MATLAB / CONTROLO

Função de Transferência -- Resposta Temporal -- Root-Locus

1 - Introdução

Este trabalho tem como principal objectivo a familiarização dos alunos com algumas das capacidades do **MATLAB** e da sua *"Control System Toolbox*".

2 - Apresentação do Matlab

O Matlab tendo uma interface simples e intuitiva, através de linha de comandos, apresenta as seguintes características:

1. As Variáveis são tratadas como escalares, vectores ou matrizes de uma forma transparente. Pode-se pensar no **MAT**rix **LAB**oratory como uma linguagem que foi criada para a manipulação de matrizes.
2. Tem funções predefinidas, que podem ser utilizadas para resolver vários tipos de problemas.
3. Considera de forma diferente os caracteres maiúsculos e minúsculos
4. Dispõe de um HELP a que se deve recorrer sempre que necessário.
5. As funções de transferência são definidas a partir dos polinómios do numerador e denominador, os quais são representados por vectores com os seus coeficientes.

3 - Trabalho a Efectuar

NOTA: Para que o aluno compreenda devidamente cada um dos comandos do Matlab, é necessário que, antes de os utilizar, observe os seus conteúdos recorrendo ao **help.**

Exemplo: se desejar saber o conteúdo de uma instrução, faça:
»help *nome da função*

4 - Complexos e Matrizes

4.1 - Números Complexos

O MATLAB trabalha com números complexos. Por defeito as variáveis i e j estão definidas como $\sqrt{-1}$. Em engenharia electrotécnica utiliza-se normalmente o símbolo j para a parte imaginária.

```
» j
ans =
         0 + 1.0000i
» i
ans =
          0 + 1.0000i
```

De notar que o MATLAB privilegia o símbolo i.

4.1.1 Representação dos números complexos

O MATLAB pode representar graficamente os números complexos.

```
»  a=2+3*j
»  b=3+4j
»  c=a+b
»  d=a*b
» compass([a,b,c,d])
```

Nota: para apresentar a janela de gráfico escrever `shg`. Para limpar a janela de gráfico escrever `clg`.

4.1.2 Ainda sobre números complexos

Alguma teoria!

Se z é um número complexo $z = x + jy$, então a forma polar de z é $z = re^{j\theta}$ onde $r = \sqrt{x^2 + y^2}$ é o comprimento do vector complexo $|z|$ e θ é o ângulo de z, $\theta = \tan^{-1}(y/x)$ (radianos).

O MATLAB lida com a forma exponencial (ou polar) de um número complexo como com a forma cartesiana.

```
» z = 1 + 2*j;
» r = abs(z)
» theta = angle(z)
```

Agora, quer $e^{j\theta}$ e $r(\cos\theta + j\sin\theta)$ deverá ser igual a *z*.

```
»  w1 = r*exp(j*theta)
»  w2 = r*(cos(theta)+j*sin(theta))
```

O MATLAB usa radianos para todas as funções trigonométricas (não existe modo de definição da grandeza angular como existe nas máquinas calculadoras cientificas). Para converter um ângulo em radianos para graus, ou *vice-versa* lembrar a fórmula simples:

```
»  tg = theta*180/pi % radianos para graus
»  tr = tg*pi/180 % graus para radianos
```

4.2 - Matrizes

```
»  A = [0, 1<Enter>
-2, -3] <Enter>
```

Para matrizes pequenas, pode-se usar a sintaxe alternativa:

```
»   A=[0, 1; -2, -3]
```

4.2.1 Definição de Vectores

Os vectores definem-se de forma idêntica às matrizes, assim pode-se definir o vector linha **b** como

```
»  b=[1, 2]
```

e podemos definir o vector coluna **c** como

```
»   c=[3; 4]
```

4.2.2 Funções para Matrizes

O MATLAB é bastante poderoso no cálculo de operações com matrizes, o que o torna muito útil em aplicações onde se têm de resolver sistemas de equações.
Por exemplo, a matriz inversa, o seu determinante e os valores próprios podem determinar-se facilmente

```
»   inv(A)   % A matriz inversa de A
»   det(A)   %  Determinante de A
»   eig(A)   %  Valores próprios de A
```

5 - Função de transferência

Considere a seguinte Função de Transferência :

$$G(s) = \frac{(s+1)}{(s+2)(s+3)} = \frac{(s+1)}{(s^2+5s+6)}$$

escreva no MATLAB:

```
»num=[1 1]
»den=[1 5 6]
```

em que **num** e **den** são vectores que contêm os coeficientes dos polinómios do numerador e do denominador da função de transferência.

