

# Identificação determinística por subespaços

S. D. M. Borjas <sup>1</sup>, DCEN, Universidade Federal Rural do Semi-árido, UFERSA, Campus Mossoró, Av. Francisco Mota 570, Bairro Costa e Silva, 59625-900 Mossoró, RN, Brasil.

C. Garcia <sup>2</sup>, Universidade de São Paulo -USP Departamento PTC Escola Politécnica Campus USP -Capital 05508-900 São Paulo, SP, Brasil.

**Resumo.** Neste artigo é apresentada uma visão geral sobre o problema de identificação por subespaços em malha aberta. Existem diversos algoritmos que solucionam este problema (MOESP, DSR, N4SID, CVA). Baseado nos métodos MOESP e N4SID é apresentado um algoritmo alternativo para identificar sistemas determinísticos operando em malha aberta. Um processo simulado é usado para mostrar o desempenho deste algoritmo.

**Palavras-chave.** Identificação por subespaços, métodos de identificação, sistemas de identificação determinística.

## 1. Introdução

A teoria da realização de sistemas para o caso determinístico pretende, em primeira instância, resolver o seguinte problema: dada a função de transferência, como encontrar um ou mais modelos no espaço de estados, que representem tal sistema. Existem técnicas que permitem solucionar tal problema, sobretudo para o caso SISO. Como a generalização para o sistema MIMO não é trivial, uma alternativa é trabalhar com as matrizes de resposta ao impulso. Existem muitas técnicas para solucionar este problema, porém a teoria da realização, utilizando a decomposição de valores singulares (SVD), tornou-se predominante por ser computacionalmente robusta [6]. Esta técnica é usada nos métodos de identificação por subespaços. Nos métodos de identificação de sistemas dinâmicos por subespaços são tratados modelos de sistemas lineares invariantes no tempo em espaço de estados operando em tempo discreto. Pelas restrições citadas, pode parecer uma classe altamente restrita de modelos (especialmente por serem lineares), no entanto é bastante surpreendente como muitos processos industriais podem ser descritos com precisão por este tipo de modelo [4], [2] e [5]. Por outro lado, existe um grande número de ferramentas disponíveis de projeto de controladores para tais sistemas e modelos. Para empregar tais métodos é necessário o uso de ferramentas de teoria de sistemas, geometria

---

<sup>1</sup>santos.borjas@ufersa.edu.br

<sup>2</sup>clgarcia@lac.usp.br

e álgebra linear [3] e [4]. Os métodos lineares de identificação por subespaços estão relacionados com modelos da forma

$$x_k + 1 = Ax_k + Bu_k + Ke_k \quad (1.1)$$

$$y_k = Cx_k + Du_k + e_k \quad (1.2)$$

onde  $u_k \in \mathfrak{R}^m$ ,  $e, y(k) \in \mathfrak{R}^l$  são, respectivamente os valores medidos das entradas e saídas no instante  $k$  dos processos com  $m$  entradas e  $l$  saídas,  $x_k \in \mathfrak{R}^n$  é o vetor de estados com valor inicial  $x_0$ ,  $e_k$  é um sinal do tipo ruído branco com média zero e matriz de covariância  $E(e_k, e_k^T) = \Delta$ .  $A, B, C, D$  e  $K$  são matrizes de dimensões apropriadas. Os algoritmos de identificação por subespaços calculam modelos em espaço de estados, a partir de dados de entrada e saída. É uma prática comum distinguir três casos distintos na identificação de sistemas por subespaços [6]: a) o caso puramente determinístico  $e_k = 0$ , b) o caso puramente estocástico  $u_k = 0$  e c) o caso combinado determinístico e estocástico. Neste trabalho será tratado o caso puramente determinístico. O problema determinístico é: dada uma sequência de dados de entrada e saída determine a orden  $n$  e as matrizes  $(A, B, C, D)$  do sistema desconhecido. Existem diversos modelos que solucionam esse problema. Entre os mais populares há o método MOESP (Multivariable Output-Error State sPace)[9] e N4SID (Numerical gorithms for Subspace State Space System IDentification)[6]. O objetivo deste trabalho é apresentar um algoritmo que usa a tecnica do método MOESP para computar uma certa matriz da qual se recupera as matrizes do sistema usando a tecnica N4SID.

