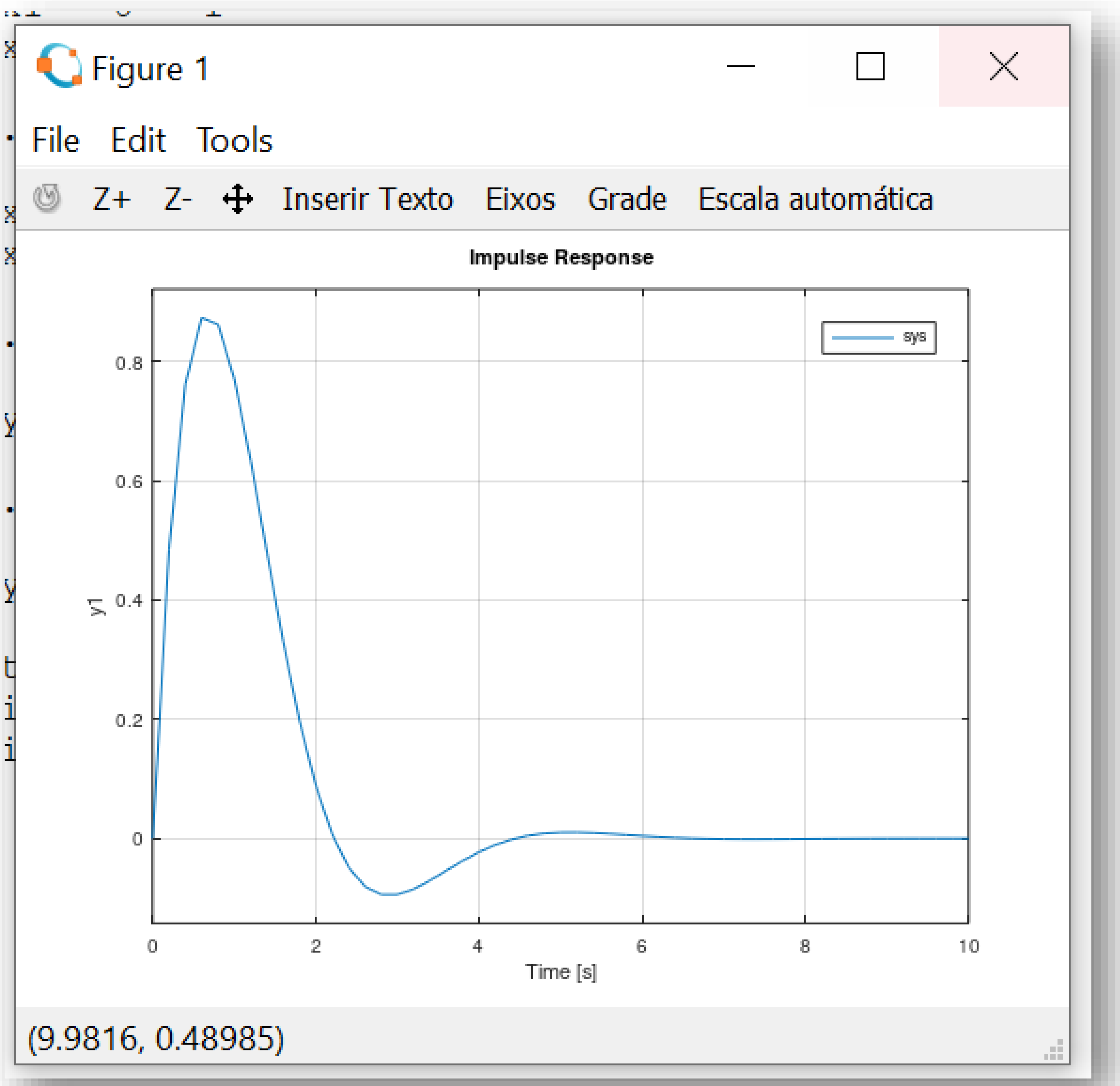
istórico de Comandos

ltrar

lc
kg load control
=[0 1;-3 -2];
=[0;3];
=[1 0];

Janela de Comandos

```
>> pkg load control
>> A=[0 1;-3 -2];
>> B=[0;3];
>> C=[1 0];
>> D=[0];
>> sys=ss(A,B,C,D)

sys.a =
        x1   x2
   x1    0    1
   x2   -3   -2

sys.b =
        u1
   x1    0
   x2    3

sys.c =
        x1   x2
   y1    1    0

sys.d =
        u1
   y1    0

Continuous-time model.
>> 
```



impulse(sys,10)

CONT...

A simulação da resposta a uma entrada em degrau unitário é feita pela função `step`,

```
» step(sys);

» [y t] = step(sys,10);
```

