UFC/CT/DETI

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

SINAIS E SISTEMAS

3a. Avaliação

Estimação de Parâmetros de Modelo ARX via Mínimos Quadrados

2025

1 – INTRODUÇÃO

Na Engenharia, estamos frequentemente interessados no desenvolvimento de um modelo matemático de um fenômeno físico qualquer a fim de se fazer uma previsão analítica sobre o comportamento do referido fenômeno. Nas aplicações de automação, é de grande importância a modelagem da planta física na qual se deseja controlar para a prever o efeito do esforço de controle e de perturbações na planta. Pela planta, entende-se como qualquer processo caracterizado por certo número de entradas *u(t)* e de saídas *y(t)*.

Em geral, o modelo da planta pode ser obtido pela aplicação dos princípios físicos via as equações dinâmicas governantes. Às vezes, o modelo não pode ser obtido de argumentos físicos por causa, por exemplo, da complexidade dos processos físicos. Neste caso, recorrer-se-á ao método experimental para se obter o modelo da planta. Ao mesmo, dá-se o nome de *Identificação ou Estimação de parâmetros*. Essa abordagem permite um tratamento genérico na modelagem de sistemas de natureza variada, como por exemplo: robôs móveis, processos biotecnológicos e canais de comunicação, para citar alguns sistemas.

A técnica utilizada neste experimento emprega um sinal determinístico de entrada degrau *u(t)* e um modelo ARX (autoregressivo e entrada exógena), para a estimação dos parâmetros do modelo. Estas equações constituem o modelo matemático da planta.

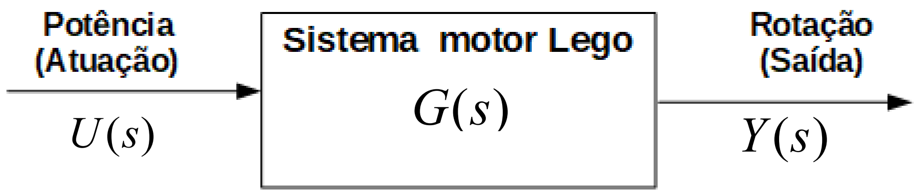
Observação: Para melhor compreensão da modelagem consultar as notas de aula.

2 – APLICAÇÃO DA METODOLOGIA DE ESTIMAÇÃO OFF-LINE

Como aplicação da metodologia descrita, será considerado a estimação de parâmetros de um sistema Motor do Lego em que os parâmetros do processo são invariantes no tempo. Para a análise e obtenção do modelo poderá ser utilizado o OCTAVE ou MATLAB.

2.1 – Sistema Motor do Lego

Um ponto de partida razoável para a modelagem é supor que o sistema de motor Lego possa ser modelado por um sistema dinâmico de segunda ordem. **Por que ?** Vide o resultado do ensaio.



- *G(s)* - representa a F.T. entre o sinal de saída e de entrada do sistema
- *Ys)* - sinal de saída do sistema (rotação)
- *U(s)* - entrada do sistema (potência).

Para sistema de motor Lego, considerar os seguintes modelos representados pelas equações à diferenças:

Modelo de Primeira Ordem

$$y(k) = a_1 y(k-1) + b_1 u(k-1)$$

Modelo de Segunda Ordem

$$\boxed{y(k) = a_1 y(k\text{-}1) + a_0 y(k\text{-}2) + b_1 u(k\text{-}1) + b_0 u(k\text{-}2)}$$

onde os valores de entrada e saída são dados experimentais fornecidos no problema e os valores dos parâmetros a_1, a_0, b_1, b_0, serão estimados. Vide anexo 2 para um rápido entendimento sobre como função de transferência *G(z)= Y(z)/U(z) é* "vista" pelo computador.

A sequência de entrada (controle *u(k)*) aplicada é uma sequência degrau a partir de *k = 1*. Tanto a sequência de entrada *u(k)* como a sequência de resposta *y(k)* gerada estão no arquivo GrupoRoboA.mat (a letra A indica a equipe). Este arquivo contem o vetor *z* com a primeira coluna formada pelos valores de *y(k)*, e a segunda coluna formada pelos valores de *u(k)*, para *k* inteiro. O tempo de amostragem é *T=0,01* segundos.

