Identificação de sistemas lineares - Trabalho 5

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*—Identificar um modelo ARX para um processo que não pode ser representado por este modelo e também um modelo completo. Ao fim, será feito um comparativo qualitativo das duas estimativas obtidas.**

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

I. Introdução

Neste trabalho será apresentado um modelo ARX para um sistema que não pode ser completamente representado por este modelo (1). Em seguida será utilizado um modelo mais completo e será feito um comparativo qualitativo das estimativas obtidas para o modelo utilizando cada um dos métodos.

$$G_0(q) = \frac{2}{q-0.8} \qquad H_0(q) = \frac{q+0.9}{q-0.5} \tag{1}$$

II. Modelo ARX

O sistema real apresentado em (1) será identificado pelo modelo ARX onde genericamente o modelo utilizado é como apresentado em (2) e para o modelo ARX tem-se que apenas os polinômios A e B são diferentes de 1. [1]

$$A(q,\theta)Y(t) = \frac{B(q,\theta)}{F(q,\theta)}U(t) + \frac{C(q,\theta)}{D(q,\theta)}e(t) \tag{2}$$

Onde:

$$\begin{aligned} A(q,\theta) &= 1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_{na}q^{-na} \\ B(q,\theta) &= b_1q^{-1} + b_2q^{-2} + \cdots + b_{nb}q^{-nb} \\ C(q,\theta) &= 1 + c_1q^{-1} + c_2q^{-2} + \cdots + c_{nc}q^{-nc} \\ D(q,\theta) &= 1 + d_1q^{-1} + d_2q^{-2} + \cdots + d_{na}q^{-nd} \\ F(q,\theta) &= 1 + f_1q^{-1} + f_2q^{-2} + \cdots + f_{nf}q^{-nf} \end{aligned}$$

Desta forma o modelo ARX pode ser representado como em (**??**). Para o sistema apresentado em (1), o modelo ARX fica como em (4).

$$A(q,\theta)Y(t) = B(q,\theta)U(t) + e(t) \tag{3}$$

$$G(q,\theta) = \frac{a}{q-b} \qquad H(q,\theta) = \frac{q}{q-b} \tag{4}$$

$$y(t) = G(q,\theta)r(t) + H(q,\theta)e(t)$$

Onde $e(t)$ é ruido branco com média zero.

Este modelo não consegue representar o sistema descrito em (1). Foi utilizado o script do matlab apresentado no Anexo (A) para simular as estimativas obtidas para os parâmetros a e b deste modelo, o script utiliza o método dos mínimos quadrados para estimar os parâmetros.

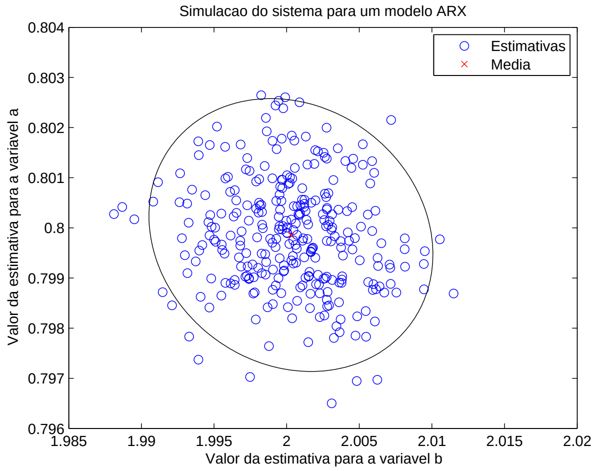

Figura 1. Simulação do sistema para uma entrada aleatória e utilizando o modelo ARX.

O resultado da simulação é apresentado na Figura (1).

A média das estimativas obtidas para o sistema foi de $a = 2.003$ e $b = 0.7999$.

Aplicando na entrada do processo uma senoide de frequência $\pi/4$ obtém-se a estimativa como apresentado na Figura (2).