As funções `impulse` e `step` permitem que o usuário forneça um vetor de tempo a ser usado na simulação

```
t = 0:0.01:15;
step(sys,t);
```

https://octave.sourceforge.io/control/function/step.html avegador de Arquivos

:/Users/driem

Nome
.anaconda
.cisco
.conda
.config
.ipynb_checkpoints
.ipython

mbiente de Trabalho

trar

lome	Classe	Dime
	double	2x2
	double	2x1
	double	1x2
	double	1x1
's	ss	1x1

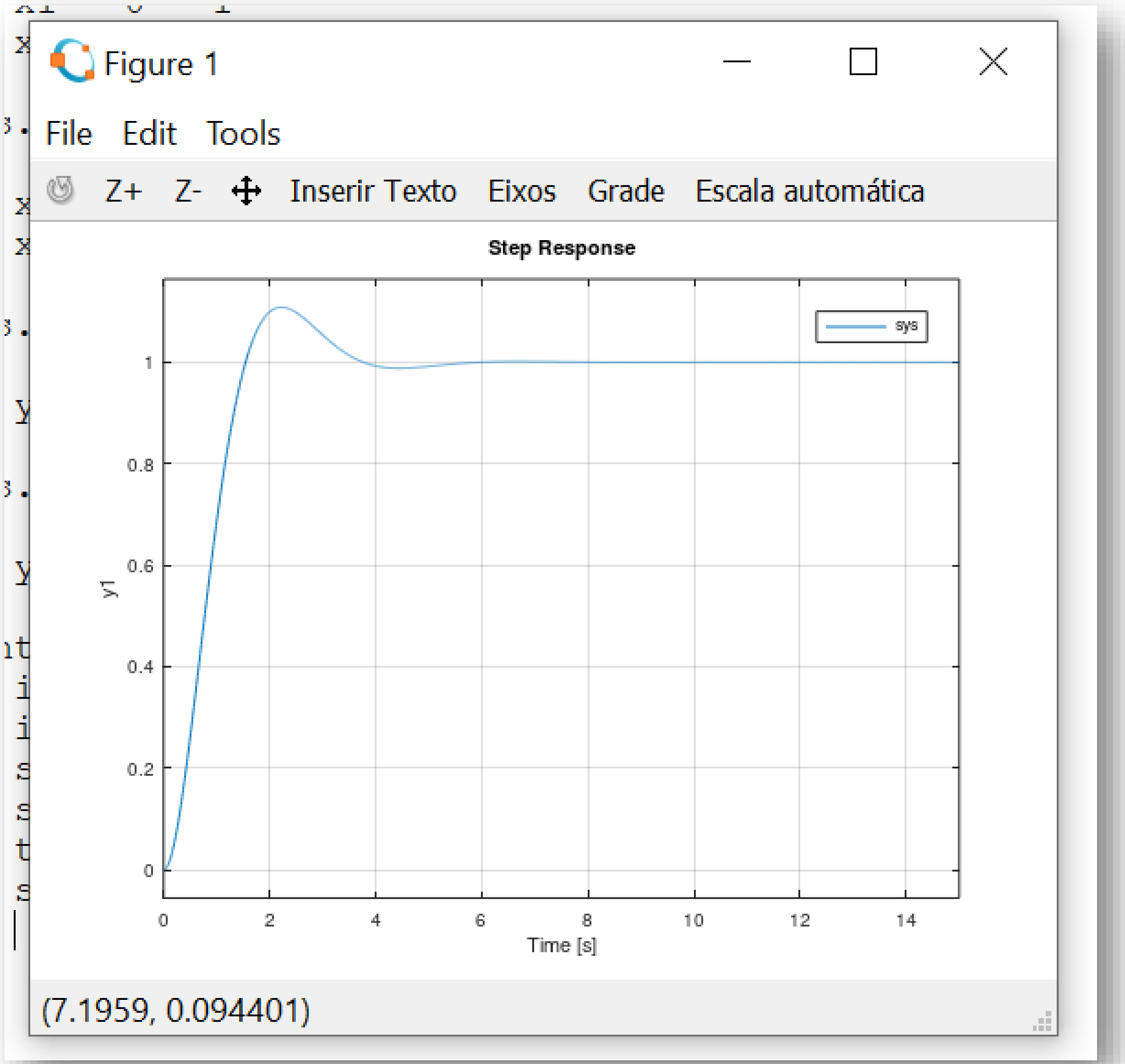
stórico de Comandos

trar

=[0;3];
=[1 0];
=[0];
/s=ss(A,B,C,D)
npulse(sys)
npulse(sys,10)

Janela de Comandos

```
>> pkg load control
>> A=[0 1;-3 -2];
>> B=[0;3];
>> C=[1 0];
>> D=[0];
>> sys=ss(A,B,C,D)

sys.a =
        x1   x2
   x1    0    1
   x2   -3   -2

sys.b =
        u1
   x1    0
   x2    3

sys.c =
        x1   x2
   y1    1    0

sys.d =
        u1
   y1    0

Continuous-time model.
>> impulse(sys)
>> impulse(sys,10)
>> step(sys)
>>
```




```
t=0:0.01:15;
step(sys,t)
```

ENTRADA GENÉRICA

Exemplo_slim_Octave.m

```
clear all; clc; close all;
pkg load control

m=1/3;
k =1;
b = 2/3;

a1=b/m; a0=k/m; b0=1/m;

t = 0:0.01:10;
u = 1.*[zeros(1,100),ones(1,length(t)-100)];
p1=plot(t,u,'r-.');
hold on

A = [0     1;
     -a0 -a1];
B = [0;
      b0];
C = [1 0];
D = 0;
sys = ss(A,B,C,D);
x0 = [0 0];
lsim(sys, u, t, x0);

```

Vejam o arquivo Exemplo_slim_Octave.m

Mudem a forma da entrada e as condições iniciais.

