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

II. Questão 2

Questão: Seja o sistema ARX (1):

$$G_o(z) = \frac{2}{z - 0.8} \quad H_o(z) = \frac{z}{z - 0.8} \tag{1}$$

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).

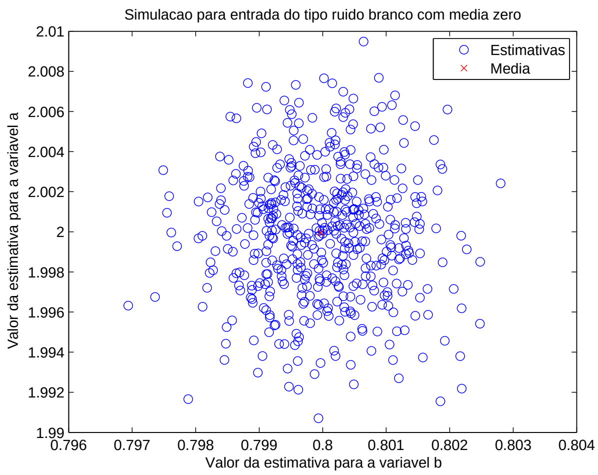

Figura 1. Entrada aleatória aplicada no processo para a identificação do sistema.

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 (**??**).

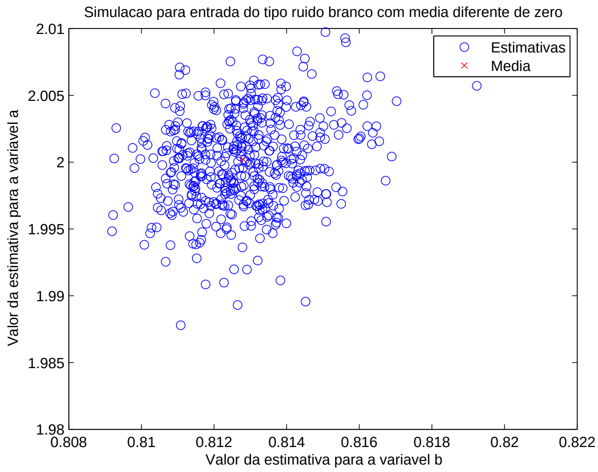

Figura 2. Entrada aleatória aplicada no processo para a identificação do sistema. Média diferente de zero.

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 (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;
```

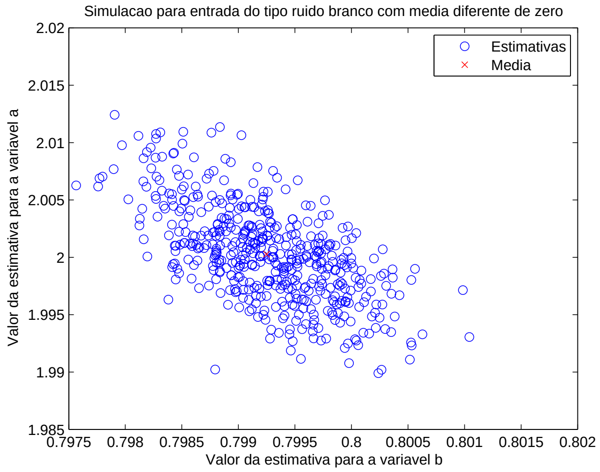

Figura 3. Entrada aleatória aplicada no processo para a identificação do sistema. Ruido sujeito a $H(z) = 1$.

```
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