

Instituto Superior Técnico

Licenciatura em Eng^a Electrotécnica e de Computadores

Controlo

Introdução ao Matlab e Simulink

Elaborado por F. Garcia e M. Ortigueira - Setembro 2001

Reformulado por J. Gaspar - Setembro 2003

Secção de Sistemas e Controlo

Departamento de Engenharia Electrotécnica e de Computadores

Introdução ao MATLAB e Simulink

1.1 - Operações elementares

<ul style="list-style-type: none"> » <code>x=3+5</code> Atribui à variável x o valor da operação e mostra o resultado no monitor » <code>x</code> Mostra o valor de x no monitor » <code>4-10</code> Apenas apresenta o resultado da operação no monitor » <code>x=4*6;</code> O “;” impede o resultado de aparecer no monitor » <code>x=log(8)</code> Logaritmo natural. Usar “log10(x)” para calcular o logaritmo na base 10 de x. » <code>x=exp(log(8))</code> » <code>x=5^3</code> Eleva um número à potência 3 	<p>Números complexos:</p> <ul style="list-style-type: none"> » <code>z = 3 + j*4</code> tanto “i” como “j” definem a variável imaginária pura “sqrt(-1)” » <code>conj(z)</code> Conjugado » <code>abs(z)</code> Módulo » <code>angle(z)</code> Fase » <code>real(z)</code> Parte real » <code>imag(z)</code> Parte imaginária » <code>exp(i*pi)</code> Exponencial de um complexo
---	---

Faça “help elfun” para ajuda relativa a outras funções elementares. Nota: “Helpwin” pode ser usado no lugar de “help” para uma visualização numa janela individual.

1.2 - Manipulação de vectores e matrizes

<ul style="list-style-type: none"> » <code>x=exp(j*[pi/4:-0.1:-pi/4])</code> Cria um vector de elementos complexos » <code>x=[0 1 -1 3 4]</code> Cria um vector linha com os elementos indicados. » <code>x = x'</code> Cria o transposto do vector anterior. » <code>y=1:5</code> Cria um vector linha com inteiros ordenados de 1 a 5 » <code>y=5:-0.5:3</code> cria um vector decrescente com valores espaçados de 0.5 » <code>z=x*3</code> Multiplica todos os elementos do vector por 3 » <code>z=x.*y</code> produto elemento a elemento. 	<ul style="list-style-type: none"> » <code>sum(z)</code> soma dos elementos do vector » <code>z=x./y;</code> Divisão elemento a elemento. » <code>x = [1 2 3;4 5 6]; z = x*x'</code> Produto entre uma matriz e a sua transposta » <code>x = zeros(3,2)</code> Cria uma matriz de zeros de dimensão (3x2) » <code>x = ones(2,5)</code> Cria uma matriz de “1” de dimensão (2x5) » <code>size(x)</code> Devolve a dimensão de x
--	--

Faça help elmat para ajuda relativa a funções elementares com matrizes

1.3 - Representação gráfica de sinais representados por vectores

<ul style="list-style-type: none"> » <code>N=1000;</code> » <code>t=1:N;</code> cria um vector de dimensão N com os naturais de 1 a N. » <code>x=cos(2*pi*0.005*t);</code> cria um vector com os valores de uma sinusóide. 	<ul style="list-style-type: none"> » <code>plot(t,x);</code> » <code>plot(t,x,'*');</code> visualização discreta. » <code>plot(t(1:100),y(1:100),'*');</code> desenha, apenas, uma parte do sinal. » <code>t=0.1:0.1:20;</code>
---	---

Faça help graph2d para ajuda relativa a gráficos a duas dimensões

1.4 - Definição e análise no tempo e em frequência de SLITs contínuos através do Matlab

<p>» num = [2 1] » den = [1 2 3] » roots(den) Determina as raízes do polinómio cujos coeficientes correspondem aos elementos de "den" (neste caso será o polinómio $s^2 + 2s + 3$) » sys = tf(num,den) Cria o SLIT cuja função de transferência tem numerador "num" e denominador "den"</p>	<p>» step(sys) Resposta ao escalão do sistema "sys" » t = 0:1:10; u = sin(3*t); lsim(sys,u,t) Resposta de "sys" à entrada "u" definida nos instantes "t" » bode(sys) Desenha o diagrama de bode de amplitude e fase do sistema "sys"</p>
--	--

Faça Help de cada uma das funções anteriores para mais detalhes.

Faça help control para ajuda relativa a funções de Matlab para problemas de controlo.

1.5 - Criação de funções (ficheiros ".m")

As instruções podem ser encadeadas de forma a constituírem uma rotina ou função. **Devem ficar num ficheiro cujo nome terminará, necessariamente, por .m.**

Exemplos: $y = \exp(-t) \cdot \cos(t)$

Nome do ficheiro - fff.m

O conteúdo do ficheiro é o detalhado na caixa seguinte

```
function y = fff(x)
y = exp(-x).*cos(x);
```

Este ficheiro deve estar na directoria de trabalho (usar o comando "cd" para mudar de directoria)

Testar o ficheiro fazendo na linha de comandos

```
» t = 0:1:10; y = fff(t); plot(t,y)
```

1.6 - Simulação de sistemas em Simulink

Na janela de Matlab, o comando

```
» simulink
```

abre a janela "Simulink Library Browser" (designada em seguida por SLB) que contém os blocos de sistemas, entrada/saída e conexão (entre outros) para a simulação de sistemas (ver figura 1 - esquerda). Descrevemos aqui alguns dos blocos mais importantes.