### 1.1. Equações matriciais por subespaços

Das equações (1.1) e (1.2) obtém-se

$$Y_f = \Gamma_i X_f + H_i U_f \quad (1.3)$$

onde a matriz  $U_f$  é definida de forma similar à matriz  $Y_f$ . As matrizes  $Y_f, H_i$ , são definidas por

$$Y_f = \begin{bmatrix} y_i & y_{i+1} & \cdots & y_{i+j-1} \\ y_{i+1} & y_{i+2} & \cdots & y_{i+j} \\ \vdots & \vdots & \ddots & \vdots \\ y_{2i-1} & y_{2i} & \cdots & y_{2i+j-2} \end{bmatrix} \quad (1.4)$$

$$H_i = \begin{bmatrix} D_i & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{i-2}B & CA^{i-3} & \cdots & D \end{bmatrix} \quad (1.5)$$

O número de colunas em  $Y_f$  e  $U_f$  é  $j = N - 2i + 1$ , onde  $N$  representa a quantidade de dados e  $i$  é o número de linhas definido pelo usuário (por exemplo  $i = 10$ ).

$\Gamma_i \in \mathbb{R}^{lixn}$  é a matriz de observabilidade estendida e é definida por

$$\Gamma_i^T = [ (C)^T \quad (CA)^T \dots (CA^{i-1})^T ]^T \quad (1.6)$$

onde  $(*)^T$  denota a transposta da matriz  $(*)$ .  $X_p = X_0 = [ x_0, \dots, x_{j-1} ]$  e  $X_f = X_i = [ x_i, \dots, x_{i+j-1} ]$  representam os estados passados e futuros respectivamente, o símbolo  $p$  denota dados passados e  $f$  os dados futuros.

## 1.2. Projção ortogonal e projção oblqua

A projção ortogonal do espao linha de A sobre o espao linha de B é [6]

$$A/B = AB(BB^T)^\dagger B \quad (1.7)$$

onde  $(*)^\dagger$  denota a pseudo-inversa da matriz  $(*)$ .

A projção oblqua do espao de linhas de  $G$  no espao de linhas de  $H$  sobre o espao de linhas de  $J$  é [6]

$$G/HJ = [G/H^\perp].[J/H^\perp]^\dagger.J \quad (1.8)$$

onde  $(*)^\perp$  denota o complemento ortogonal da matriz  $(*)$ . Propriedades da projção ortogonal e projção oblqua

$$A_x/A_x^\perp = 0 \quad (1.9)$$

$$A_x/A_x C_x = 0 \quad (1.10)$$

Para a prova, ver [6].

## 2. Métodos de identificação

### 2.1. Método MOESP

A seguir é apresentado o método MOESP básico, do qual um grande número de variações foi criado para diferentes tipos de problemas [9]. O objetivo deste método é computar  $\Gamma_i$  a partir da equação (1.3), então aplicando a projção ortogonal da equação (1.3) sobre o espao linha de  $U_f^\perp$  resulta:

$$Y_f/U_f^\perp = \Gamma_i X_f/U_f^\perp + H_i^d U_f/U_f^\perp \quad (2.1)$$

Pela propriedade (1.9) tem-se  $U_f/U_f^\perp = 0$ . A equação (2.1) pode ser simplificada

$$Y_f/U_f^\perp = \Gamma_i X_f/U_f^\perp \quad (2.2)$$

a equação (2.2) indica que o espaço coluna de  $\Gamma_i$  pode ser estimado pela SVD de  $Y_f/U_f^\perp$  [9].  $\Gamma_i$  pode ser estimada da fatoração LQ a partir dos dados de entrada e saída, na forma

$$\begin{bmatrix} U_f \\ U_p \\ Y_p \\ Y_f \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} \begin{bmatrix} Q1 \\ Q2 \\ Q3 \\ Q4 \end{bmatrix} \quad (2.3)$$

