Universidade Federal de Campina Grande
Centro de Ciências e Tecnologia
Departamento de Engenharia Elétrica
Laboratório de Controle Digital

Identificação de Sistemas

15 de dezembro de 2022

1 Identificação de Sistemas

A identificação de um modelo para o sistema a ser controlado é essencial para o projeto de controladores. Em alguns casos, parâmetros do modelo do sistema não são conhecidos e precisam ser estimados; em outros, não se tem informação nenhuma do modelo do sistema, seja por não se conhecer as suas propriedades físicas, ou pelo fato de se ter um modelo muito complexo. Assim, a identificação de sistemas surge como uma ferramenta importante para o projetista do sistema de controle e algumas das técnicas mais conhecidas para obtenção de modelos lineares a partir de experimentos serão apresentadas neste laboratório.

2 Modelagem a Partir da Resposta ao Degrau utilizando Método dos Mínimos Quadrados

Suponha que um processo esteja em **estado inicial nulo** e um degrau de amplitude h seja aplicado em $t = 0$ à sua entrada. Tanto a saída quanto a entrada do processo são armazenadas, **desde a aplicação do degrau até o tempo em que a saída entra em regime permanente**. Suponha que o modelo do processo é descrito pela função de transferência

$$G(s) = \frac{Y(s)}{U(s)} = \frac{G_o}{(T_1 s + 1)} e^{-Ls} \ . \tag{1}$$

Esse modelo é conhecido como modelo de Primeira Ordem com Atraso (FOPTD, da sigla em inglês *First Order Plus Time-Delay*). Para a relação 1, a saída $y(t)$ para uma entrada em degrau de amplitude h, é dada por

$$y(t) = hG_o\left(1 - e^{-\frac{t-L}{T_1}}\right) + \omega(t) \quad , \quad t \geq L,$$

onde $\omega(t)$ é o ruído branco presente na medição de $y(t)$. A equação anterior pode ser reescrita como

$$e^{-\frac{t-L}{T_1}} = 1 - \frac{y(t)}{hG_o} + \frac{\omega(t)}{hG_o} \quad , \quad t \geq L. \tag{2}$$

Integrando $y(t)$ de $t = 0$ até $t = \tau$ com $\tau \geq L$ ($y(t) = 0$ para $t < L$), tem-se

$$\int_0^\tau y(t)\, dt = hG_o\left(t + T_1 e^{-\frac{t-L}{T_1}}\right)\Big|_{t=L}^{t=\tau} + \int_0^\tau \omega(t)\, dt \ . \tag{3}$$

Usando a Eq. (2) e o fato de que $y(L) = 0$, tem-se então

$$\int_0^\tau y(t)\, dt = hG_o\left[\tau - L - T_1\frac{y(\tau)}{hG_o}\right] + \left[T_1\omega(t)\right]\Big|_{t=L}^{t=\tau} + \int_0^\tau \omega(t)\, dt \ . \tag{4}$$

Defina

$$A(\tau) = \int_0^\tau y(t)\, dt$$

e

$$\delta(\tau) = \left[T_1\omega(t)\right]\Big|_{t=L}^{t=\tau} + \int_0^\tau \omega(t)\, dt \ .$$

Então a Eq. (4) pode ser reescrita como

$$\begin{aligned} A(\tau) &= hG_o \left[\tau - L - T_1 \frac{y(\tau)}{hG_o}\right] + \delta(\tau) \\ &= \begin{bmatrix} h\tau & -h & -y(\tau) \end{bmatrix} \begin{bmatrix} G_o \\ G_o L \\ T_1 \end{bmatrix} + \delta(\tau) \end{aligned}$$

de modo que

$$A(\tau) = \phi^T(\tau)\,\theta + \delta(\tau) \ , \tag{5}$$

em que

$$\phi(\tau) = \begin{bmatrix} h\tau & -h & -y(\tau) \end{bmatrix}^T, \tag{6}$$

$$\theta = \begin{bmatrix} G_o & G_o L & T_1 \end{bmatrix}, \tag{7}$$

com o ruído dado pelo termo $\delta(\tau)$.