$$\frac{d^n}{dt^n}y(t) + a_{n-1}\frac{d^{n-1}}{dt^{n-1}}y(t) + \ldots + a_1\frac{d}{dt}y(t) + a_0 y(t) = b_m\frac{d^m}{dt^m}u(t) + \ldots + b_1\frac{d}{dt}u(t) + b_0 u(t) \tag{13}$$

onde

$b = [b_m, b_{m-1}, b_{m-2}, \ldots, b_1, b_0]$ é o vetor de coeficientes especificados no lado direito da Equação 13;

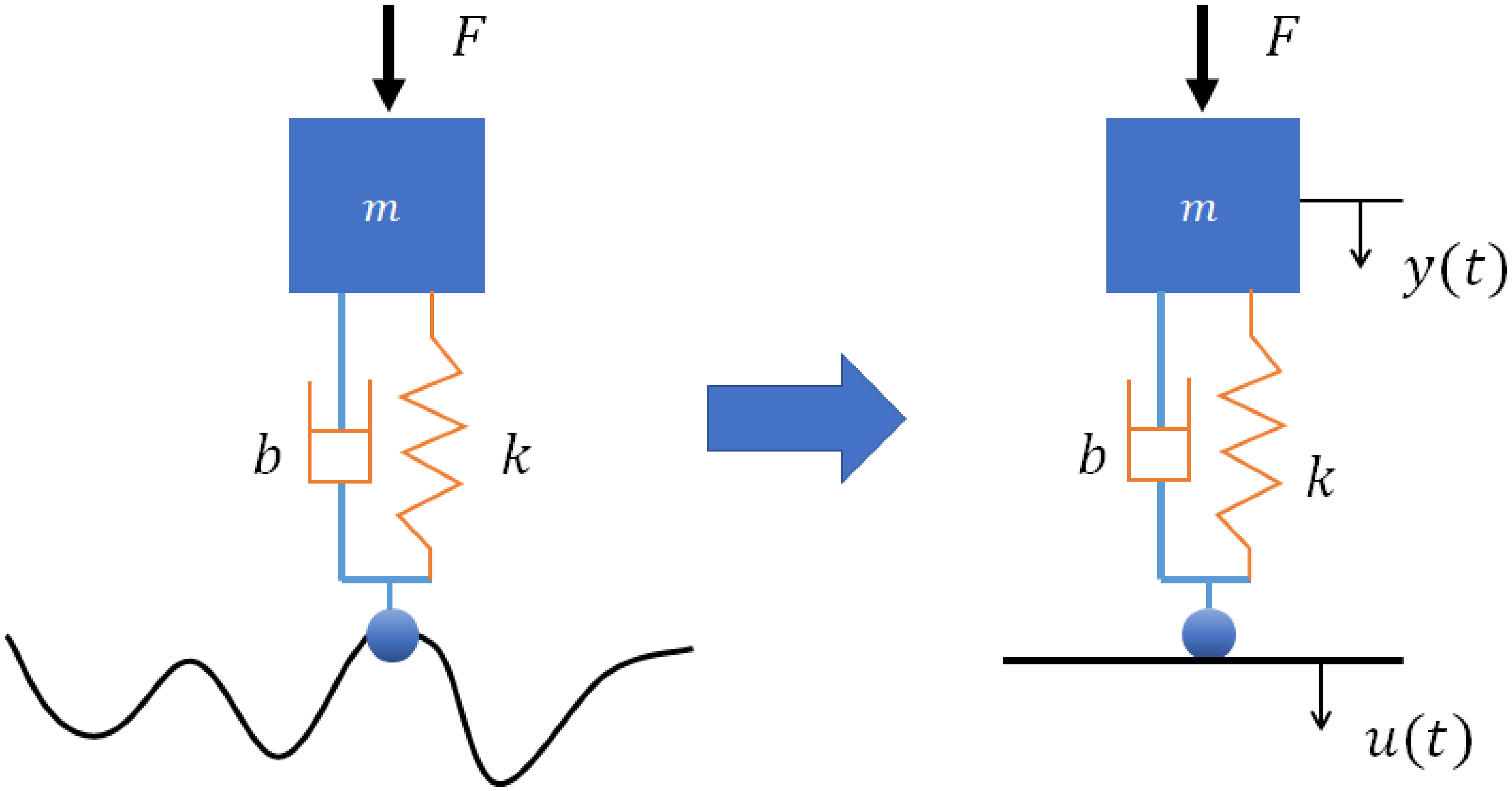
$a = [1, a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$ é o vetor de coeficientes do lado esquerdo da Equação 13;

$u =$ é o vetor de instantes conhecidos do sinal $u(t)$ especificados na Equação 13;

$t =$ vetor da mesma dimensão de u, o k-ésimo elemento $t(k)$ de t é o tempo, em segundos, no qual ocorre a entrada $u(k)$;

