



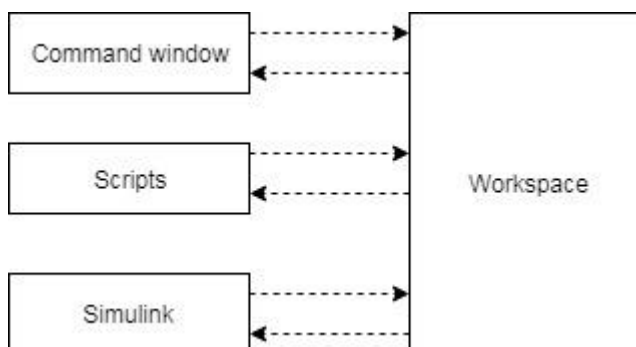
ROTEIRO DE LABORATÓRIO

1. Código da Experiência: 1A
2. Título: Introdução ao MATLAB®, Simulink® e QUARC®
3. Objetivos: Esta prática tem como objetivos:
 - Introduzir os conceitos básicos sobre o MATLAB®, Simulink® e QUARC®.
 - Introdução a programação por linguagens de blocos através do Simulink®.
4. Equipamento Utilizado: São necessários para realização desta experiência:
 - Um microcomputador PC com um os softwares necessários (Windows, MATLAB®/SIMULINK®, compilador C, QUARC®);
5. Primeiros Passos com o MATLAB®:

O MATLAB® é uma poderosa ferramenta matemática e IDE de desenvolvimento. Seu processamento é composto em suma por estruturas e equacionamentos matriciais. Alguns dos seus principais elementos são:

- Janela de comando (Command window);
- Área de trabalho (Workspace);
- Janela de Scripts;
- Ambiente Simulink;

Os códigos podem ser digitados diretamente na janela de comando, janela de scripts ou através de linguagem de blocos no Simulink®. Todos compartilham, leem e escrevem variáveis do workspace.



5.1. *Sintaxe Básica, Declaração de variáveis, operações matemáticas e funções*

Exemplo 1: Definição de variáveis e operações elementares

```
% Não é necessário declarar o tipo de variável, basta definir um
valor inicial.

a=4; % Ponto e virgula é usado para ocultar o valor da
variável na janela de comandos
b=2;

% Operações matemáticas elementares podem ser realizadas
normalmente
c1=a+b % adição
c2=a-b % subtração
c3=a*b % multiplicação
c4=a/b % divisão

% A variável ans armazena o resultado da ultima operação.

% Os comandos clear, close e clc devem ser usados para limpar o
workspace e a janela de comandos.
```

Operações com vetores e matrizes podem ser feitas da mesma forma, respeitando-se as devidas restrições matemáticas.

```
A=[1 2 3;4 5 6;7 8 9]; % Definição da matriz A 3x3
B=[3 2 1;6 5 4;9 8 7]; % Definição da matriz B 3x3
C=[1;0;1]; % Definição do vetor C 3x1

D1=A+B; %Adição Matricial
D2=A-B; %Subtração Matricial
D3=A*C; %Multiplicação Matricial
D4=A^2 %Eleva a matriz A ao quadrado A*A
D5=A^.2 %Eleva ao quadrado cada um dos elementos de A
E=rand(3); %Gera 3x3 com valores aleatórios entre 0 e 1

%Algumas operações extras com matrizes e vetores

A(2,1)=8; %Altera o elemento da segunda linha e primeira
coluna da matriz Aij em que i=2 e j=1.
A(:,1) = A(:,1)+C; %A primeira coluna da matriz A será
igual a própria coluna somada com os valores do vetor C.
A2 = A'; % Matriz Transposta
A3 = inv(A); % Matriz Inversa
d = det(A); % Determinante

% O comando help deve ser usado para consultar o funcionamento
de comandos e/ou funções
```

Exemplo 2: Um sistema físico pode ser representado por meio de matrizes através de suas variáveis de estado. Um sistema representado por variável de estado é definido por quatro matrizes A, B, C e D.

