Métodos paramétricos de identificação de sistemas - Trabalho 4

Tassiano Neuhaus
Universidade Federal do Rio Grande do Sul - Departamento de Engenharia Elétrica
Av. Osvaldo Aranha, 103 - Bairro Bom Fim CEP: 90035-190 - Porto Alegre - RS - Brasil

***Resumo*—Trabalho 4 - Questões**

***Palavras-chave*—Identificação de sistemas lineares, métodos paramétricos.**

I. Questão 1

Questão: verificar de o LSE (estimador de mínimos quadrados) produz estimativas não polarizadas no caso de estrituras do tipo ARX ARMAX e OE.

Genericamente, modelos utilizados para identificação de sistemas podem ser representados pela equação (1).

$$A(z,\theta)Y(t) = \frac{B(z,\theta)}{F(z,\theta)}U(t) + \frac{C(z,\theta)}{D(z,\theta)}e(t) \qquad (1)$$

Onde:

$$\begin{aligned} A(z,\theta) &= 1 + a_1 Z^{-1} + a_2 Z^{-2} + \cdots + a_{na} Z^{-na} \\ B(z,\theta) &= b_1 Z^{-1} + b_2 Z^{-2} + \cdots + b_{nb} Z^{-nb} \\ C(z,\theta) &= 1 + c_1 Z^{-1} + c_2 Z^{-2} + \cdots + c_{nc} Z^{-nc} \\ D(z,\theta) &= 1 + d_1 Z^{-1} + d_2 Z^{-2} + \cdots + d_{na} Z^{-nd} \\ F(z,\theta) &= 1 + f_1 Z^{-1} + f_2 Z^{-2} + \cdots + f_{nf} Z^{-nf} \end{aligned}$$

Baseado nestas informações existem modelos onde apenas alguns destes polinómios são diferentes de 1. Na Tabela (I) são apresentados alguns destes modelos mais comumente utilizados.

Tabela I
Modelos comumente utilizados para identificação de sistemas

Modelo	Polinómios diferentes de 1
FIR	B
ARX	A B
ARMAX	A B C
ARMA	A C
ARARMAX	A B C D
OE	B F
BJ	B F C D

A. ARX

A partir de (1) tem-se que o modelo ARX fica (2).

$$A(z,\theta)Y(t) = B(z,\theta)U(t) + e(t) \qquad (2)$$

A equação (2) pode ser reescrita :

$$\begin{gathered} Y(t) = b_1 Z^{-1}U(t) + b_2 Z^{-2}U(t) + \cdots + b_{nb} Z^{-nb}U(t) \\ -a_1 Z^{-1}Y(t) - b_2 Z^{-2}Y(t) - \cdots - a_{na} Z^{-na}Y(t) + e(t) \end{gathered}$$

O que pode ser escrito como em (3).

$$Y(t) = \varphi'\theta + e(t) \qquad (3)$$

Onde:

$$\theta = \begin{bmatrix} a_1 \\ \vdots \\ a_{na} \\ b_1 \\ \vdots \\ b_{nb} \end{bmatrix} \qquad \varphi(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-na) \\ u(t-1) \\ \vdots \\ u(t-nb) \end{bmatrix}$$

$$\Phi = \begin{bmatrix} \varphi'(1) \\ \varphi'(2) \\ \vdots \\ \varphi'(N) \end{bmatrix}$$

Pode -se então encontrar a estimativa de θ, ou seja $\hat{\theta}$.

$$\begin{gathered} \hat{\theta} = (\Phi'\Phi)^{-1}(\Phi'(\Phi\theta_0 + e)) = \theta_0 + (\Phi'\Phi)^{-1}\Phi' e \\ E\{\hat{\theta}\} = \theta_0 + (\Phi'\Phi)^{-1}\Phi' E\{e(t)\} \end{gathered}$$

Mas como $E\{e(t)\} = 0$, pois o ruido possui média zero. A estimativa de $\hat{\theta}$ é o próprio θ.