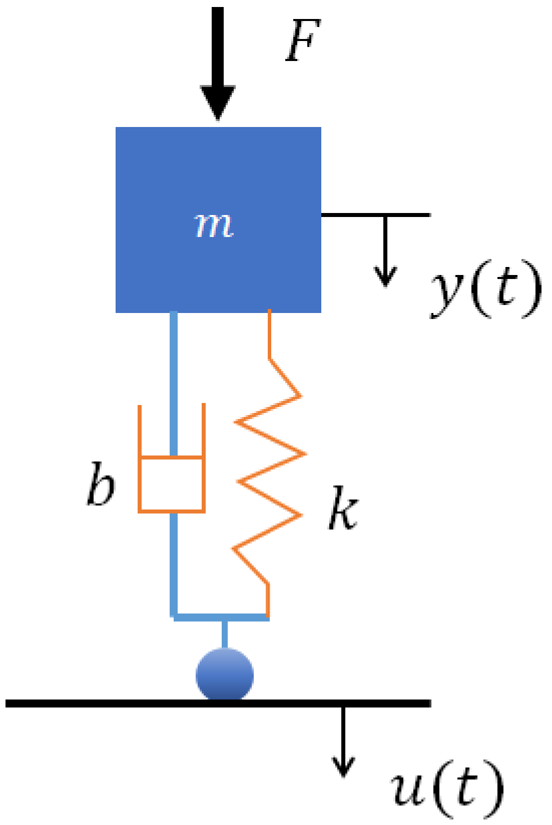
$y =$ vetor da mesma dimensão de u e t que representa instantes do sinal $y(t)$ que satisfazem a Equação 13.

EXERCÍCIO



$$m\ddot{y} + b(\dot{y} - \dot{u}) + k(y - u) = F$$

SCRIPT MATLAB



O script "*VeiculoComObstaculo*", disponível no STOA, exemplifica o caso ilustrado quando a perturbação $u(t)$ na pista é um degrau. Valores de massa, rigidez e amortecimento são, respectivamente, $m = 1000;\ k = 2000\ e\ b = 500$, definidos em unidades coerentes.

A título de ilustração, o amortecimento é duplicado e quadruplicado no segundo script.

SOLUÇÃO DE EQUAÇÕES DIFERENCIAIS NÃO LINEARES

- No MATLAB®, há diversas funções, chamadas solucionadores, do inglês solvers, que utilizam o método Runge-Kutta em passo variável para resolver equações diferenciais numericamente. Os dois solucionadores mais utilizados são a função ode45 e a função ode15s. A função básica, e que deve ser sempre testada primeiro, é a ode45, que utiliza combinação dos métodos de Runge-Kutta de quarta e quinta ordem. Se a solução da equação com esse solucionador apresentar problema de convergência ou erro, então utilize a função ode15s.
- No Octave, existe a ode45 e a ode23.

https://octave.sourceforge.io/octave/function/ode45.html

EDO DE PRIMEIRA ORDEM $\dot{y} = f(t, y) \qquad y(t_0) = y_0$

```
>> [tout,yout] = ode45(dydt,tspan,y0,options);
```

```
dydt=@(y,t) (y+3*t)/t^2; % funcao anonima
```

```
>> [tout,yout] = ode45(@ydot,tspan,y0,options);
```

```
function [ dydt ] = ydot(y,t);
  dydt=(y+3*t)/t^2;
end
```

$$\begin{cases} t^2\dot{y} & = y + 3t \\ y(1) & = -2 \end{cases} \qquad 1 \leqslant t \leqslant 4;$$

```
dydt=@(y,t) (y+3*t)/t^2;          % funcao anonima
tspan = [1 4];                    % vetor intervalo de integracao
y0 = -2;                          % condicao inicial
[tout,yout] = ode45(dydt,tspan,y0); % resolve o problema
plot(tout,yout)
```

```
dydt=@(y,t) (y+3*t)/t^2;                % funcao anonima
tspan = [1 4];                          % vetor intervalo de integracao
y0 = -2;                                % condicao inicial
[tout,yout] = ode45(dydt,tspan,y0);     % resolve o problema
plot(tout,yout)
```

EDO DE SEGUNDA ORDEM

$$5\ddot{y} + 7\dot{y} + 4y = u(t) \qquad 0 \leqslant t \leqslant 6 \qquad y(0) = 0 \text{ e } \dot{y}(0) = 9.$$

