

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—Trabalho 5, identificar um modelo ARX e também um modelo para $S \in M$.

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.

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

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. [?]

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

Onde:

$$\begin{aligned} A(q, \theta) &= 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_naq^{-na} \\ B(q, \theta) &= b_1q^{-1} + b_2q^{-2} + \dots + b_nbq^{-nb} \\ C(q, \theta) &= 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_nqc^{-nc} \\ D(q, \theta) &= 1 + d_1q^{-1} + d_2q^{-2} + \dots + d_ndq^{-nd} \\ F(q, \theta) &= 1 + f_1q^{-1} + f_2q^{-2} + \dots + f_nfq^{-nf} \end{aligned}$$

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

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

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

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.

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

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

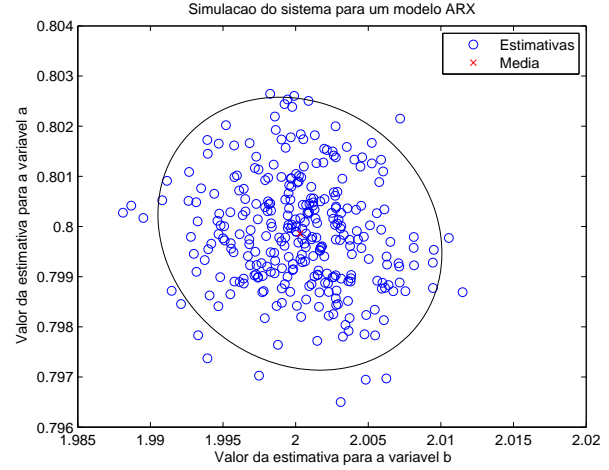


Figura 1. Simulação do sistema para uma entrada aleatória e utilizando o modelo ARX.

Aplicando na entrada do processo uma senoide de frequência $\pi/4$ obtém-se a estimativa como apresentado na Figura (3).

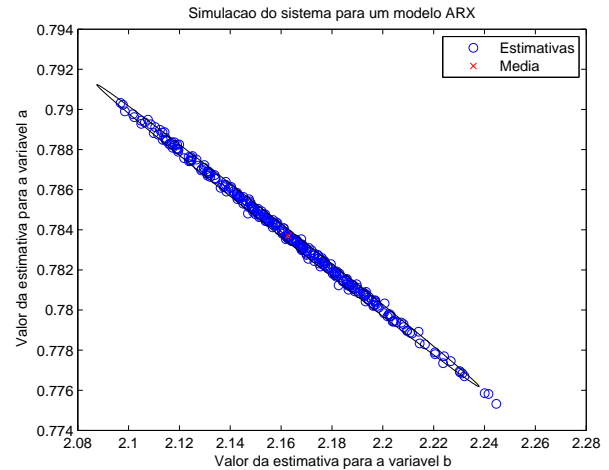


Figura 2. Simulação do sistema para uma entrada $\sin(\pi/4)$ e utilizando o modelo ARX.

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 (??).

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

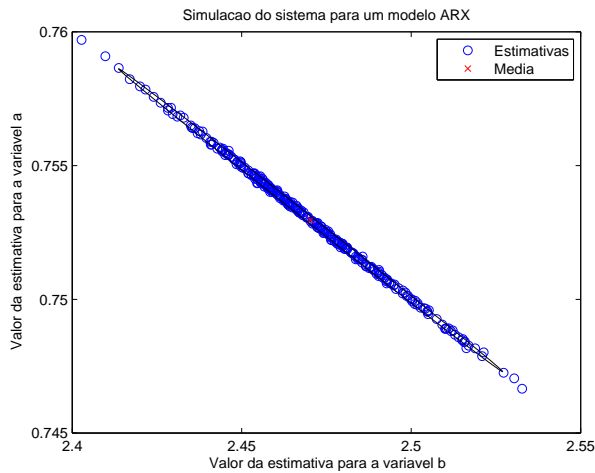


Figura 3. Simulação do sistema para uma entrada $\sin(\pi/4)$ e utilizando o modelo ARX.

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

III. MODELO COMPLETO

IV. CONCLUSÕES

heheheheh jlhadkjkhk;j hrlkjhe l;ohkwe rphfsl;khdf gl; isuhg-fliusdfnh lgsdfkjhg;

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 saída 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 random 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 random noise with std = 1
    ran=rand(N, 1);
    s=std(ran);
    m=mean(ran);
    % now rr has std=1;
    rr=(ran-m)/s;
    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 b')
ylabel('Valor da estimativa para a variavel a')
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));
ellipse=[cos(ang) sin(ang)] * Diagonal * avetor' +
    repmat(mean(PN), 360, 1);
line(ellipse(:,1), ellipse(:,2), 'linestyle', '-',
    'color', 'k');
```

Listing 1. Descriptive Caption Text