Atividades

Atividade 1

1) Escrever um programa utilizando os ambientes de simulação mencionados para a montagem da equação normal e estimação dos parâmetros para os modelos de primeira e de segunda ordem. Ver formato do código no anexo 1.

2) Gerar um gráfico com os valores dos valores de *y(k)* medidos e estimados pelos modelos de primeira e de segunda ordem.

3) Verificar o critério de validação dos modelos de primeira e de segunda ordem.

4) Obter a função de transferência *G(z).* Para obter *G(s)* utilizar o comando *d2c* no OCTAVE ou MATLAB.

5) Comparar a resposta ao degrau do tempo discreto e com a resposta do tempo contínuo associado sistema *G(s)* para o modelo de segunda ordem.

Atividade 2

Estudar como as características de frequência de um sistema de tempo discreto dependem do tempo de amostragem.

Considerar o sistema de tempo discreto definido pela seguinte equação a diferenças:

$$y(k) = 1{,}238y(k\text{-}1) - 0{,}3016y(k) + 0{,}03175x(k\text{-}2) - 0{,}03175x(k)$$

Pede-se:

a) Calcular a função de transferência, $G(z)$, do sistema de tempo discreto.

b) Transformar $G(z)$ em um sistema de tempo contínuo $G(s)$ equivalente usando a aproximação bilinear (*Tustin*). **Observação 1:** $G(s)$ resultante será uma função do tempo de amostragem T, que pode não ser conhecido. **Observação 2:** Compreender o comportamento da frequência em um sistema discreto não é tão intuitivo como no caso do tempo contínuo.

c) Obter as funções de transferência correspondentes $G_1(s)=G(s)$ e $G_2(s)=G(s)$ para $T = 0{,}1$ e $T = 1$, respectivamente.

d) Comparar a resposta a sequência degrau unitário do sistema $G(z)$ e com as respostas ao degrau unitário dos sistemas associados $G_1(s)$ e $G_2(s)$, quando $T = 0{,}1$ e $T = 1$. (Usar o comando *step* no OCTAVE ou MATLAB). Comentar suas conclusões a respeito de como o tempo de amostragem determina a resposta do sistema discreto.