$$x(1) = y$$
$$x(2) = \dot{y};$$

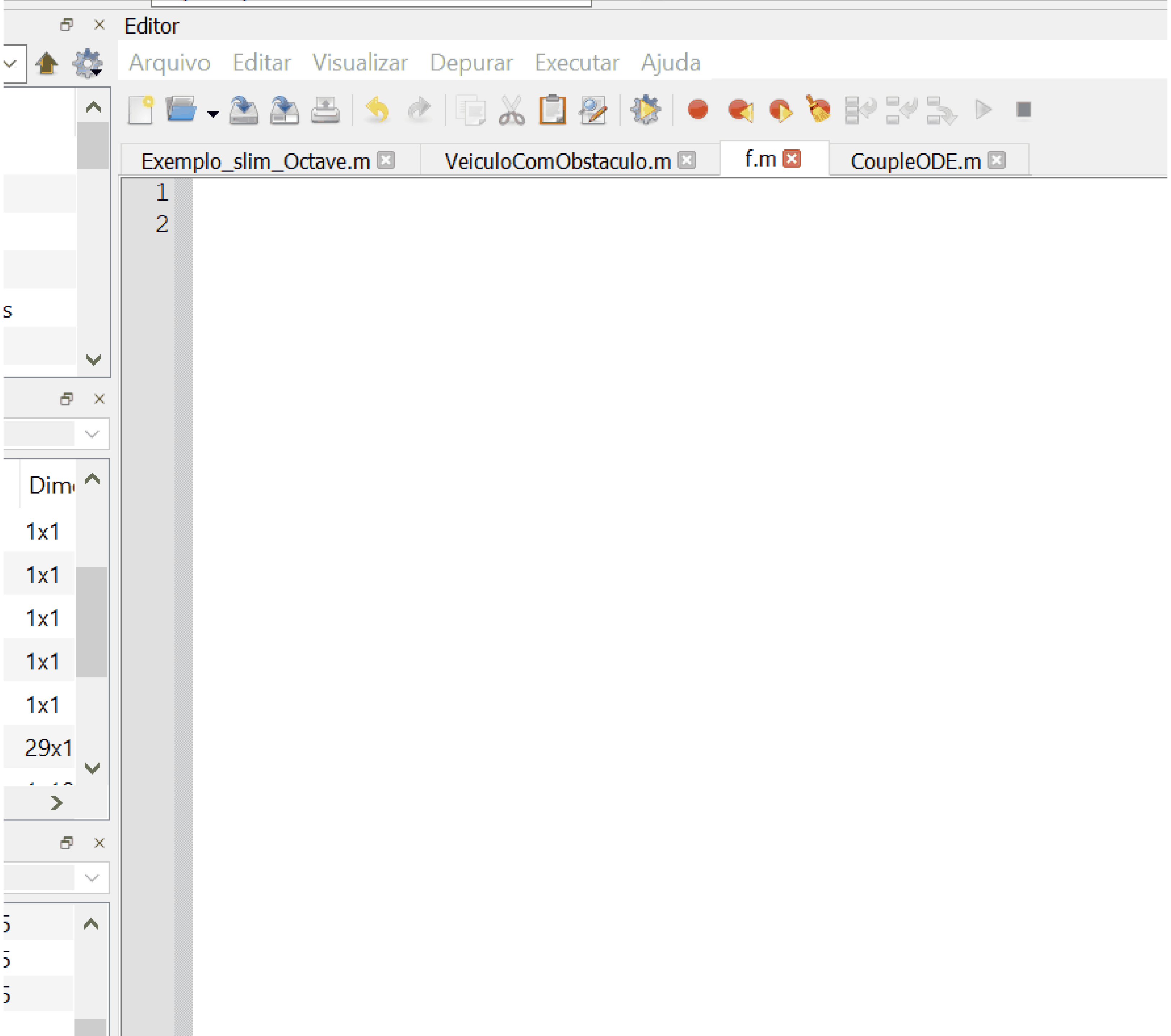
$$\dot{x}(1) = x(2)$$
$$\dot{x}(2) = \ddot{y} = \frac{1}{5}u(t) - \frac{4}{5}x(1) - \frac{7}{5}x(2)$$

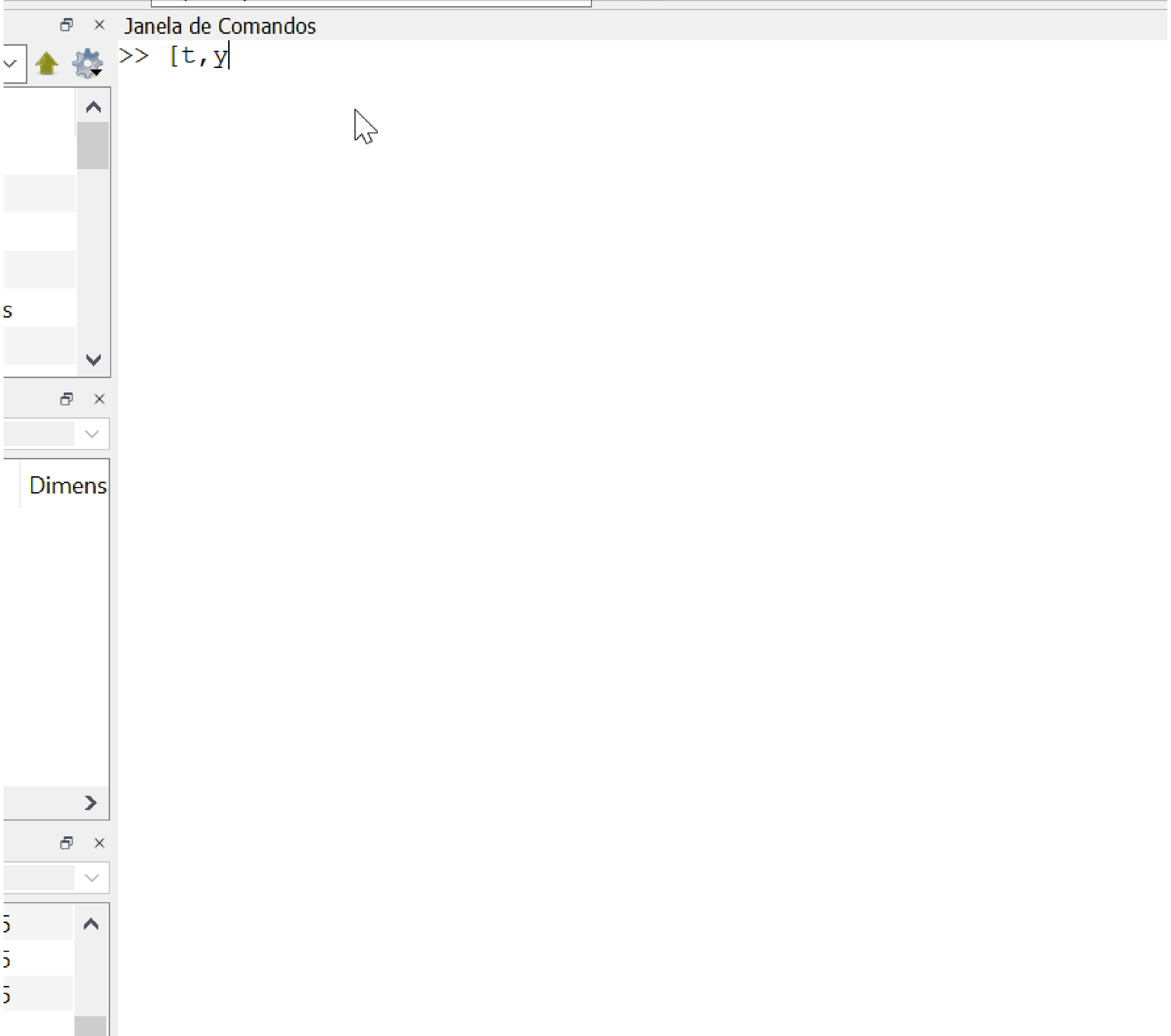
$u(t) = \sin t$

```
function xdot=estado_1(t,x)
  xdot=[x(2); (1/5)*(sin(t)-4*x(1)-7*x(2))];
end
```

```
[t,x]=ode45(@estado_1,[0,6],[0,9])
plot(t,x(:,1))}
```