Assumindo que foram realizadas medições da entrada $u(t)$ e da saída $y(t)$ durante N períodos de amostragem, pode-se definir a matriz de dados

$$Z(N) = \begin{bmatrix} y(1) & y(2) & \cdots & y(N-1) & y(N) \\ u(1) & u(2) & \cdots & u(N-1) & u(N) \end{bmatrix}^T$$

e se obter uma estimação para os parâmetros do modelo (ou seja, uma estimação de θ). O objetivo da estimação de parâmetros é obter um vetor θ tal que o erro de predição $\varepsilon(t,\theta)$ seja o menor possível, de acordo com uma certa função de custo (ou critério de desempenho).

Defina a função de custo mínimos quadrados

$$V_N(\theta, Z(N)) = \frac{1}{N}\sum_{\tau=1}^{N}\left[A(\tau) - \phi^T(\tau)\,\theta\right]^2. \tag{8}$$

A estimativa para os parâmetros mínimos quadrados $\hat{\theta}^{LS}(N)$ é obtida de

$$\hat{\theta}^{LS}(N) = \arg\min_{\theta} V_N(\theta, Z(N)). \tag{9}$$

Como V_N é quadrática em θ, pode-se encontrar o valor mínimo derivando a equação acima e igualando o resultado a zero. Assim,

$$\frac{d}{d\theta}V_N(\theta, Z(N)) = \frac{2}{N}\sum_{\tau=1}^{N}\phi(\tau)\left(A(\tau) - \phi^T(\tau)\,\theta\right) = 0,$$

obtendo

$$\sum_{\tau=1}^{N}\phi(\tau)\,A(\tau) = \sum_{\tau=1}^{N}\phi(\tau)\,\phi^T(\tau)\,\theta. \tag{10}$$

Definindo

$$R_{n\times n} = \sum_{\tau=1}^{N}\phi(\tau)\,\phi^T(\tau)\,, \tag{11}$$

$$f_{n\times 1} = \sum_{\tau=1}^{N}\phi(\tau)\,A(\tau)\,, \tag{12}$$

tem-se que uma estimativa para θ pode ser calculada, assumindo-se que a inversa existe, por

$$\hat{\theta}(N) = \hat{\theta}^{LS}(N) = R^{-1}f, \tag{13}$$

a partir de que pode-se obter os parâmetros do modelo FOPTD (G_0, T_1 e L).

O algoritmo utilizado requer uma estimativa inicial do atraso L, mas este é recalculado por meio do segundo parâmetro de θ. **Assumiremos a estimativa inicial do atraso ($L_{inicial}$) sempre igual a um instante de amostragem ΔT e utilizaremos no modelo o valor de atraso recalculado por meio da equação 7**:

$$\frac{\theta(2)}{\theta(1)} = \frac{G_0 L_{estimado}}{G_0} = L_{estimado}.$$

3 Preparação

Prática 1 *Desenvolva uma função para a estimação de parâmetros a partir da resposta ao degrau. Nomeie a função criada* `parametrosFOPTD.m`.

Instruções:

*Para desenvolver a função, siga os seguintes passos (**sugestão de comandos no Matlab com descrições e exemplos de utilização em anexo no final deste guia**):*

1. *A função deve possuir o seguinte cabeçalho:*

   `function [G0, T1, L] = parametrosFOPTD(y,h,DeltaT)`

   onde as entradas são o vetor de saídas y, a amplitude do degrau h e o tempo de amostragem Δt. As saídas são os parâmetros do modelo FOPTD.

2. *Defina o tamanho do vetor de parâmetros, θ, a partir da expressão (7).*

3. *Crie matrizes R e f com valores nulos, nas dimensões: $R \rightarrow$ Matriz quadrada de ordem igual ao número de colunas de θ; $f \rightarrow$ Matriz coluna, com tamanho igual ao tamanho de θ .*

4. *Neste passo, deve-se criar um laço usando o comando* `for...end` *para implementar os somatórios das expressões 11 e 12. Lembre-se de que no Matlab os vetores possuem índices a partir do numeral 1 (não existe $y(0)$, $y(-1)$, etc.). Observe, portanto, o índice do vetor na hora de definir a variável de contagem do laço* `for...end`.

