Modelagem e Identificação de sistemas lineares

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*—**
Neste trabalho será apresentado diversos meios para a identificação de sistemas lineares. Existem dois grupos principais de métodos para esta identificação, sendo um deles conhecido como *identificação não paramétrica* **onde existem infinitos parâmetros para serem estimados e que normalmente é utilizado para identificação de funções gráficas. Outro método é conhecido como** *identificação paramétrica* **onde o número de parâmetros a ser estimado é finito. Este último método será mais abordado neste trabalho, por possuir uma aplicabilidade maior devido a possibilidade de estimar processos em funções matemáticas que descrevem o comportamento do sistema muitas vezes com mais informação que os métodos gráficos.**

Para este trabalho será utilizado um processo de controle de posição angular, controlado por um motor de corrente continua. Serão apresentados diversos métodos de identificação e ao fim será feito um comparativo entre os resultados obtidos.

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

I. Introdução

II. Modelagem do sistema

III. Modelagem não paramétrica

IV. Modelos paramétricos simples

V. Método dos mínimos quadrados

O método dos mínimos quadrados (MMQ) é um dos mais conhecidos e mais utilizados nas mais diversas áreas da ciência e tecnologia. A origem da ideia básica pode ser encontrada nos trabalhos de Gaus sobre o estudo astronômicos. [1]

A. Sistema com solução única

Considerando-se que o sistema que será observado seja linear e invariante no tempo. Se a função f que descreve o sistema for não linear o sistema poderá em principio ser identificado por modelos não lineares. Com base nestas restrições temos que:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

$$y = X\theta$$

Com $X \in \Re^{nxn}$. Desde que X seja não singular é possível determinar θ:

$$\theta = X^{-1}y \tag{1}$$

Para sistemas sobredeterminados onde $N > n$, A variável X da equação (1) fica $X \in \Re^{Nxn}$. Como esta matriz não é quadrada, não é possível de ser invertida. Multiplicando-se a equação (1) por X^T tem-se: [1]

$$X^T y = X^T X\theta$$

De onde vem:

$$\theta = [X^T X]^{-1} X^T y \tag{2}$$

O método dos mínimos quadrados minimiza o critério apresentado em (3).

$$J(\theta) = \frac{1}{2N}\sum_{t=1}^{N}[y(t) - \hat{y}(t,\theta)]^2 \tag{3}$$

Onde $\hat{y}(t,\theta)$ é a predição do sistema e pode ser representado como abaixo:

$$\hat{y}(t,\theta) = \theta^T\phi(t)$$

Desta forma pode se dizer que o sistema real é o próprio sistema estimado mais algum erro de estimativa:

$$y(t) = \hat{y}(t,\theta) + e(t) = \theta^T\phi(t) + e(t)$$

VI. Classes de modelos Genéricas

VII. Conclusões

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;
Tf=10;
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 0.9],[1 -0.5], Ts);
% item 3
%H=tf([1],[1], 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);
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=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;
    % to be used in grafic plot
    a(j)=teta_r(1);
    b(j)=teta_r(3);
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 para entrada do tipo ruido branco
    com media zero - 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));
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

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.