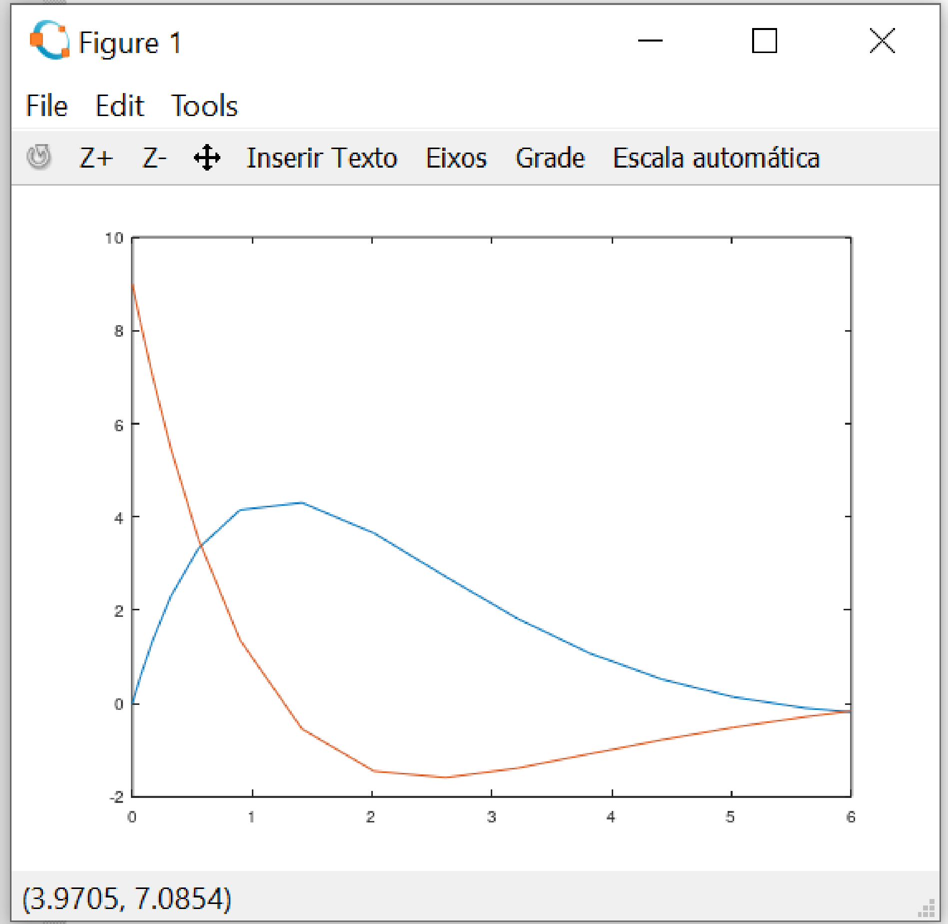
[t, y] = ode45 (fun, trange, init)


Editor
Arquivo
Editar
Visualizar
Depurar
Executar
Ajuda
Exemplo_slim_Octave.m
VeiculoComObstaculo.m
f.m
CoupleODE.m
Dim
1x1
1x1
1x1
1x1
1x1
29x1


Janela de Comandos
>> [t,y
Dimens

Exemplo_slim_Octave.m | * VeiculoComObstaculo.m | f.m | SegundaOrdem_ODE45.m

```
fxdot = @(t,x) [x(2); (1/5)*(sin(t)-4*x(1)-7*x(2))];
[t,x] = ode45(fxdot,[0,6],[0,9])
plot(t,x)
```



