UNIVERSIDADE ESTADUAL DO RIO GRANDE DO SUL
CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL

SISTEMAS II
Prof. Leo Weber

Laboratório 03: Sistemas de Dados Amostrados e Controle Digital

1. Introdução:

Neste laboratório serão apresentados os recursos e comandos para a manipulação e análise da estabilidade de sistemas da dados amostrados visando o projeto de controladores digitais com a *Toolbox de Sistemas de Controle, The Student Edition of Matlab - version 5.3.*

O roteiro previsto para este laboratório inclui:

- Representação de sistemas discretos pelo Matlab;
- Análise da estabilidade pelo Root Locus no Plano Z;
- Análise da estabilidade pelos Diagramas de Bode no Plano V;
- Apresentação do comando *step* para a determinação da resposta no tempo ao degrau unitário de um sistema amostrado;

2. Representação de um sistema discreto pelo Matlab:

Uma função de transferência pode ser expressa como um polinômio no numerador dividido por um polinômio no denominador, isto é, *F(s) = N(s) / D(s)*. Ambos, numerador e denominador, podem ser representados por vetores linha *numf* e *denf*. Outrossim, pode-se formar *F(s)* com o comando *F = tf (numf, denf)*, onde *F* é chamado um objeto linear e invariante no tempo (em inglês *LTI – Linear Time Invariant object*). Para criar sistemas discretos no tempo usando Matlab, basta especificar o período de amostragem associado ao sistema como o último parâmetro desta função. Exemplificando:

```
EDU» numf = 150*[1  2  7];
EDU» denf = [1  5  4  0];
EDU» F = tf(numf,denf,0.01)
```

Também pode-se criar funções de transferência em que o numerador e o denominador estão representados na forma fatorada. Neste caso, *G(s) = K * N(s) / D(s)* pode ser expresso usando o comando *G = zpk (numg, deng, k)*, onde:

numg = vetor linha contendo as raízes de *N(s)*, que são os zeros

deng = vetor linha contendo as raízes de *D(s)*, que são os pólos
k = ganho da função de transferência

Da mesma forma que a função anterior, para criar sistemas discretos no tempo usando Matlab, basta especificar o período de amostragem associado ao sistema como o último parâmetro. Exemplificando:

```
EDU» numg = [-2  -4];
EDU» deng = [-7  -8  -9];
EDU» k = 20;
EDU» G = zpk(numg, deng, k, 0.01)
```

Podemos converter uma função de transferência expressa no domínio da variável contínua s, G1(s), em cascata com um reconstrutor de ordem zero (*zoh*) em uma função de transferência no domínio da variável discreta z, G(z), usando o comando do Matlab *G = c2d(G1,T,'zoh')*, onde:

G1 = objeto de sistema contínuo LTI
G = objeto de sistema amostrado LTI
T = tempo de amostragem
'zoh' é um método de transformação que supõe G1(s) em cascata com um zoh

Colocamos simplesmente G1(s) no comando (o *zoh* é levado em conta automaticamente) e o comando retorna G(z). Exemplificando:

```
EDU» T=1;
EDU» numg1s=[1];
EDU» deng1s=[1 1 0];
EDU» G1=tf(numg1s,deng1s)
EDU» G=c2d(G1,T,'zoh')
```

Podemos utilizar o MATLAB para converter G(s) em G(z) quando G(s) não está em cascata com um amostrador de ordem zero (*zoh*). O comando visto anteriormente *H = c2d(F,T,'zoh')* transforma F(s) em cascata com um z*oh*, onde
*H(z) = ((z - 1)/z)*Z{F(s)/s}*.

Se fizermos *F(s) = sG(s)*, o comando resolve H(z), onde
*H(z) = ((z - 1)/z)*Z{G(s)}*.

Portanto, *Z{G(s)} = (z/[z - 1])*H(z)*.

Em resumo, entre com *F(s) = sG(s)* e multiplique o resultado de *H = c2d(F,T,'zoh')* por *(z/[z -1])*. Este processo é equivalente a obter a transformada Z de G(s). Exemplificando:

```
EDU» T=1;
EDU» numgs=[1];
EDU» dengs=[1 1 0];
EDU» Gs=tf(numgs,dengs)
EDU» Fs=Gs*tf([1 0],1);
EDU» Fs=minreal(Fs);
```

```
EDU» Hz=c2d(Fs,T,'zoh');
EDU» Gz=Hz*tf([1 0],[1 -1],[]);
EDU» Gz=minreal(Gz)
```

O MATLAB pode ser utilizado para converter G(z) em G(s), ou transformada Z inversa, quando G(s) não estiver em cascata com um reconstrutor de ordem zero (*zoh*). Primeiro criamos uma função de transferência discreta LTI. O comando *F = d2c(H,T,'zoh')* transforma H(z) em F(s) em cascata com um *zoh*, onde

*H(z) = ((z - 1)/z)*Z{F(s)/s}*.

Se considerarmos *F(s) = sG(s)*, o comando resolve para sG(s), dado H(z). Finalmente *sG(s)/s = G(s)* conduz ao resultado final. Em resumo, forme H(z), onde

*H(z)= ((z - 1)/z)*G(z)*.

