MATLAB - Sintaxe Simbólica e aplicações em Sistemas Lineares

Paulo Oliveira Lenzi Valente

2016

Conteúdo

Introdução

Este guia tem como objetivo apresentar o leitor à linguagem simbólica do MATLAB, mostrando algumas de suas aplicações na análise de sistemas lineares.

1 Linguagem Simbólica

Variáveis e operações simbólicas são uma particularidade do MATLAB que permitem o programador operar variáveis de uma forma similar à que fazemos no papel. Assim, operações em tempo contínuo não precisam ser discretizadas.

Para se traçar um gráfico em programação, normalmente é necessário definir vetores discretos para os eixos vertical e horizontal, para então criá-lo. Os símbolos em MATLAB permitem que o usuário opere mais próximo do tempo contínuo, permitindo comandos como: `ezplot(x^2)` para criar o gráfico de uma parábola.

A seguir o leitor será apresentado ao código comentado para criar dois gráficos em uma mesma janela. No exemplo, o primeiro é da reta $x = 5t$. O outro, da parábola $y = t^2 + 5t$.

```
% primeiro, declaramos a variavel independente t, simbolica
syms t;

% outra forma de realizar a mesma declaracao seria ...
   utilizando a funcao sym, que possui outros parametros ...
   que podem ser estudados em seu manual. Um deles pode ...
   ser o tipo esperado para a variavel
% t = sym('t','real');

% a seguir, vamos criar as duas variaveis que guardarao ...
   nossas funcoes. Importante notar que nao precisamos ...
   declarar elas, apenas inicializar
y = t^2 + 5*t;
x = 5*t

% Com as funcoes em maos, vamos criar uma figura para ...
   guardar os graficos:
figure; % nos da uma nova janela
subplot(1,2,1); % divide a janela em 1 linha, 2 colunas e ...
   acessa a posicao 1.
ezplot(x,[-10 10]); % cria, no espaco atual, o grafico da ...
   nossa reta, com t variando entre -10 e 10
titulo = sprintf('Reta: x = %s',char(x)); % escrevendo a ...
   variavel que guarda o novo titulo
title(titulo); % modificando o titulo do grafico
grid on; % ativando a grade quadriculada

%abaixo, repetiremos o processo para o segundo grafico
%nao repetimos o comando figure para nao criar uma nova janela
subplot(1,2,2); % Escolhendo a segunda posicao do grafico ...
   ja existente
ezplot(y,[-10 10]); % cria, no espaco atual, o grafico da ...
   parabola, com t variando entre -10 e 10
titulo = sprintf('Reta: y = %s',char(y)); % escrevendo a ...
   variavel que guarda o novo titulo
title(titulo); % modificando o titulo do grafico
grid on; % ativando a grade quadriculada
```

Para finalizar a seção, um breve comentário sobre a função *ezplot*. Essa função aceita argumentos como equações paramétricas e funções simbólicas e cria o gráfico sem a necessidade de se definir valores discretos para o eixo horizontal nem para o parâmetro. Vale notar, também, que as opções para gráficos de uma variável podem ser encontradas ao se clicar com o botão direito em seu nome, na janela de *workspace* do MATLAB.

2 Análise de Sistemas Lineares

Antes de seguir com a seção, será necessário introduzir um novo tipo de variável. A classe *tf* difere da classe *sym* no sentido de que ela guarda informações a mais, embora também seja simbólica. A classe *tf* serve para representar funções de transferência.

A seguir, o código para criar uma função de transferência, de duas formas diferentes, e como realizar a conversão entre uma função de transferência do tipo *sym* para uma do tipo *tf*. Representaremos um sistema de exemplo com a função de transferência $\frac{s+1}{(s+2)(s+3)}$ com um zero em -1 e pólos em -2 e -3.

```
sistema_tf = tf([1 1],[1 5 6]);
% para descrever nosso sistema como tf, precisamos ...
   escrever os polinomios na forma reduzida, o que pode ...
   ser trabalhoso. Abaixo, escreveremos como sym e depois ...
   converteremos. Importante notar que a variavel s e ...
   reservada para tf
syms y
sistema_sym = (y + 1)/((y + 2)*(y + 3));
[numerador, denominador] = numden(sistema_sym); % a funcao ...
   separa numerador e denominador de uma expressao simbolica
numerador_tf = sym2poly(numerador); % sym2poly converte ...
   uma expressao simbolica para a notacao vetorial de ...
   polinomios
denominador_tf = sym2poly(denominador);
sistema_tf = tf(numerador_tf,denominador_tf); % tendo em ...
   maos as notacoes vetoriais, podemos criar a variavel tf ...
   que representa nosso sistema
```

Agora que sabemos representar nosso sistema dentro do MATLAB, veremos uma lista de funções que podem ser úteis na análise de sistemas lineares. Os exemplos assumirão *sys* como uma *tf* e *simbolo* como expressão simbólica, ambas representando a funcao de transferência do sistema.

```
bodeplot(sys); % diagrama de bode
pzmap(sys); % diagrama de polos e zeros
stepplot(sys); % resposta no tempo a um degrau unitario. ...
   Nao funciona para funcoes com mais zeros do que polos
stepplot(sys / s); % resposta no tempo a uma rampa causal. ...
   Nao funciona para funcoes com mais zeros do que polos ...
   pois tambem se baseia em steplot.

% Encontrando a resposta a uma entrada x(t)
resposta_freq = laplace(x) * simbolo;
resposta = ilaplace(resposta_freq);
ezplot(resposta);

%laplace inverso de uma constante:
syms t x y;
ilaplace(1); % resulta em erro
ilaplace(1,t); % usa a variavel s com padrao e tenta levar ...
   para t
ilaplace(1,y,x); % tenta levar do dominio y para x com a ...
   transformada inversa de laplace. Resulta em dirac(x)
% a logica de escolha do dominio de origem para o dominio ...
   de destino acima equivale para laplace, fourier e ifourier

%Funcoes possivelmente uteis:
heaviside(t) % degrau unitario com valor 1/2 em 0;
% pode-se alterar o valor de heaviside em 0 com:
% sympref('HeavisideAtOrigin', valor_na_origem);
dirac(t) % impulso unitario

% Assim como temos laplace e ilaplace realizando a ...
   transformada de laplace e sua inversa, temos:
fourier(x);
ifourier(X);

%Por fim, a podemos substituir valores especificos em uma ...
   dada funcao simbolica:
syms t a b;
y = a*t + b;
a = 5;
b = 0;
y = subs(y); % agora, y = 5*t;

syms x;
y = subs(y,x); % trocamos a variavel independente t por x
```

Referências

[1] Página de Sistemas Lineares da Mathworks:
`http://www.mathworks.com/help/control/examples/plotting-system-responses.html`

[2] Página sobre a linguagem e funcionalidades do MATLAB:
`http://www.mathworks.com/help/matlab/index.html`

[3] Minicurso introdutório de MATLAB:
`http://polvalente.github.io/files/matlab_basico.pdf`