A projeção ortogonal do lado esquerdo de (2.2) pode ser computada da matriz  $RR = [R_{42}R_{43}]$ . A SVD da matriz  $RR$  é [9]

$$[RR] = [U_1 \ U_2] \begin{bmatrix} S_n & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = USV \quad (2.4)$$

a ordem  $n$  do sistema é igual ao número de valores singulares não nulos em  $S$ . O espaço coluna de  $U_1$  aproxima consistentemente a matriz  $\Gamma_i$  [9]

$$\Gamma_i \approx U_1 \quad (2.5)$$

as matrizes  $A$  e  $C$  são obtidas de  $C = \Gamma_i(1 : l, 1 : n)$  e  $\Gamma_i(1 : l(i-1), 1 : n)A = \Gamma_i(l+1 : il, 1 : n)$  respectivamente. Para obter as matrizes  $B$  e  $D$  ver [9].

## 2.2. Método N4SID

O método N4SID soluciona o problema de identificação determinística recuperando os estados passados e futuros do sistema desconhecido. A sequência de estados  $X_f$  pode ser expressa  $X_f = L_p W_p$  como combinação linear das entradas passadas e saídas passadas [6]. Logo, substituindo-se  $X_f = L_p W_p$  na equação (1.3) tem-se

$$Y_f = \Gamma_i L_p W_p + H_i U_f \quad (2.6)$$

Aplicando-se a projeção ortogonal da equação (2.6) sobre o espaço linha de  $U_f^\perp$  resulta:

$$Y_f/U_f^\perp = \Gamma_i L_p W_p/U_f^\perp \quad (2.7)$$

Multiplicando-se a equação (2.7) por  $[W_p/U_f^\perp]^\dagger \cdot W_p$  de ambos os lados,

$$Y_f/U_f^\perp [W_p/U_f^\perp]^\dagger \cdot W_p = \Gamma_i L_p W_p/U_f^\perp [W_p/U_f^\perp]^\dagger \cdot W_p \quad (2.8)$$

como  $W_p = W_p/U_f^\perp [W_p/U_f^\perp]^\dagger$  e usando a equação (1.8) tem-se a projeção oblíqua  $\Theta_i$  definida por [6]

$$\Theta_i = Y_f/U_f^\perp W_p = \Gamma_i X_f \quad (2.9)$$

De forma similar é obtido  $\Theta_{i+1}$ , ver [6].  $\Theta_i$  dado na equação (2.9), pode ser computado da fatoração LQ a partir dos dados de entrada e saída, colocados na forma

$[U_p^T U_f^T Y_p^T Y_f^T]^T$ . A equação (2.9) indica que o espaço coluna de  $\Gamma_i$  pode ser estimado pela SVD de  $\Theta_i$ , então resulta

$$\Gamma_i \approx U_1 S_1^{1/2} \quad (2.10)$$

Usando a equação (2.10) computa-se  $X_i = \Gamma_i^\dagger \Theta_i$  e  $X_{i+1} = \Gamma_{i-1}^\dagger \Theta_{i-1}$ , logo as matrizes do sistema são estimadas da equação (2.11) [6].

$$\begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix} \quad (2.11)$$

### 2.3. Combinação dos métodos MOESP e N4SID

O algoritmo MON1 é um algoritmo alternativo que resulta da combinação da técnica MOESP e da técnica N4SID. Esse algoritmo estima a matriz de observabilidade estendida  $\Gamma_i$  no mesmo sentido do método MOESP e usa a técnica do método N4SID para estimar as matrizes do sistema  $A, B, C$  e  $D$  a partir da equação (2.11). Para isto é necessário conhecer as matrizes das sequências dos estados  $X_i$  e  $X_{i+1}$ , portanto o problema agora é computar as sequências de estado. A projeção oblíqua  $\Theta_i$  dada pela equação (2.9) pode ser computada a partir da equação (2.3) [1]