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

O MATLAB oferece diversas funções prontas que podem ser usadas em sistemas de controle:

```
A=[1 2 3;4 5 6;7 8 9];
B=[2;0;1];
C=[1 1 2];
D=0;
```

```

O=obsv(A,C); %Matriz de Observabilidade
W=ctrb(A,B); %Matriz de Controlabilidade
AV=eig(A); %Autovalores da Matriz A
r=rank(O); %Calcula o posto da matriz de Observabilidade
I=eye(size(A)) %Gera uma matriz identidade com a mesma
dimensão de A.
Z=zeros(3,3) %Gera uma matriz de zeros 3x3
%Algumas operações extras com matrizes e vetores
P=poly(A); %Calcula os coeficientes do polinômio
característico (sI-A)

```

Exemplo 3: Criando funções personalizadas através de uma janela de script

```

%A função myfunction recebe como entrada um valor escalar d e
fornece como saídas as matrizes A, B e C calculadas dentro da
função.

function [C,A,B] = teste(d)
    A = rand(d);
    B = rand(d)*10;
    C = A^2*B+B^.2;
end

```

5.2. Gráficos

Dado dois conjuntos de dados referentes aos eixos x e y, pode-se plotar o gráfico dos pontos.

Exemplo 4: Plotando gráfico

```

x=-2*pi:pi/10:2*pi;
y=sin(x);

figure(1); %rótulo para o gráfico 1
plot(x,y,'*'); %plota os pontos x,y
grid on; %grade
legend('Pontos xy'); %legenda
xlabel('x'); %descrição do eixo x
ylabel('y'); %descrição do eixo y

%Pode-se plotar gráficos múltiplos gráficos na mesma janela a
partir do comando hold on

hold on

y2=cos(x);
plot(x,y,'g'); %plota os pontos x,y
grid on; %grade
legend('Pontos xy'); %legenda
xlabel('x'); %descrição do eixo x
ylabel('y'); %descrição do eixo y

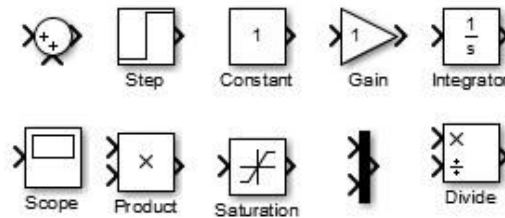
```

6. Primeiros passos com o Simulink®:

Simulink® é uma ferramenta para modelagem, simulação e análise de sistemas dinâmicos através de uma linguagem de blocos. O software oferece alta integração com o resto do ambiente MATLAB®.

6.1 Blocos básicos no contexto de sistemas de controle

O Simulink® possui um ambiente de desenvolvimento no qual é possível arrastar blocos para criar diagramas. No contexto da disciplina de controle, esses blocos são usados para realizar simulações de sistemas dinâmicos e efetuar o controle das plantas didáticas da Quaser®. Alguns dos blocos mais usados em controle são: somador; degrau; constante; ganho; operadores matemáticos elementares; saturação; integradores; multiplexador e scopes.

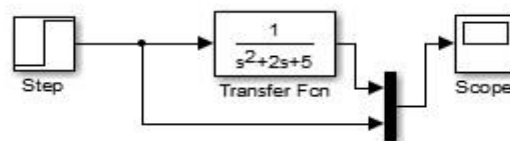


Exemplo 5: Usando blocos, tempo de simulação, uso do scope e exportação de variáveis para o workspace:

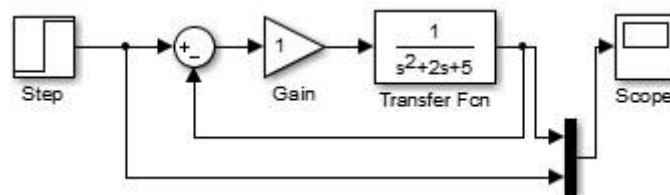
Exemplo 5.1: Resposta ao degrau de um sistema de segunda ordem



Exemplo 5.2: Resposta ao degrau de um sistema de segunda ordem, observando múltiplas curvas no scope



Exemplo 5.3: Resposta ao degrau de um sistema de segunda ordem, em malha fechada, varia-se o ganho (Ação Proporcional)