De forma semelhante pode-se escrever:

$$\begin{gathered} E\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)'\} = E\{((\Phi'\Phi)^{-1}\Phi' e)((\Phi'\Phi)^{-1}\Phi' e)'\} \\ = (\Phi'\Phi)^{-1}\Phi' E\{e\ e'\}\Phi(\Phi'\Phi)^{-1} \\ = E\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)'\} = conv(\hat{\theta}) = \lambda^2(\Phi'\Phi)^{-1} \\ pois : E\{e\ e'\} = \lambda^2 I \end{gathered}$$

Já que a convolução da estimativa $\hat{\theta}$ corresponde ao valor encontrado, o erro de polarização para o sistema ARX é igual a zero.

B. ARMAX

A partir de (1) tem-se que o modelo ARMAX fica (4).

$$A(z)Y(t) = B(z,\theta)U(t) + C(z,\theta)e(t) \qquad (4)$$

II. Questão 2

Questão: Seja o sistema ARX (5):

$$G_o(z) = \frac{2}{z - 0.8} \quad H_o(z) = \frac{z}{z - 0.8} \tag{5}$$

E com ruido branco com $\lambda^2 = 0.1$.

- Realize uma simulação aplicando na entrada um ruido branco com $\lambda^2 = 1$.
- Plote 100 estimativas de $\hat{\theta}$, a elipse de 95% de confiança e verifique o valor médio obtido e avalie a polarização da estimativa.
- Repita o item anterior com $H(z) = 1$.

A. *Item 1 e 2*

Para realização desta simulação foi utilizado o script apresentado no Anexo 1.

Utilizando um ruido branco como entrada, com média zero, e $\lambda^2 = 1$, obtém-se os resultados apresentados na Figura (2).

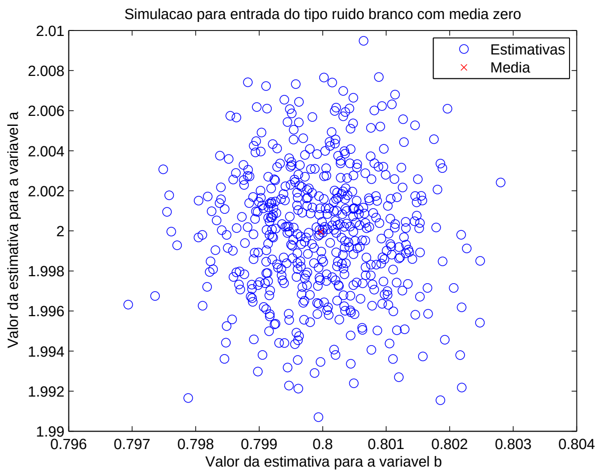

Figura 1. Entrada aleatória aplicada no processo para a identificação do sistema.

Observa-se que as estimativas em média chegam relativamente próximas ao valor real (a=2, b=0.8). Desta forma conclui-se que não há erro de polarização. Quando o ruido inserido não possui média zero, há a observação de erro de polarização, como é apresentado na Figura (**??**).

Neste caso, os valores médios encontrados foram b=0.8129 e a=2.003 e no caso onde a média é zero, os valore estimados médios foram de a=1.9999 e b=0.8000.

B. *Item 3*

Na figura (3) observa-se a simulação para o mesmo sistema do item anterior, mas com o ruido branco sujeito a função de transferência $H(z) = 1$. Observa-se que a acuracidade da média dos pontos não é a mesma que quando a função de transferência $H(z)$ é como em (5.

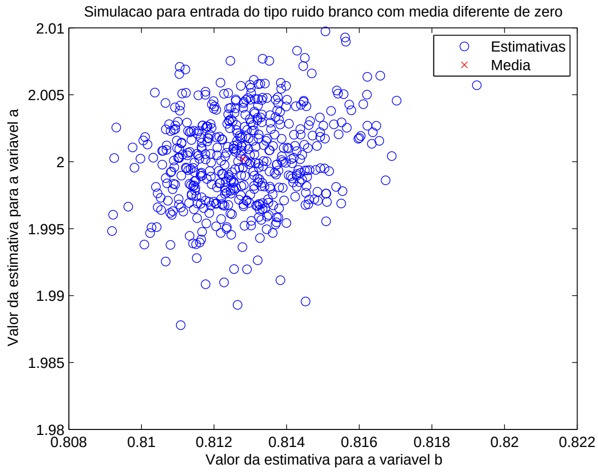

Figura 2. Entrada aleatória aplicada no processo para a identificação do sistema. Média diferente de zero.