$$\Theta_i = L_{32} L_{22}^\dagger \begin{bmatrix} L_{21} & L_{22} \end{bmatrix} \quad (2.12)$$

onde  $L_{32} = RR$ ,  $L_{22} = \begin{bmatrix} R_{22} & 0 \\ R_{32} & R_{33} \end{bmatrix}$  e  $L_{21} = \begin{bmatrix} R_{21} \\ R_{31} \end{bmatrix}$ . Uma estimativa da sequência de estados totais é [1]

$$X = (\Gamma_i)^\dagger L_{32} L_{22}^\dagger \begin{bmatrix} L_{21} & L_{22} \end{bmatrix} W_p \quad (2.13)$$

e uma estimativa das matrizes  $X_i$  e  $X_{i+1}$  é dada por  $X_i = [x_k \ x_{k+1} \ x_{k+2} \ \cdots \ x_{k+N-2}]$  e  $X_{i+1} = [x_{k+1} \ x_{k+2} \ x_{k+3} \ \cdots \ x_{k+N-1}]$ . Assim, as matrizes do sistema podem ser estimadas a partir da equação (2.11).

este método pode ser resumido no seguinte algoritmo

#### Algoritmo MON1

1) Construir as matrizes  $U_f, U_p$  e  $Y_f, Y_p$  e a fatoração  $LQ$  dada em (2.3). 2) Calcular a SVD da matriz  $L_{32} = RR$ . 3) Determinar a ordem do sistema por inspeção dos valores singulares em  $S$  dada em (2.4). 4) Determine  $\Gamma_i$  da equação (2.5), a sequência de estados  $X$  de (2.13) e as sequências  $X_{i+1}$  e  $X_i$ . 5) Determinar as matrizes  $A, B, C$ , e  $D$  da equação (2.11).

## 3. PROCESSO SIMULADO

Em [8] e [7] é citado um exemplo para fazer a identificação em malha fechada de uma planta e seu controlador. Esta planta representa um sistema SISO e é usada

neste artigo para fazer a identificação determinística em malha aberta. As matrizes do sistema que simulam a planta são:  $a = [4.40 \ 1 \ 0 \ 0 \ 0; -8.09 \ 0 \ 1 \ 0 \ 0; 7.83 \ 0 \ 0 \ 1 \ 0; -4 \ 0 \ 0 \ 1; 0.86 \ 0 \ 0 \ 0 \ 0]$ ;  $b = [0.00098; 0.01299; 0.01859; 0.0033; -0.00002]$ ;  $c = [1 \ 0 \ 0 \ 0 \ 0]$  e  $d=0$ . Dadas as matrizes do sistema, o passo seguinte é coletar os dados de entrada e saída. Isto é dado pelo seguinte algoritmo

$N = 1000$ ; ( número de dados a coletar)

$ny = 1$  (número de saídas)

$u1 = \text{idinput}([N, ny], 'prbs', [0 \ 0.3])$ ;

$ut = [u1 - \text{mean}(u1)]$  (dados das entradas coletadas)

$yt = \text{dlsim}(a, b, c, d, ut)$ ; (dados das saídas coletadas)

Para este modelo com sinal de entrada  $u1$ , foram coletados 1000 dados, dos quais 700 foram aplicados para identificação e o restante para validação. Os sinais pré-tratados usados na identificação são mostrados na figura 1.

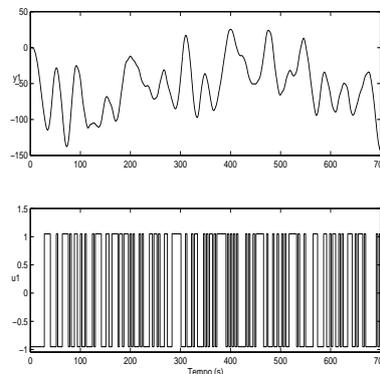


Figure 1: Sinais de saída e entrada usados na identificação.