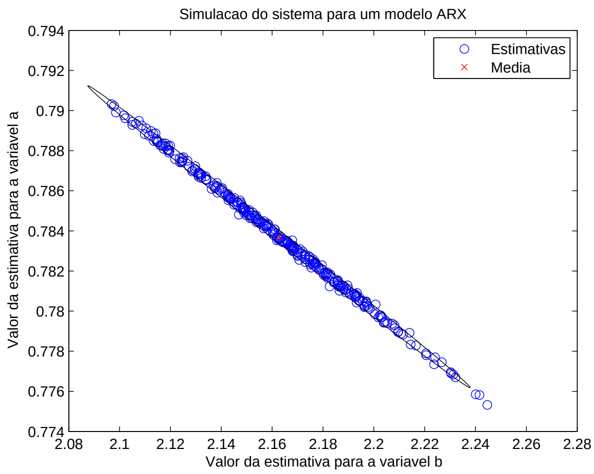

Figura 2. Simulação do sistema para uma entrada $sin(\pi/4)$ e utilizando o modelo ARX.

A média das estimativas obtidas para o sistema foi de $a = 2.1627$ e $b = 0.7837$.

Aplicando na entrada do processo uma senoide de frequência $\pi/20$ obtém-se a estimativa como apresentado na Figura (3).

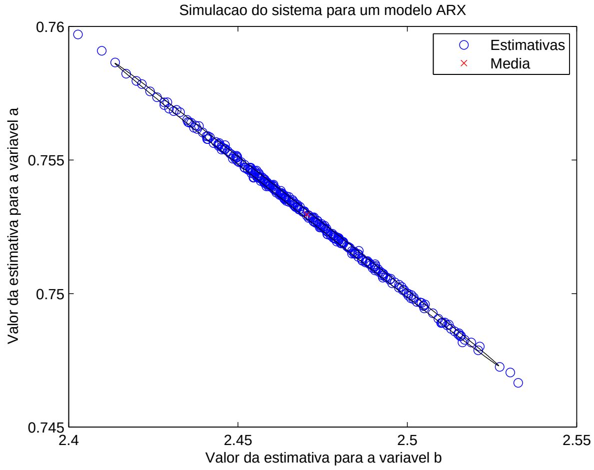

Figura 3. Simulação do sistema para uma entrada $sin(\pi/20)$ e utilizando o modelo ARX.

A média das estimativas obtidas para o sistema foi de $a = 2.1687$ e $b = 0.7831$.

Observa-se claramente que a estimativa está polarizada, ou seja, a média das estimativas não está centrada nos valores reais dos parâmetros. Isso de deve ao fato que o modelo utilizado para a estimativa não consegue representar na totalidade o sistema original.

III. Modelo Completo

Como apresentado na seção (II) o modelo ARX não consegue representar o sistema (1) completamente, e a estimativa dos parâmetros da Função de transferência são polarizados. Para contornar este problema utilizaremos um modelo para descrever o sistema (1) de forma completa. [2]

O modelo escolhido para representar o sistema real é apresentado em (5).

$$G(q,\theta) = \frac{a}{q-b} \qquad H(q,\theta) = \frac{q-c}{q-d} \tag{5}$$

Utilizando o estimador ótimo (6) obtém-se a equação de diferenças apresentada em (7). Utilizando o *script* A obtém-se o resultado para os parâmetros a, b, c e d para a função $G(q,\theta)$ e $H(q,\theta)$.

$$\hat{y}(t/t-1,\theta) = H^{-1}(q,\theta)G(q,\theta)u(t) + [1 - H^{-1}(q,\theta)]y(t) \tag{6}$$

$$\begin{aligned}\hat{y}(t/t-1,\theta) = \quad & a\,u(t-1) - ad\,u(t-2) + (d-c)y(t-1) \\ & -b(d-c)y(t-2) + (b+c)\hat{y}(t-1/t-2,\theta) \\ & -cb\,\hat{y}(t-2/t-3,\theta)\end{aligned} \tag{7}$$