Posso simplesmente usar uma função anônima

EQUAÇÕES NÃO LINEARES ACOPLADAS

$$\ddot{y} = \dot{z}^2 + \tan z$$
$$\ddot{z} = \dot{y} + \dot{z} + \cos z$$

$$x(1) = y$$
$$x(2) = \dot{y}$$
$$x(3) = z$$
$$x(4) = \dot{z}$$

$$\dot{x}(1) = x(2)$$
$$\dot{x}(2) = x(4)^2 + \tan x(3)$$
$$\dot{x}(3) = x(4)$$
$$\dot{x}(4) = x(2) + x(4) + \cos x(3)$$

```
couplode = @(t,x) [x(2); x(4)^2 + tan(x(3)); x(4);  x(2)+x(4)+cos(x(3))];
[t,x] = ode45(couplode, [0 0.4999*pi], [0;0;0;0]);
figure(1)
% Lembrando que x = [y, ydot, z, zdot]
plot(t, x)
grid
str = {'$$ y $$','$$ \dot{y} $$', '$$ z $$', '$$ \dot{z} $$'};
legend(str, 'Interpreter','latex', 'Location','NW')
% ou
y=x(:,1);
ydot=x(:,2);
z=x(:,3);
zdot=x(:,4);
figure(2)
plot(t,zdot)
legend({'$$ \dot{z} $$'}, 'Interpreter','latex')
```

Arquivo *CoupleODE.m*
(no STOA e pag. 68 apostila)

Arquivo Editar Visualizar Depurar Executar Ajuda

lo_slim_Octave.m | PrimeiraOrdem_ODE45.m | * VeiculoComObstaculo.m | f.m | SegundaOrdem_ODE45.m | CoupleODE.m

```
couplode = @(t,x) [x(2); x(4)^2 + tan(x(3)); x(4);  x(2)+x(4)+cos(x(3))];
[t,x] = ode45(couplode, [0 0.4999*pi], [0;0;0;0]);
figure(1)
% Lembrando que x = [y, ydot, z, zdot]
plot(t, x)
```

Figure 1

Edit Tools

Z- Inserir Texto Eixos Grade Escala automática

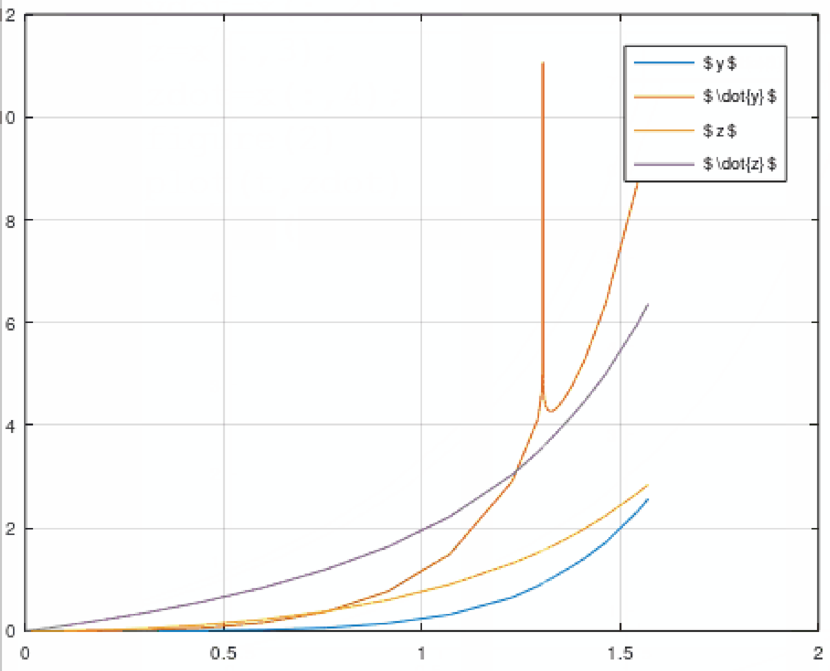
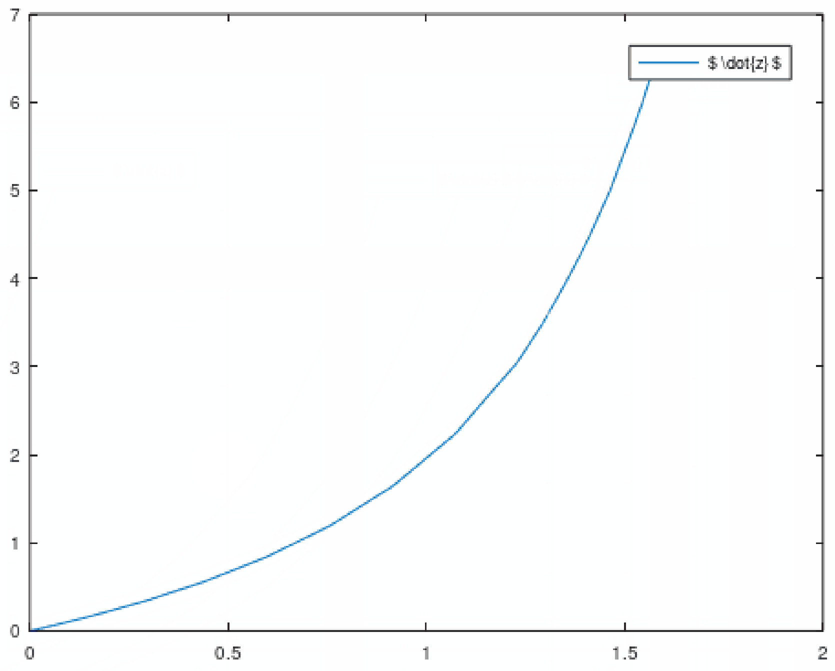