   Como se convencionou que a estimativa inicial de L, $L_{inicial}$, deve ser igual a um instante de amostragem, a função deve ser escrita levando isso em conta. Logo, a variável de controle do laço (`k`) pode ser iniciada em $k = 1$, de modo que o usuário não precisa selecionar o $L_{inicial}$ (ver Eq. (3)). Ex:

   ```
   for k = 1:N
   % ... Código do laço
   end
   ```

A cada iteração do laço:

(a) Crie uma variável Phi contendo o vetor $\phi(\tau)$, com **k** *a iteração atual, usando a expressão 6. Lembre-se que o tempo de amostragem não é unitário e, portanto, o valor de τ no primeiro elemento do vetor ($h\tau$) deve variar de acordo com o tempo de amostragem.*

(b) Calcule a matriz R como sendo:

$$R = R + \phi(\tau)\,\phi(\tau)^T$$

(c) Calcule $A(\tau)$, aproximando a integral por uma soma a cada iteração, i.e.,

$$A(\tau) = \int_0^\tau y(t)\,dt \cong \Delta t \sum_{n=0}^{\tau} y(n\Delta t) \ .$$

Use a função **sum** *do Matlab para calcular o somatório a cada iteração. Lembre-se de fornecer os limites no sinal na hora de utilizar a função e de levar em conta o valor do tempo de amostragem Δt. Ex:*

```
A = sum(y(1:k)) * Deltat;
```

(d) Calcule a matriz f como sendo:

$$f = f + \phi(\tau)\,A(\tau)$$

5. Ao sair do laço, crie uma variável teta que será o vetor de parâmetros obtidos a partir da expressão 13, i.e.,

$$\hat{\theta} = R^{-1} f$$

6. Extraia os parâmetros da FOPTD a partir da variável teta obtida no passo anterior.

Prática 2 *Teste a função criada. Para isso, execute o seguinte trecho de código no MATLAB:*

```
Greal = tf(2, [1 2 1], 'iodelay', 1);
h = 5;
DeltaT = 0.2;
t = 0:DeltaT:10;
u = h*ones(size(t));
randn('seed',314);
y = lsim(Greal, u, t) + 0.2*randn(length(t),1);
[G0, T1, L] = parametrosFOPTD(y,h,DeltaT)
Gid = tf(G0, [T1 1], 'iodelay', L);
yy = lsim(Gid, u, t);
plot(t,y,t,yy);
```

Como resultado, devem ser obtidos os valores: $G_0 = 2.0883$, $T_1 = 2.3434$ e $L = 1.1229$.

Prática 3 *Pesquise sobre o conceito de excitação persistente em identificação de sistemas (em inglês, persistent excitation). Observando a equação (6), qual seria o resultado do algoritmo de mínimos quadrados (eq. 13) caso a amplitude do degrau fosse nula ($h = 0$).*

4 Prática Experimental

Prática 4 *Construa no Simulink o diagrama do STH conforme fora feito no experimento de linearização e discretização:*