Desta forma temos para o método dos mínimos quadrados as seguintes equiões:

$$\Theta = \begin{bmatrix} a \\ a\,d \\ (d-c) \\ b(d-c) \\ (b+c) \\ c\,b \end{bmatrix}$$

$$\varphi(t) = \begin{bmatrix} u(t-1) \\ -u(t-2) \\ y(t-1) \\ -y(t-2) \\ \hat{y}(t-1/t-2,\theta) \\ -\hat{y}(t-2/t-3,\theta) \end{bmatrix}^T$$

Nas Figuras (4) e (5) apresentam-se os valores estimados para os parâmetros das funções de transferência $G(q)$ e $H(q)$ respectivamente.

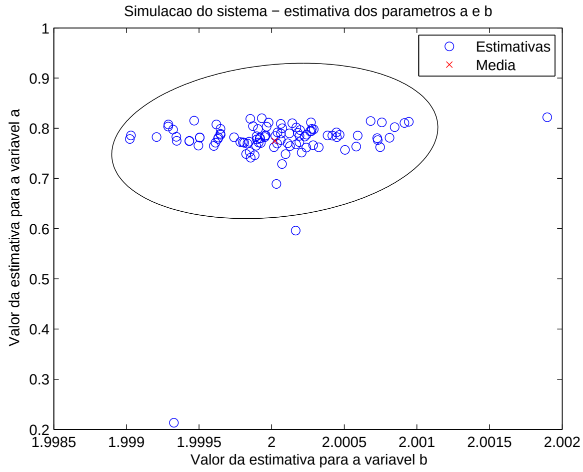

Figura 4. Simulação do sistema para uma entrada aleatória e utilizando o modelo completo - variáveis do processo $G(q)$ a e b.

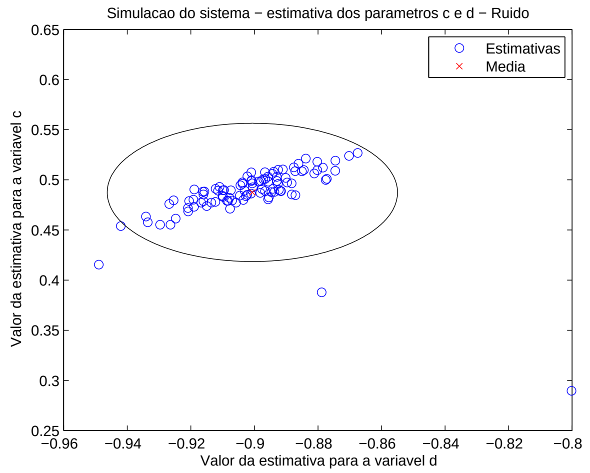

Figura 5. Simulação do sistema para uma entrada aleatória e utilizando o modelo completo - variáveis do ruido $H(q)$ c e d.

A média das estimativas para os parâmetros em questão é apresentado na Tabela (I).

Tabela I
MÉDIA DA ESTIMATIVA PARA OS PARÂMETROS DE $G(q)$ E $H(q)$

Parâmetro	Média
a	2.0000
b	0.7748
c	-0.9006
d	0.4875

IV. CONCLUSÕES

Na seção (II) obteve-se uma estimativa para o sistema (1), mesmo que o modelo ARX utilizado (4) não conseguisse representar o sistema completamente. Desta forma obteve-se estimativas polarizadas para os parâmetros a e b, como observado nas Figuras (2) e (3), além de que a informação estimada para o ruido não era representativa com a realidade, já que para o modelo ARX considera-se que o ruido é submetido a influencia dos mesmos polos da função de transferência $G(q)$.

Para resolver este grave problema da polarização foi utilizado o método dos mínimos quadrados generalizado, onde utilizando-se outro estimador (6) para sistema, obteve-se resultados bem mais promissores, que podem ser observados nas Figuras (4) e (5), com este algoritmo, foi possível considerar um modelo completo para representar o sistema, o que trouxe a estimativa alem dos parâmetros de $G(q)$ também da função $H(q)$.