A primeira operação a fazer consiste em abrir um espaço de trabalho Simulink, clicando com o mouse no botão "create a new model" da janela SLB. O sistema para simulação será criado fazendo "drag-and-drop" dos blocos da SLB para o espaço de trabalho Simulink.

A resposta a um SLIT de primeira ordem pode ser obtida da seguinte forma:

i) Usar "find" do SLB para encontrar cada um dos blocos "Transfer Fcn", "Step" e "Scope".

Fazer drag-and-drop destes blocos para a janela de trabalho. Ligar os blocos utilizando o mouse (ver fig 1 - meio).

ii) Duplo clique em "Scope" para abrir janela onde serão mostrados os resultados.

iii) Premir Ctrl+T para realizar a simulação e mostrar o resultado.

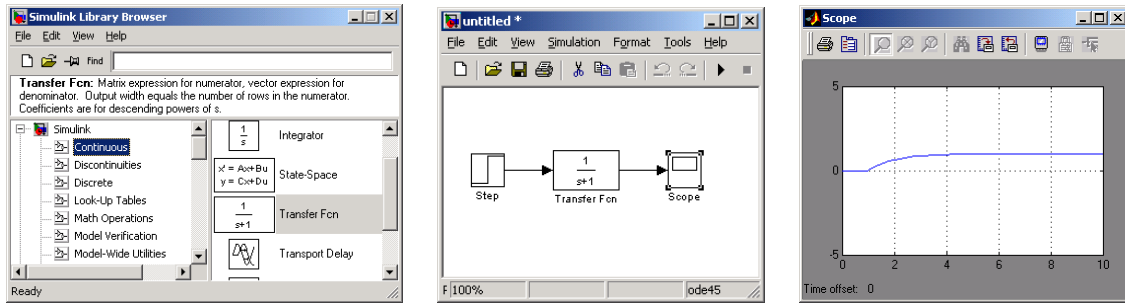


Figura 1: Simulink Library Browser (esquerda), exemplo de um sistema de primeira ordem (meio) e resposta do sistema ao escalão (direita).

Notas:

- Clicando duas vezes nos blocos com o mouse é possível modificar os parâmetros dos blocos.
- Os parâmetros de simulação são escolhidos clicando em "Simulation → Parameters" na janela do espaço de trabalho Simulink (convém, em particular, escolher o tempo de simulação desejado).

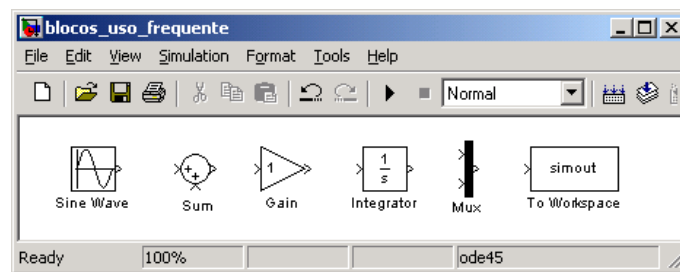


Figura 2: Blocos frequentemente utilizados.

Na figura 2 mostram-se mais alguns blocos frequentemente utilizados na simulação de sistemas de controlo. Os blocos indicados permitem colocar nos sistemas a simular: entradas sinusoidais (bloco "Sine Wave"), pontos de soma (bloco "Sum"), ganhos (bloco "Gain") e integradores (bloco "Integrator"). O bloco "Mux" é útil para juntar sinais e desta forma operá-los em conjunto. O bloco "To Workspace" permite obter os resultados da simulação no espaço de trabalho do Matlab, o que é útil por exemplo para criar gráficos de comparação entre resultados. No exemplo da figura 3 é simulado um sistema de segunda ordem, e o resultado é visualizado a partir da linha de comando.

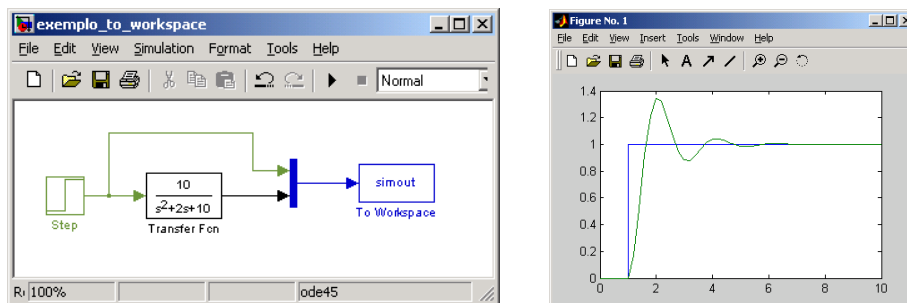


Figura 3: Simulação de um sistema de segunda ordem com exportação de resultados para o workspace. O gráfico é obtido com o comando de matlab: `plot(tout, simout.signals.values)`

1.7 - Nota final

Para obter mais detalhes sobre os comandos e os blocos simulink indicados ao longo deste documento, consultar a documentação on-line do matlab, existente na forma de ajuda rápida na linha de comando ("help" ou "helpwin") e em documentos organizados em páginas html (menu help).