Para obter a função de transferência de um sistema a partir da equação do numerador e denominador, escreva:

```
» sys=tf(num,den)
```

A variável **sys** é um objecto tipo função de transferência.

Para ver as propriedades dos sistemas lineares e invariantes no tempo (LTI) escreva:

```
» help ltiprops
```

```
»b=roots(den)
```

Permite encontrar as raízes do polinómio característico.

```
»c=poly(b)
```

Retorna o vector **den.**

```
»[z,p,k]=tf2zp(num,den)
```

Permite obter os zeros, pólos e o ganho da função de transferência.

```
»[num,den]=zp2tf(z,p,k)
```

Transforma os zeros, pólos e o ganho na função de transferência correspondente.

```
»[r,p,k]=residue(num,den)
```

Obtém os resíduos, os pólos e os termos constantes da expansão em fracções parciais da função de transferência G(s).

Para obter a função de transferência em malha fechada quando a realimentação é negativa e unitária, escreva:

```
» sys1=feedback(sys, 1)
```

Considere que G(s) é a função de transferência da malha para a frente e que $H(s)=\frac{1}{s}$ é a função de transferência da malha de realimentação negativa.

```
»num1=[1]
»den1=[1 0]
» sys1=tf(num1, den1)
```

Para obter a Função de Transferência em malha fechada do sistema faça:

```
» sysmf=feedback(sys,sys1)
```

Confirme analiticamente o resultado obtido.

O MATLAB por defeito assume realimentação negativa. Se se pretende saber a função de transferência do sistema com realimentação positiva faz--se:

```
» sysmfrp=feedback(sys,sys1,+1)
```

Para conhecer outras funções relacionadas com a simplificação de diagramas de blocos escreva:

```
»help series
»help parallel
```

6 - Resposta Temporal

Considere um sistema com a seguinte função de transferência : $G(s)=\frac{2}{s^2+2s+2}$

```
» sys=tf([2],[1 2 2])
»[wn,z]=damp(sys)
```

Permite determinar a frequência natural não amortecida (w_n) e o coeficiente de amortecimento(z) do polinómio **den**.

Desenhe o gráfico da resposta temporal a uma entrada degrau unitário, escrevendo:

```
»step(sys)
```

Para obter a resposta temporal a uma entrada degrau com amplitude 5 durante 20 segundos, escreva:

```
»step(5*sys,20)
```

Desenhe o gráfico da resposta temporal a uma entrada impulso, escrevendo:

```
»impulse(sys)
```

```
»u=0:0.1:10;
```

u é um vector com valores entre 0 e 10 com incrementos de 0.1. Observe que o facto de ter concluído a instrução com **";"** , tem como consequência a não visualização do resultado da instrução.

```
»lsim(sys,u,u)
```

Esta instrução permite desenhar o gráfico da resposta de um sistema a entradas arbitrárias. Nesta situação, a entrada é uma rampa unitária.

Altere a legenda no eixo das abcissas escrevendo:

```
»xlabel(‘Tempo(seg)’)
```

Altere a legenda no eixo das ordenadas recorrendo a :

```
»ylabel(‘amplitude’)
```

Coloque um título no gráfico utilizando o seguinte comando:

```
»title(‘Resposta Temporal’)
```

Modifique o limite superior e inferior dos eixos usando:

```
»axis([0 12 0 15])
```

7 - Lugar Geométrico das Raízes

Considere a seguinte função de transferência em malha aberta,

$$G(s)H(s) = \frac{s+1}{s(s+10)}$$

que é representada no MATLAB, como:

```
»sys=tf([1 1],[1 10 0])
```

Note que mesmo não aparecendo no denominador o 3º coeficiente, este deve ser colocado, no seu vector representativo, como zero.

Obtenha o Root-Locus

```
»rlocus(sys)
```

Determine as raízes da equação característica para k=10.

```
»k=10
»r= rlocus(sys,k)
```

Para determinar a localização das raízes de forma interactiva no Root-locus, faça:

```
»rlocfind(sys)
```

Para determinar a localização, no Root-locus, dos pontos com coeficiente de amortecimento igual a 0.5 e frequência natural não amortecida igual a 5 rad/s, faça:

```
»sys=tf([1],[1 10 5])
»rlocus(sys)
»sgrid(0.5,5)
```

»ltiview

Interface do utilizador para estudar os sistemas LTI existentes no espaço de trabalho nas suas componentes de espaço de estados, função de transferência e ganho-zero-pólo.