Neste trabalho observamos que quando a família de modelos não representa o sistema real, tem-se erro de polarização das estimativas efetuadas. Desta forma, mesmo utilizando mais pontos para a simulação e fazendo-se mais simulações, os parâmetros estimados na média não chegam ao valor real do sistema. Foi apontado um algoritmo para resolução de problemas onde o modelo padrão dos mínimos quadrados (ARX) não pode ser utilizado, e os resultados foram apresentados.

APÊNDICE

1 - Script para Simulação do modelo ARX

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

% Definitions
Ts=10e-3;
% frequency used when u(t) is a sinusoidal signal.
freq=pi/20;

Tf=10*2*pi/freq;

STD=0.1;
tempo = 0:Ts:Tf;
N=size(tempo, 2);

M=300;

% TFs
G=tf([2],[1 -0.8], Ts);
% item 1 e 2
%H=tf([1 0],[1 -0.8], Ts);
H=tf([1 0.9],[1 -0.5], 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=[2; 0.8];
n=size(teta, 1);
% e entrada u saida do controlador
%phy=[ u(t-1); y(t-1)]

% numero de vezes que sera aplicado o metodo.
a=zeros(M,1);
b=zeros(M,1);
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 = 1
%     ran=rand(N, 1);
%     s=std(ran);
%     m=mean(ran);
%     % now rr has std=1nusoidal input signal
%     rr=(ran-m)/s;
    %sim
    rr=sin(freq*tempo);
    mean(rr)

    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)=u(t-1);
        phy(t, 2)=y(t-1);
    end

    % make sure, rank(phy) = n :)
    teta_r=inv(phy'*phy)*phy'*y;
    % to be used in grafic plot
    a(j)=teta_r(1);
    b(j)=teta_r(2);
end
PN=[a, b];
ma=mean(a)
sa=std(a);
mb=mean(b)
sb=std(b);
plot(a, b, 'bo');
hold;
plot(ma, mb, 'rx');
hold;
title('Simulacao do sistema para um modelo ARX')
xlabel('Valor da estimativa para a variavel a')
ylabel('Valor da estimativa para a variavel b')
legend('Estimativas', 'Media')

%valor da tabela chi-quadrado para 95% de confianca
chi = 5.991;
ang = linspace(0,2*pi,360)';
[avetor,SCR,avl] = princomp(PN);
```

