

UFMS - Universidade Federal de Mato Grosso do Sul

Facom - Faculdade de Computação

Curso: Engenharia de Computação

Data: 29/08/2024

Professor: Dr. Victor Leonardo Yoshimura

Disciplina: Controle e Servomecanismos

Laboratório 1

Scilab

Polinômios

Definamos, primeiramente, um Sistema Linear, Contínuo e Invariante no Tempo (SLIT-C) usando Funções Transferência (FTs). O comando `poly` será útil e sua sintaxe é:

`NomeDoPolinômio=poly(VetorDeCoeficientesOuRaízes, NomeDaVariável, String)`

- **NomeDoPolinômio**: Variável no ambiente de trabalho que irá armazenar o polinômio.
- **VetorDeCoeficientesOuRaízes**: Vetor que contém os coeficientes $[\alpha_0 \ \alpha_1 \ \dots \ \alpha_{n-1}]$ ou as raízes do polinômio.
- **NomeDaVariável**: Variável do polinômio. Ex.: x ou s .
- **String** (opcional): `'roots'` (default) ou `'coeff'`.

Exemplo 1 `G = poly([1 2 3], 's')` e `G = poly([1 2 3], 's', 'roots')` resultam o mesmo polinômio:

$$G = (s - 1) * (s - 2) * (s - 3) = -6 + 11s - 6s^2 + s^3$$

enquanto o comando `G = poly([1 2 3], 's', 'coeff')` gera o polinômio:

$$G = 1 + 2s + 3s^2$$

O comando `poly(0, 's')` é utilizado com muita frequência. Este comando pode ser substituído por `%s`.

Definindo um Sistema

Estamos em condições para definir um SLIT-C. Para tanto, será usado o comando `syslin`, cuja sintaxe é:

`NomeDoSistema = syslin(Domínio, DescriçãoDoSistema)`

- **NomeDoSistema**: Variável no ambiente de trabalho que irá armazenar o sistema.
- **Domínio**: String que determina o domínio de estudo: tempo contínuo (`'c'`) ou discreto (`'d'`).
- **DescriçãoDoSistema**: Descrição do sistema. Pode ser feita de três formas: espaço de estado, razão de polinômios ou Função Transferência.
 - Espaço de estado: Neste caso, **DescriçãoDoSistema** assumirá o formato `A,B,C,D,xo`, que são as matrizes do espaço de estado e o estado inicial. `D` (matriz nula por default) e `xo` (estado nulo por default) são opcionais;
 - Razão de polinômios: Neste caso, **DescriçãoDoSistema** assumirá o formato `N,D`, que são os polinômios numerador e denominador da FT do sistema;
 - Função Transferência: Neste caso, **DescriçãoDoSistema** assumirá o formato `FTMF`, que é a FT do sistema.

Exemplo 2 Definamos o SLIT-C, sob condições iniciais nulas:

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} -1 \\ 1 \end{bmatrix} \mathbf{u} \\ \mathbf{y} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}\end{aligned}$$

Para isto, usamos os comandos:

```
A = [1, 2; 3, 4]; B = [-1; 1]; C = [1, 0];  
sys = syslin('c', A, B, C)
```

O sistema ficará armazenado na variável `sys` no ambiente do Scilab.

Simulando um Sistema

Para fazer a simulação de um SLIT-C, usamos o comando `csim`, cuja sintaxe é:

```
[Saída, Estado] = csim(Entrada, InstantesDeAvaliação, SLIT-C, EstadoInicial, Tolerâncias)
```

- **Saída:** Variável que armazenará a saída do sistema;
- **Estado:** Variável que armazenará o estado do sistema (opcional);
- **Entrada:** Vetor da entrada do sistema, cujos valores estão determinados nos instantes de avaliação. Para entradas impulso e degrau unitário, pode-se utilizar, respectivamente, as strings `'impuls'` e `'step'`;
- **InstantesDeAvaliação:** Vetor com os instantes de tempo onde a entrada, a saída e o estado serão avaliados;
- **SLIT-C:** Sistema a ser simulado, não necessariamente gerado pela função `syslin`;
- **EstadoInicial:** Estado inicial, pode estar incluso no sistema, caso seja utilizada a função `syslin` (opcional, default nulo);
- **Tolerâncias:** Tolerâncias absoluta e relativa, utilizadas pelo solver de EDOs (opcional).