Anexo 1

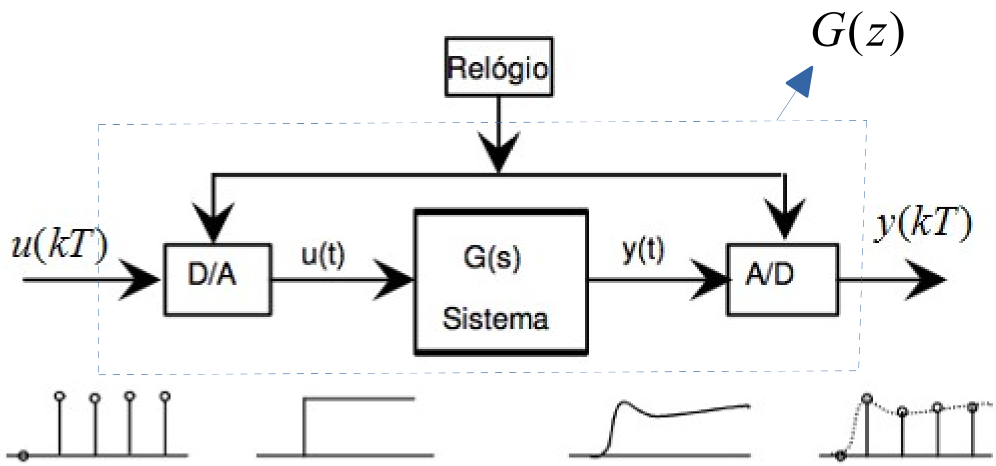
```
close all
clear all
clc

fprintf('\nA Equação a diferença de primeira ordem : y(k) = a1y(k-1) +
b1u(k-1)\n')
fprintf('\nA Equação a diferença de segunda ordem  : y(k) = a1y(k-1) +
a0y(k-2) + b1u(k-1) + b0u(k-2)\n')
fprintf('Será utilizada para a formulação das matrizes e estimação dos
parâmetros a1, a0, b1 e b0.\n')

load GrupoRobo_A.mat;  % baixar os dados do grupo A

l = length(z1); % comprimento do vetor z1
y = z1(:,1);    % valores medidos de y
u = z1(:,2);    % entrada de sinal


%$$$$$$$$$$$$$$$$$$$$$$ Modelo de 1a. Ordem $$$$$$$$$$$$$$$$$$$$$$$$$

%                    y(k) = a1y(k-1) + b1u(k-1)

% Escrever o código para o modelo de primeira ordem

theta = [a1 b1];      % Obtenção dos parâmetros a1,b1

y_est1;               % Obtenção do sinal estimado
erro = y - y_est;     % erro de estimação
Jest = norm(erro)/l;  % MSE


fprintf('Os parâmetos estimados foram:\n\n'), theta
fprintf('A equação a diferenças é:\n\ny(k) = %.4fy(k-1)  + %.4fu(k-
1) )\n', theta(1), theta(2))

fprintf('\nO aproximação J é: \n\n%d\n\n', Jest)

fprintf('-------------------------------------------------------------
------------------------------------------------------------\n')

fprintf('A função em Z é:\n\nY(z) = %.4fY(z)z^-1 + %.4fU(z)z^-1 \n',
theta(1), theta(2))

fprintf('A função de transferência em z, G(z) é:\n\n')

fprintf('Y(z)/U(z) = (%.4fz^-1 )\n/(1 - %.4fz^-1 )',  theta(1),
abs(theta(2)))

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

%$$$$$$$$$$$$$$$$$$$$$$$ Modelo de segunda ordem $$$$$$$$$$$$$$$$$$$$$


%             y(k) = a1y(k-1) + a0y(k-2) + b1u(k-1) + b0u(k-2)

% Escrever o código para o modelo de segunda ordem
```

```
theta = [a1 a0 b1 b0];  % Obtenção dos parâmetros a1, a0, b1, b0

y_est2;                 % Obtenção do sinal estimado
erro = y - y_est;       % erro de estimação
Jest = norm(erro)/l;    % MSE

fprintf('Os parâmetos estimados foram:\n\n'),theta

fprintf('A equação a diferenças é:\n\ny(k) = %.4fy(k-1) + %.4fy(k-2) +
%.4fu(k-1) + %.4fu(k-2)\n', theta(1), theta(2), theta(3), theta(4))

fprintf('\nO aproximação Jest é: \n\n%d\n\n', Jest)

fprintf('-------------------------------------------------------------
---------------------------------------------------------------\n')

fprintf('A função em Z é:\n\nY(z) = %.4fY(z)z^-1 + %.4fY(z)z^-2 +
%.4fU(z)z^-1 + %.4fU(z)z^-2\n', theta(1), theta(2), theta(3),
theta(4))

fprintf('A função de transferência em z, G(z) é:\n\n')

fprintf('Y(z)/U(z) = (%.4fz^-1 + %.4fz^-2)\n/(1 - %.4fz^-1 + %.4fz^-
2)', theta(3), theta(4), theta(1), abs(theta(2)))

%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$Fim Modelo
de Segunda Ordem$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

% Gerar gráficos
figure (1)
plot(y)
hold on
plot(y_est1)   % Modelo de primeira ordem
plot(y1_est2)  % Modelo de segunda ordem
legend({'y_m_e_d_i_d_o','y_e_s_t_2_a_._o_r_d_e_m','y_e_s_t_1_a_._o_r_d
_e_m'},'Location','southeast')
```

Anexo 2

Conceito Simplificado

Se aplicarmos uma sequência degrau *u(kT)* à entrada do sistema contínuo aparecerá também um degrau *u(t)* na entrada da planta conforme mostrado nas figuras abaixo.



A função de transferência *G(z)* "vista" pelo computador é ilustrada.