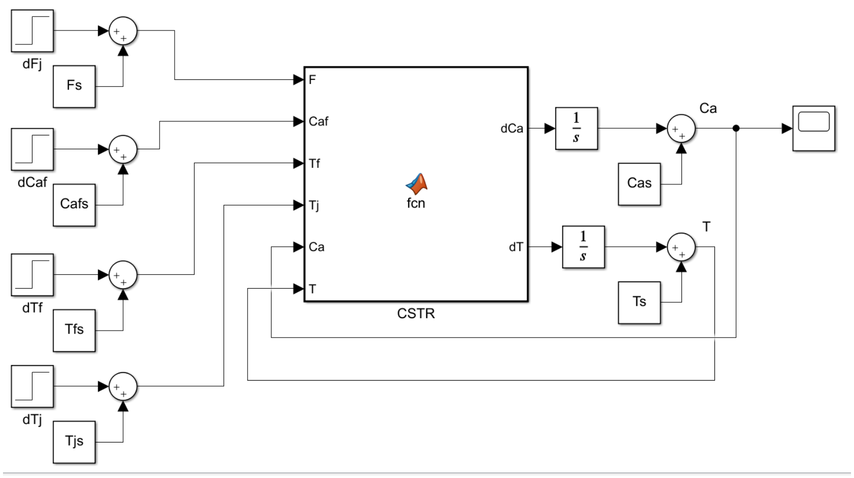

Figura 1: Diagrama de simulação do modelo não linear do CSTR

Os parâmetros de simulação são:

Parâmetros:	*Ponto de operação:*
$V = 1\ m^3$	$F_s = 1\ m^3/h$
$\Delta E = 11843\ J/mol$	$C_{Afs} = 10\ kgmol/m^3$
$\rho c_p = 500\ J/Km^3$	$T_{fs} = 298\ K$
$R = 1.987\ L/molK$	$T_{js} = 298\ K$
$UA = 150\ J/Kh$	$T_s = 311.2K$
$k_0 = 9703 \times 3600\ h^{-1}$	$C_{As} = 8.564\ kgmol/m^3$
$\Delta H = -5960\ J/mol$	

No Simulink, navegue para $Modeling \rightarrow ModelSettings$ *e altere o tempo de simulação para* 15 h*, o solver para passo fixo, e o passo de simulação para* 0.01 h*, conforme destacado na figura 2.*

Nesse experimento, o objetivo é identificar as funções de transferência entre a temperatura e cada uma das entradas:

- $G_{11}(s)$ – *saída* C_A*, entrada* F
- $G_{12}(s)$ – *saída* C_A*, perturbação* C_{Af}
- $G_{13}(s)$ – *saída* C_A*, perturbação* T_f
- $G_{14}(s)$ – *saída* C_A*, perturbação* T_j

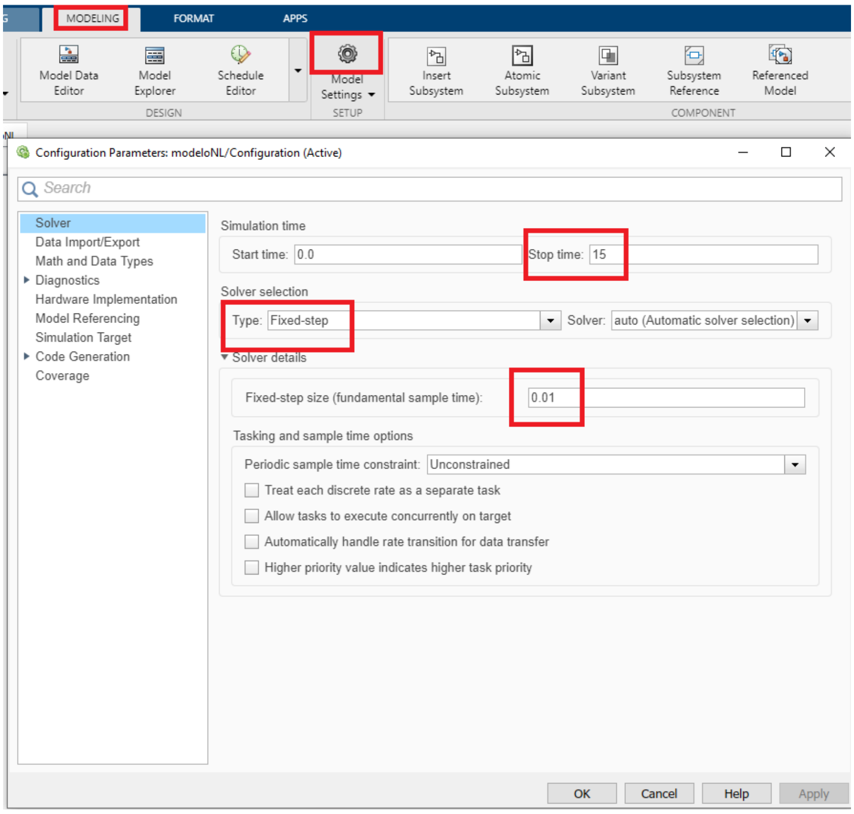

Figura 2: Configurações da simulação.