Exemplo 3 *A partir do exemplo anterior e ao executar os comandos:*

```
t = 0 : .01 : 10;  
y = csim('step', t, sys);
```

Qual o resultado esperado?

Resta plotar os resultados obtidos da simulação `csim`. O comando `plot` faz essa função e sua sintaxe é:

```
plot(Abscissa1, Ordenada1, Especificação1, ..., AbscissaN, OrdenadaN, EspecificaçãoN)
```

- **AbscissaI:** *i*-ésima abscissa;
- **OrdenadaI:** *i*-ésima ordenada;
- **EspecificaçãoI:** *i*-ésima especificação de plotagem (sólido, tracejado, cores, etc.).

É útil também o comando `xtitle`:

```
xtitle(TítuloDaFigura, TítuloDoEixoX, TítuloDoEixoY)
```

Octave¹

Definindo um Sistema

Em Octave, os comandos são diferentes para definir um SLIT-C como uma FT ou no espaço de estado. Para a primeira, o comando é `tf`, cuja sintaxe é:

```
NomeDoSistema = tf(CoeficientesDoNumerador, CoeficientesDoDenominador, TaxaDeAmostragem)
```

- **NomeDoSistema:** Variável no ambiente de trabalho que irá armazenar o sistema.
- **CoeficientesDoNumerador:** Vetor com coeficientes em ordem decrescente de grau.
- **CoeficientesDoDenominador:** Vetor com coeficientes em ordem decrescente de grau.
- **TaxaDeAmostragem:** Para sistemas discretos. Se for omitido, o Octave assume sistema contínuo.

Para representação no espaço de estado, o comando é `ss`, cuja sintaxe é:

```
NomeDoSistema = ss(A, B, C, D, TaxaDeAmostragem)
```

- **A, B, C, D:** Matrizes da representação no espaço de estado.
- **TaxaDeAmostragem:** Para sistemas discretos. Se for omitido, o Octave assume sistema contínuo.

Exemplo 4 *Declaremos o sistema do Exemplo 2 com o comando*

```
A = [1, 2; 3, 4]; B = [-1; 1]; C = [1, 0];  
sys = ss(A, B, C)
```

Verifique o que ocorre após executar o comando `tf(sys)`.

¹Execute o comando `pkg install -forge control` para instalar o pacote Control.

Simulando um Sistema

Para fazer a simulação de um SLIT-C ou SLIT-D, usamos o comando `lsim`, cuja sintaxe é:

```
[Saída, InstantesDeAvaliação, Estado] = lsim(SLIT, Entrada, InstantesDeAvaliação,  
                                             EstadoInicial)
```

- **Saída:** Variável que armazenará a saída do sistema;
- **InstantesDeAvaliação:** Vetor com os instantes de tempo onde a entrada, a saída e o estado serão avaliados;
- **Estado:** Variável que armazenará o estado do sistema (opcional);
- **SLIT:** Sistema a ser simulado: pode ser `tf` ou `ss`;
- **Entrada:** Vetor da entrada do sistema, cujos valores estão determinados nos instantes de avaliação;
- **EstadoInicial:** Estado inicial (opcional, default nulo).

Comandos específicos para simular entrada degrau, impulso e sistema sujeito apenas ao estado inicial são, respectivamente, `step`, `impulse` e `initial`, cujas sintaxes são

```
[Saída, InstantesDeAvaliação, Estado] = step(SLIT, InstantesDeAvaliação)  
[Saída, InstantesDeAvaliação, Estado] = impulse(SLIT, InstantesDeAvaliação)  
[Saída, InstantesDeAvaliação, Estado] = initial(SLIT, Estado Inicial, InstantesDeAvaliação)
```

Exemplo 5 *Simulemos o mesmo sistema usado no Scilab.*