Agora deve-se encontrar o melhor modelo que simule o processo. Neste trabalho são usados os modelos  $N4sid1$ ,  $N4sid2$  e  $MOESP$ , onde  $N4sid1$  denota aqui o algoritmo determinístico 1 (empregando estados),  $N4sid2$  denota aqui o algoritmo determinístico 2 (empregando matriz de observabilidade estendida) [6],  $MOESP$  denota aqui o algoritmo  $MOESP$  (caso determinístico). O algoritmo alternativo proposto para sistemas determinísticos é o  $MON1$  (empregando estados). Todos eles foram implementados no Matlab 6.5. Com o objetivo de ver o desempenho destes algoritmos, foram comparados com os algoritmos  $N4SID$ ,  $MPVerh$  e  $PEM$ . O  $N4SID$  se encontra implementado no Toolbox do Matlab,  $MPVerh$  denota aqui o método  $MOESP$  implementado por Michael Verhaegen. No total foram usados 7 algoritmos para fazer a identificação determinística.

Comparar a simulação do modelo obtido com dados medidos é provavelmente a forma mais usual de se validar um modelo. Nesse caso, deseja-se saber se o modelo reproduz ao longo do tempo os dados observados. Os indicadores de desempenho mais usados são média relativa do erro quadrático ( $MRSE$ ), média da variância

relativa (MVAF) e o FIT, os quais são definidos como

$$MRSE(\%) = \frac{1}{l} \sum_{i=1}^l \sqrt{\frac{\sum_{j=1}^N (y - \hat{y})^2}{\sum_{j=1}^N (\hat{y})^2}} \cdot 100 \quad (3.1)$$

$$MVAF(\%) = \frac{1}{l} \sum_{i=1}^l \left(1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)}\right) \cdot 100 \quad (3.2)$$

$$FIT(\%) = \text{norma}\left(\frac{(y - \hat{y})^2}{\sqrt{N}}\right) \cdot 100 \quad (3.3)$$

onde  $y$  é a saída real e  $\hat{y}$  é a saída estimada pelo modelo obtido. O índice MRSE é muito usado na literatura e o índice MVAF é usado pelo SMI toolbox. O índice FIT é usado pela função "compare" no Matlab. Estes índices de desempenho são empregados para se avaliar a qualidade do modelo produzido por cada algoritmo, como mostra a tabela 1.

A ordem  $n = 5$  do sistema é dada pelos valores singulares mais significativos, como mostra a figura 2.

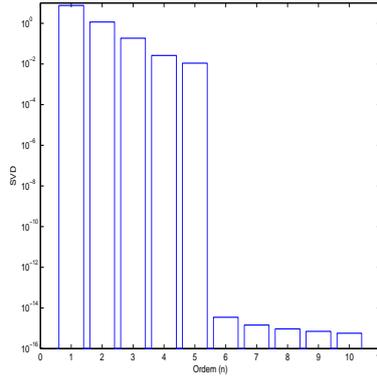


Figure 2: Valores singulares para o modelo MON1.

Analisando-se os valores da tabela 1, todos os modelos tiveram um bom desempenho em termos de validação. Verifica-se que o tempo de processamento para obtenção do modelo é menor para MON1. Com o objetivo de visualizar o desempenho do algoritmo proposto, optou-se pelo método MON1 para identificar o processo. A figura 3 mostra as saídas do processo real (linha contínua) e aquelas geradas pelo modelo determinístico identificado (linha tracejada). Pode-se observar que o modelo identificado reproduz muito bem as principais características do processo. Foram consideradas condições iniciais nulas.

Table 1: Resultados numéricos do desempenho dos algoritmos .