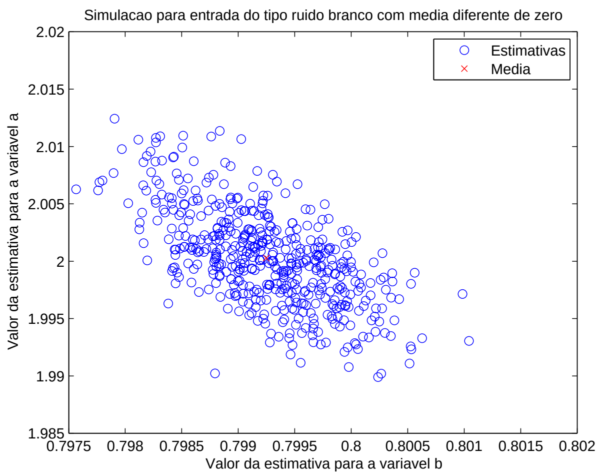

Figura 3. Entrada aleatória aplicada no processo para a identificação do sistema. Ruido sujeito a $H(z) = 1$.

Apêndice

1 - Script para estimar parâmetros da função de transferência

```
%===================================
% Identificacao de sistemas
% Tassiano Neuhaus
% tassianors@gmail.com
% UFRGS
%===================================
close all; clear all;

% Definitions
Ts=10e-3;
Tf=10;
STD=0.1;
tempo = 0:Ts:Tf;
N=size(tempo, 2);

M=500;

% TFs
G=tf([2],[1 -0.8], Ts);
H=tf([1 0],[1 -0.8], Ts);
```

```
% Replace the default stream with a stream whose
    seed is based on CLOCK, so
% RAND will return different values in different
    MATLAB sessions
RandStream.setDefaultStream(
    RandStream('mt19937ar', 'seed',
    sum(100*clock)));

% identification using MMQ
% model y(t)=2*u(t-1)+0.8*y(t-1) +u(t) +0.8*y(t-1)
teta=[0.8; 1; 2];
n=size(teta, 1);
% e entrada u saida do controlador
%phy=[y(t-1); u(t); u(t-1)]

% numero de vezes que sera aplicado o metodo.
a=zeros(M,1);
b=zeros(M,1);
err_teta=zeros(M,2);
for j=1:M
    % make a randon noise with std = 0.1
    ran=rand(N, 1);
    s=std(ran);
    % now ran_s has std=1;
    ran_s=ran/s;
    m=mean(ran_s);
    % make noise be zero mean
    rh=(ran_s-m)*STD;

    % make a randon noise with std = 0.1
    ran=rand(N, 1);
    s=std(ran);
    m=mean(ran);
    % now rr has std=1;
    rr=(ran-m)/s;

    yr=lsim(G, rr, tempo);
    ynoise=lsim(H, rh, tempo);
    y=yr+ynoise;
    u=rr;

    phy=zeros(N, n);
    for t=2:N
        phy(t, 1)=y(t-1);
        phy(t, 2)=u(t);
        phy(t, 3)=u(t-1);
    end

    % make sure, rank(phy) = n :)
    teta_r=inv(phy'*phy)*phy'*y;
    err_teta(j, 1)=teta_r(1)-0.8;
    err_teta(j, 2)=teta_r(3)-2;
    % to be used in grafic plot
    a(j)=teta_r(1);
    b(j)=teta_r(3);
end
PN=err_teta'*err_teta;
ma=mean(a)
sa=std(a);
mb=mean(b)
sb=std(b);
plot(a, b, 'bo');
hold;
plot(ma, mb, 'ro');
hold;
circle = rsmak('circle');
fnplt(circle);
ellipse = fncmb(circle,PN);
```

Listing 1.   Descriptive Caption Text