```
Diagonal= diag(sqrt(chi*avl));
elipse=[cos(ang) sin(ang)] * Diagonal * avetor' +
    repmat(mean(PN), 360, 1);
line(elipse(:,1), elipse(:,2), 'linestyle', '-',
    'color', 'k');
```

Listing 1. Descriptive Caption Text

2 - Script para Simulação do modelo completo

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

% Definitions
Ts=1e-3;
% frequency used when u(t) is a sinusoidal signal.
freq=pi/20;

Tf=1*2*pi/freq;

STD=0.1;
tempo = 0:Ts:Tf;
N=size(tempo, 2);

M=100;

% TFs
G=tf([2],[1 -0.8], Ts);
% item 1 e 2
%H=tf([1 0],[1 -0.8], Ts);
H=tf([1 0.9],[1 -0.5], 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=[2; 0.8; 0.5; 0.9; 1; 1];
n=size(teta, 1);
% e entrada u saida do controlador
%phy=[ u(t-1); y(t-1)]

% numero de vezes que sera aplicado o metodo.
t1=zeros(M,1); t2=zeros(M,1); t3=zeros(M,1);
    t4=zeros(M,1);
t5=zeros(M,1); t6=zeros(M,1); ychap=zeros(M,1);
a=zeros(M,1); b=zeros(M,1); c=zeros(M,1);
    d=zeros(M,1);

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 = 1
    ran=rand(N, 1);
    s=std(ran);
    m=mean(ran);
    % now rr has std=1nusoidal input signal
    rr=(ran-m)/s;
    %sim
%     rr=sin(freq*tempo);
    yr=lsim(G, rr, tempo);
    ynoise=lsim(H, rh, tempo);
    y=yr+ynoise;
    u=rr;

  % in the first simulation, use only u(t) and y(t)
    if j==1
        phy=zeros(N, n-2);
    else
        phy=zeros(N, n);
    end

    for t=3:N
        phy(t, 1)=u(t-1);
        phy(t, 2)=-u(t-2);
        phy(t, 3)=y(t-1);
        phy(t, 4)=-y(t-2);
        if j~=1
            ychap(t)=t1(j-1)*u(t-1) -t2(j-1)*u(t-2)
                +t3(j-1)*y(t-1) -t4(j-1)*y(t-2)
                +t5(j-1)*ychap(t-1)
                -t6(j-1)*ychap(t-2);
            phy(t, 5)=ychap(t-1);
            phy(t, 6)=-ychap(t-2);
        end
    end

    % make sure, rank(phy) = n :)
    teta_r=inv(phy'*phy)*phy'*y;
    % to be used in grafic plot
    t1(j)=teta_r(1);
    t2(j)=teta_r(2);
    t3(j)=teta_r(3);
    t4(j)=teta_r(4);
    if j~= 1
        t5(j)=teta_r(5);
        t6(j)=teta_r(6);
    end
  % get values of a b c and d
    a(j)=t1(j);
    d(j)=t2(j)/t1(j);
    c(j)=-t3(j)+d(j);
    b(j)=t4(j)/t3(j);
end

PN=[a, b];
ma=mean(a)
mb=mean(b)
mc=mean(c)
md=mean(d)
% not useful
me=mean(t5)
mf=mean(t6)

% plot a x b graph
plot(a, b, 'bo');
hold;
plot(ma, mb, 'rx');
hold;
title('Simulacao do sistema - estimativa dos
    parametros a e b')
xlabel('Valor da estimativa para a variavel a')
ylabel('Valor da estimativa para a variavel b')
legend('Estimativas', 'Media')

%valor da tabela chi-quadrado para 95% de confianca
chi = 5.991;
ang = linspace(0,2*pi,360)';
[avetor,SCR,avl] = princomp(PN);
Diagonal= diag(sqrt(chi*avl));
elipse=[cos(ang) sin(ang)] * Diagonal * avetor' +
    repmat(mean(PN), 360, 1);
line(elipse(:,1), elipse(:,2), 'linestyle', '-',
    'color', 'k');

% plot c x d graph
figure(2);
```

```
plot(c, d, 'bo');
hold;
plot(mc, md, 'rx');
hold;
title('Simulacao do sistema - estimativa dos
    parametros c e d - Ruido')
xlabel('Valor da estimativa para a variavel c')
ylabel('Valor da estimativa para a variavel d')
legend('Estimativas', 'Media')

%valor da tabela chi-quadrado para 95% de confianca
chi = 5.991;
ang = linspace(0,2*pi,360)';
[avetor,SCR,avl] = princomp([c, d]);
Diagonal= diag(sqrt(chi*avl));
elipse=[cos(ang) sin(ang)] * Diagonal * avetor' +
    repmat(mean([c, d]), 360, 1);
line(elipse(:,1), elipse(:,2), 'linestyle', '-',
    'color', 'k');
```

Listing 2. Descriptive Caption Text

REFERÊNCIAS

[1] L. A. Aguirre, *Introdução à identificação de sistemas, Técnicas lineares e não-lineares aplicadas a sistemas reais*, 2nd ed. Belo Horizonte, Minas Gerais: Editora UFMG, 2004, vol. 1.

[2] T. Soderstrom and P. Stoica, *System idendification*. New york: Prentice Hall, 2001.