Algoritmo	Tempo (s)	FIT (%)	MRSE (%)	MVAF (%)
N4sid1	0.234	100	0.0039	100
N4sid2	0.203	100	0.0046	100
MOESP	0.203	100	$2296x10^{-7}$	100
MON1	0.203	100	$9629x10^{-7}$	100
N4SID	0.359	100	0.0031	100
MPVerh	0.25	100	$9.97x10^{-9}$	100
PEM	1.187	100	0.0044	100

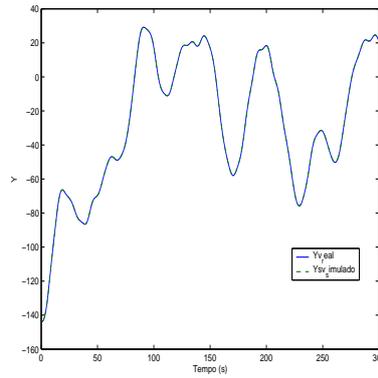


Figure 3: Comparação das respostas do processo real (contínua) versus modelo (linha tracejada).

Da figura 3 pode-se observar que a resposta do modelo obtido é similar à do sistema real, mas se é conhecida a ordem do sistema real, esta não é necessariamente a mesma para o modelo obtido, portanto as matrizes do sistema real e do modelo obtido têm diferentes dimensões.

## 4. CONCLUSÃO

No método alternativo MON1 a matriz de observabilidade estendida é computada pela técnica MOESP e a partir desta é calculada uma sequência de estados totais. A sequência de estados passados e futuros é computada a partir da sequência de estados totais, a qual tem unicamente um estado inicial, oposto ao método N4sid, pois para computar a sequência de estados passados e futuros usa duas projeções oblíquas diferentes, levando a um maior esforço computacional. Para avaliar o desempenho do algoritmo alternativo, seis algoritmos de identificação foram aplicados ao processo em estudo e seus resultados foram comparados via validação cruzada,

usando três tipos de critério. Para este caso específico, todos os algoritmos apresentaram bom desempenho. O modelo PEM foi o mais lento nos sete casos de identificação. A identificação determinística é um caso ideal, pois na prática os dados encontram-se corrompidos por ruídos, no entanto o modelo MON1 pode ser ampliado para a identificação de sistemas determinísticos-estocásticos.

**Abstract.** This paper present an overview of the identification problem for subspace in open loop. There are several algorithms that solve this problem (MOESP, DSR, N4SID, CVA). Based on the methods MOESP and N4SID is presented an alternative algorithm to identify deterministic systems operating in open loop. A process simulation is used to show the performance of this is algorithm.

## References

- [1] S. D. Borjas, C. Garcia, Subspace identification using the integration of MOESP and N4SID methods applied to the Shell benchmark of a distillation column, *TEMA-Tend.Mat. Apl. Comput.*, (2010).
- [2] S. D. M. Borjas, C. Garcia, Modelagem de FCC usando métodos de identificação por predição de erro e por subespaços. *IEEE América Latina, Revista virtual - na Internet*, **2**, No. 2, (2004), 108-113.
- [3] B. De Moor, P. Van Overschee and W. Favoreel, Algorithms for subspace state space system identification - an overview. In B. Datta (Ed.), *Applied and computational control, signal and circuits*, Vol. 1, pp. 247-311. Birkhauser: Boston (Chapter 6), 1999.
- [4] W. Favoreel, B. De Moor and P. Van Overschee, Subspace state space system identification for industrial processes. *Journal of Process Control*, **10**, No.3, (2000), 149-155.
- [5] P. Roberto, G. Kurka, H. Cambraia, Application of a multivariable input-output subspace identification technique in structural analysis, *Journal of Sound and Vibration*, **312**, No. 3, (2008), 461-47.
- [6] P. Van Overschee B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*", Dordrecht: Kluwer Academic Publishers, 1996.
- [7] P. Van Overschee, B. De Moor, Closed loop subspace systems identification, em "Proc. 36th IEEE Conference on Decision and Control", pp 1848-1853, San Diego, 1997.
- [8] M. Verhaegen, "Application of a subspace model identification technique to identify LTI systems operating in closed loop", *Automatica*, **29**, No 4, (1993), 1027-1040.

- [9] M. Verhaegen, P. Dewilde, "Subspace model identification. part i: the output-error state-space model identification class of algorithms", *International Journal of Control*, **56**, No. 1, (1992), 1187-1210.