Figure 2

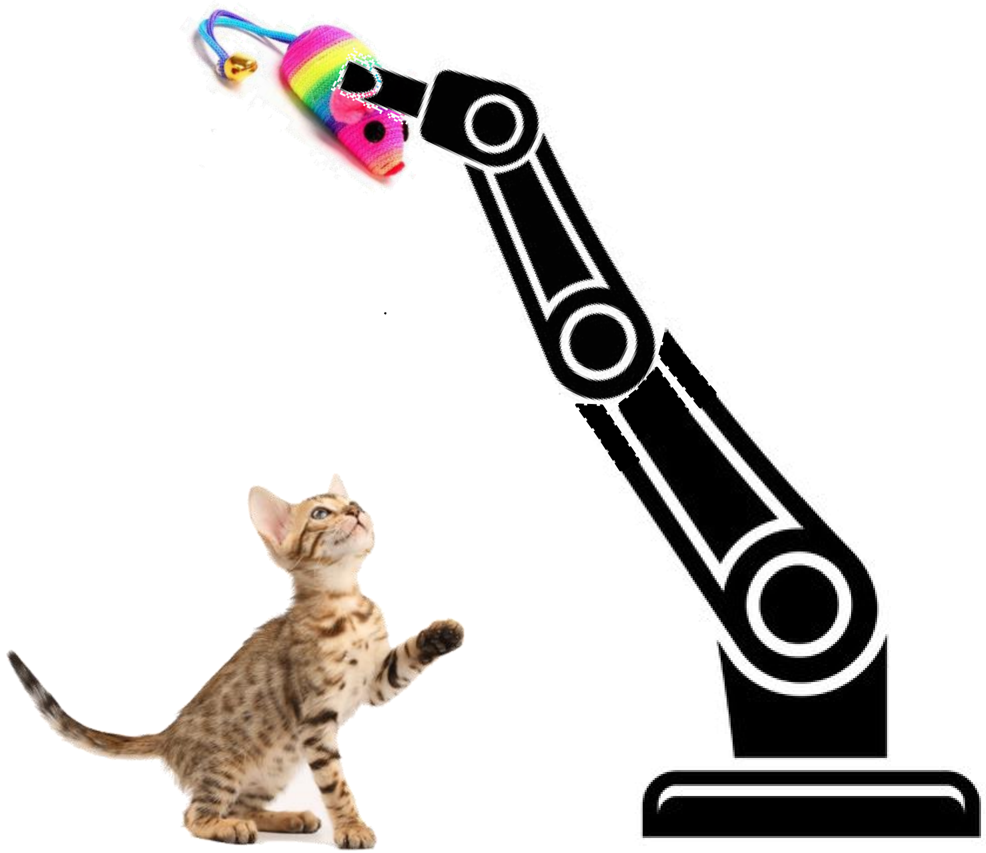
File Edit Tools

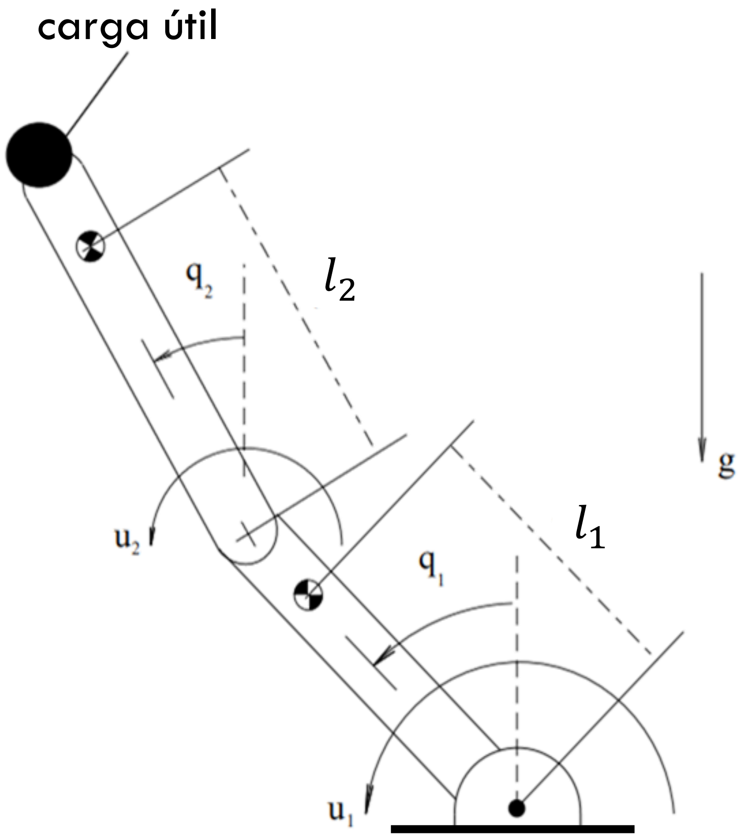
Z+ Z- Inserir Texto Eixos Grade Escala automática



ESTUDO DE CASO

BRAÇO ROBÓTICO





LIÇÃO DE CASA – ENTREGA NO MOODLE ATÉ 27/03

Defina o modelo cinemático do problema:

a) Defina as restrições do modelo;

b) Deduza as equações;

c) Defina as entradas e saídas do seu Sistema.

Faça um script que resolva o Sistema através da ODE45 e discuta os resultados.

FIM DO TERCEIRO MÓDULO