Na identificação de cada função, é necessária a realização de um experimento do degrau, variando uma entrada por vez, enquanto as demais são mantidas em zero. Para cada experimento, utilize as seguintes amplitudes do degrau:

1. $dF = 0.01$, $dC_{Af} = 0$, $dT_f = 0$ *e* $dT_j = 0$
2. $dF = 0$, $dC_{Af} = 0.007$, $dT_f = 0$ *e* $dT_j = 0$
3. $dF = 0$, $dC_{Af} = 0$, $dT_f = 1$ *e* $dT_j = 0$
4. $dF = 0$, $dC_{Af} = 0$, $dT_f = 0$ *e* $dT_j = 1$

Para cada experimento, utilize a função `parametrosFOPTD.m` *desenvolvida na prática 1 para identificar cada função de transferência descrita pelos dados gerados.*

Com as funções de transferência obtidas, obtenha a resposta ao degrau para cada caso, utilizando os mesmos valores de amplitude do experimento.

Compare a resposta ao degrau dos modelos identificados com os dados experimentais. Calcule o erro médio ao quadrado em cada caso.

*Observações (**Importante!**)*

1. *os dados obtidos no simulink devem ser exportados para o workspace antes de serem chamados pela função. No experimento 1 foi apresentado o bloco To Workspace que executa essa funcionalidade.*
2. *ao realizar a identificação dos dados em cada experimento, remova qualquer condição inicial dos vetores (ex.: Ca = Ca - Ca(1))*

Comandos e Funções do Matlab

- **Função INV :**

inv(X) é a inversa da matriz quadrada (ou número escalar) X. Ex.:

```
>> inv([1 2 ; 3 4])

ans =
   -2.0000    1.0000
    1.5000   -0.5000
```

- **Operador :**

X \Y é a inversa da matriz quadrada (ou número escalar) X multiplicada pela matriz ou vetor Y de dimensões correspondentes. Ex.:

```
>> ([1 2 ; 3 4])\[5 ; 6]

ans =

   -4.0000
    4.5000
```

- **Função ZEROS :**

zeros(N) é uma matriz de zeros $N \times N$.
zeros(M,N) ou *zeros*([M,N]) é uma matriz de zeros $M \times N$.
zeros(*size*(A)) é da mesma dimensão de A e completa de zeros.
Ex.:

```
>> R = zeros(2,4)

R =
     0     0     0     0
     0     0     0     0

>> R = zeros(3)


R =
     0     0     0
     0     0     0
     0     0     0
```

- **Função SUM :**

Para vetores, *sum*(X) é a soma dos elementos de X. Para matrizes, *sum*(X) é um vetor linha com a soma dos elementos de cada coluna de X. Ex.:

```
>> sum([1 2 3 4 5])

ans =

    15

>> X = [1 2 ; 3 4 ]

X =
     1     2
     3     4

>> sum(X)

ans =
     4     6
```

- **Laço FOR :**

for repete declarações durante um número específico de vezes. A forma geral de uma declaração *for* é

```
for variable = expr,
     statement 1;
     statement n;
end
```

Exemplo de um somatório do vetor y usando um laço FOR:

```
R = 0;
for k = 1:length(y),
     R = R + y(k);
end
```

- **Função D2C :**

[NUM, DEN] = d2c(NUMD,DEND,T) converte a função de transferência discreta NUMD(z)/DEND(z), cujo período de amostragem é T, para a função de transferência contínua NUM(s)/DEN(s).

- **Função C2D :**

[NUMD, DEND] = c2d(NUM,DEN,T) converte a função de transferência contínua NUM(s)/DEN(s) para a função de transferência discreta NUMD(z)/DEND(z), usando o período de amostragem T.