Use *F = d2c(H,T,'zoh')* para obter *F(s) = sG(s)*. Divida o resultado por s e obtenha G(s). Exemplificando (observar que manipulações próximas da origem podem gerar resultados errados):

```
EDU» T=1;
EDU» numgz=[0.6321];
EDU» dengz=[1 -1.368 0.3679];
EDU» Gz=tf(numgz,dengz,T)
EDU» Hz=Gz*tf([1 -1],[1 0],T);
EDU» Hz=minreal(Hz);
EDU» Fs=d2c(Hz,'zoh');
EDU» Gs=Fs*tf(1,[1 0]);
EDU» Gs=minreal(Gs)
```

Podemos usar o MATLAB para determinar o ganho que assegura a estabilidade em um sistema de dados amostrados. Exemplificando:

```
EDU» T=1;
EDU» numgas=1;
EDU» dengas=[1 0];
EDU» Ga=tf(numgas,dengas)
EDU» Gz=c2d(Ga,T,'zoh')
EDU» for K=1:0.1:10;
Tz=feedback(K*Gz,1);  % Obtém T(z).
r=pole(Tz);      % Obtém os pólos para este valor de K.
rm=max(abs(r));    % Determina o pólo com o maior valor
                   % absoluto para este valor de K.
if rm>=1;          % Verifica se o pólo  está fora
                   % do círculo de raio unitário.
break;             % Interrompe se o pólo estiver fora
                   % do círculo de raio unitário.
end;               % Fim de if.
end;               % Fim de for.
K                  % Mostra o valor de K.
r                  % Mostra os pólos a malha fechada
                   % para este valor de K.
rm
```

3. Estabilidade de um sistema amostrado através do Root Locus no Plano Z :

Da mesma forma que nos sistemas contínuos, pode-se fazer o lugar das raízes para determinar o ganho que assegura estabilidade. Primeiro, cria-se um objeto função de transferência digital LTI para *FTMA(z) = N(z)/D(z)*, com um intervalo de amostragem não especicificado. A função de transferência de malha aberta é criada usando *tf(numdig,dendig,[])* ou *zpk(numdig, dendig,k,[])*, onde [] indica um intervalo de amostragem não especificado e os demais itens são similares aos comandos já vistos anteriormente.

O Matlab produz o lugar das raízes no plano z com um círculo de raio unitário superposto usando o comando *zgrid(z,wn)*, com z = wn = 0. Selecionamos, então, de forma interativa a interseção do lugar das raízes com o círculo de raio unitário. O Matlab responde com os valores do ganho e dos pólos a malha fechada. Exemplificando:

```
EDU» numdig=[0.7 -1];
EDU» dendig=[0 1 1];
EDU» k=1;
EDU» ftma=zpk(numdig,dendig,k,[])
EDU» rlocus(ftma)
EDU» zgrid(0,0)
EDU» [K,p]=rlocfind(ftma)
```

4. Estabilidade de um sistema amostrado através dos Diagramas de Bode no Plano V:

A análise da estabilidade de um sistema amostrado pode ser realizada através dos Diagramas de Bode, caso a expressão seja apresentada por variável do Plano V. A utilização do Matlab ocorre da mesma forma que em um sistema contínuo, utilizando a função de transferência de malha aberta. Cuidar para não esquecer que se está operando no plano V. Exemplificando:

```
EDU» numv=0.0234*[-1 1];
EDU» denv=[1 0 0];
EDU» ftma=tf(numv,denv)
EDU» bode(numv,denv)
```

5. Determinação da Resposta no Tempo a um degrau unitário em sistemas amostrados:

O software Matlab permite determinar a Resposta no Tempo a um degrau unitário de funções de transferência de malha fechada descritas por sistemas amostrados com o uso da transformada Z. A resposta ao degrau do sistema amostrado em malha fechada usa o comando *step(Tz)*, onde Tz é a FTMF (objeto função de transferência de malha fechada amostrada LTI).

Caso tenhamos uma expressão no domínio da variável complexa S:

$FTMA(s) = 1 / [s(s+1)]$

Adotando uma planta com reconstrutor de ordem zero (*hoz*) e tendo um período de amostragem T = 1s, ficamos com:

$FTMA(z) = (0,37z + 0,26) / (z^2 - 1,37z + 0,37)$

A expressão em malha fechada fica sendo:

$FTMF(z) = (0,37z + 0,26) / (z^2 - z +0,63)$

Exemplificando para o Matlab:

```
EDU» numz=[0.37 0.26];
EDU» denz=[1 -1.37 0.37];
EDU» ftmaz=tf(numz,denz,[])
EDU» ftmfz=feedback(ftmaz,1)
EDU» step(ftmfz)
```

Como se pode observar no gráfico gerado pelo reconstrutor de ordem zero, há uma aproximação razoável da resposta do sistema contínuo original.

Assim, pede-se para determinar a resposta no tempo a um degrau unitário das funções de transferência das plantas abaixo descritas, adotando sistemas amostrados com o uso da transformada Z. O período de amostragem adotado é T = 1s e o reconstrutor é de ordem zero. Exercitar o cálculo manualmente e após confirmar com o uso do Matlab.

a) $FTMA(s) = 50 (S+1) / [s^2 (s+9)]$
b) $FTMA(s) = 1 / [(s+1) (s^2+4s+5)]$

Observar o desempenho dinâmico das mesmas através do Tempo de Pico (tp), Overshoot (Mo) e Tempo de Estabilização (ts) pelo método das 4 constantes de tempo.