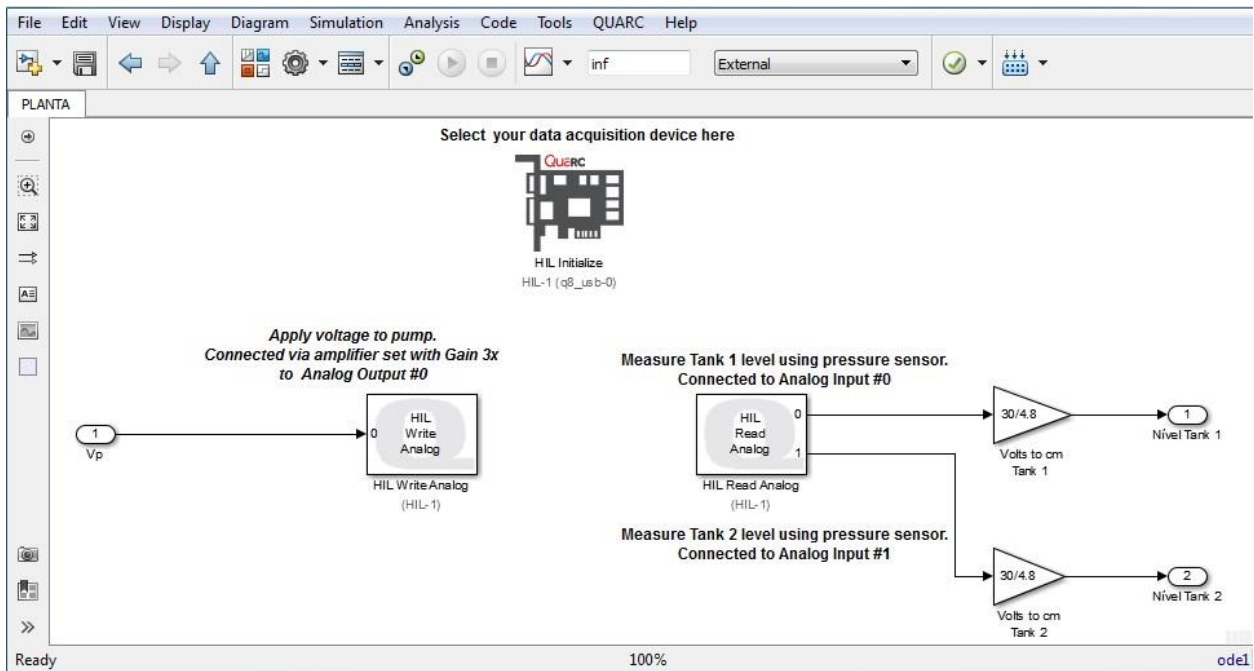


Exemplo 5.4: Exportar variáveis do Scope para o workspace

```
%Configurar o Scope para enviar os dados para o workspace (nome da variável =  
saida)  
plot(saida(:,1),saida(:,2));  
hold on  
plot(saida(:,1),saida(:,3));  
%Aplicar atributos ao gráfico gerado, conforme exemplo 4.
```

7. QUARC®: Quanser Real-time Control

O QUARC® é um software que fornece todas as ferramentas necessária para a realização de controle em tempo real das plantas da QUANSER®. O pacote fornece o driver de comunicação e utilitários que podem ser usados diretamente no Simulink®. Quando o QUARC® é instalado no sistema, novos blocos estarão disponíveis para serem utilizados. Os principais são os blocos de entrada e saída e o bloco de inicialização do driver de comunicação.



Exemplo 6: Usando os blocos fornecidos pelo driver QUARC®, gere uma tensão constante para o sistema de tanques acoplados QUANSER®, observe os níveis do tanque 1 e 2 associados a esse valor de tensão.

Obs 1: Copiar pasta Tanque e Renomear (20192TXGY), a pasta padrão do matlab deverá ser definido como a pasta do grupo.

Obs 2: Antes de executar a simulação verifique se todos os blocos e o tempo de simulação estão configurados corretamente. Em seguida, execute a simulação na sequência indicada na Figura abaixo. Coloque um SATURADOR entre o sinal Constant e o bloco da PLANTA.

