

Identificação Determinística por Subespaços

S.D.M. BORJAS ¹, Departamento de Ciências Exatas e Naturais, Universidade Federal Rural do Semi-árido, UFRSA, Campus Mossoró, Av. Francisco Mota, 570, Bairro Costa e Silva, 59625-900 Mossoró, RN, Brasil.

C. GARCIA ², Escola Politécnica, USP - Universidade de São Paulo, Av. Prof. Luciano Gualberto, trav. 3, 158, Cidade Universitária, Butantã, 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 os autores apresentam um algoritmo alternativo para identificar sistemas determinísticos operando em malha aberta. Dois processos simulados, um SISO e um MIMO são usados para mostrar o desempenho deste algoritmo.

Palavras-chave. Identificação por subespaços, métodos de identificação, 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 muitas técnicas que permitem solucionar tal problema, sobretudo para o caso SISO (Single Input, Single Output), porém a teoria da realização, utilizando a decomposição em valores singulares (SVD), tornou-se predominante por ser computacionalmente robusta [8]. Esta técnica é usada nos métodos de identificação por subespaços que lidam com 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 [2], [5] e [7]. 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 e Álgebra Linear [6] e [5]. Os métodos lineares de identificação por subespaços estão relacionados com modelos da forma:

$$x_{k+1} = Ax_k + Bu_k + Ke_k \quad (1.1)$$

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

¹santos.borjas@ufersa.edu.br

²clgarcia@lac.usp.br

onde $u_k \in \mathfrak{R}^m$ e $y_k \in \mathfrak{R}^l$ são, respectivamente, os valores medidos das entradas e das 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 [8]: 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 postulado como: dada uma sequência de dados de entrada e saída determine a ordem n e as matrizes (A, B, C, D) do sistema desconhecido. Existem diversos métodos que solucionam esse problema. Entre os mais populares há o método MOESP (Multivariable Output-Error State sPace) [10] e N4SID (Numerical algorithms for Subspace State Space System IDentification) [8]. O objetivo deste trabalho é apresentar um algoritmo que usa a técnica do método MOESP para computar uma certa matriz da qual se recuperam as matrizes do sistema usando a técnica 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 e H_i são definidas por:

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

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

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 $i = 10$). $\Gamma_i \in \mathfrak{R}^{lixn}$ é a matriz de observabilidade estendida e é definida por:

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

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

1.2. Projecção ortogonal e projecção oblíqua

A projecção ortogonal do espaço linha de A sobre o espaço linha de B é [8]:

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

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

A projecção oblíqua do espaço de linhas de G no espaço de linhas de H sobre o espaço de linhas de J é [8]:

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

onde $(*)^\perp$ denota o complemento ortogonal da matriz $(*)$. Propriedades da projecção ortogonal e projecção oblíqua:

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

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

Para a prova, ver [8].

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 [10]. O objetivo deste método é computar Γ_i a partir da equação (1.3), então aplicando a projecção ortogonal da equação (1.3) sobre o espaço linha de U_f^\perp resulta:

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

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 Γ_i pode ser estimado pela SVD de Y_f/U_f^\perp [10]. Γ_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 projecção ortogonal do lado esquerdo de (2.2) pode ser computada a partir da matriz $RR = [R_{42}R_{43}]$. A SVD da matriz RR é [10]:

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

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 Γ_i [10]:

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

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 [10].

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 por $X_f = L_p W_p$ como combinação linear das entradas passadas e saídas passadas [8]. 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 Θ_i definida por [8]:

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

De forma similar é obtido Θ_{i+1} , ver [8]. Θ_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 Γ_i pode ser estimado pela SVD de Θ_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) [8]:

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

2.3. Combinação dos métodos MOESP e N4SID

O algoritmo MON1 (modelo 1, combinação dos métodos MOESP e N4SID para o caso determinístico) é um método alternativo proposto pelos autores. Esse algoritmo estima a matriz de observabilidade estendida Γ_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 Θ_i dada pela equação (2.9) pode ser computada a partir da equação (2.3) [4]:

$$\Theta_i = L_{32}L_{22}^\dagger [L_{21} \quad L_{22}] \quad (2.12)$$

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 [4]:

$$X = (\Gamma_i)^\dagger L_{32}L_{22}^\dagger [L_{21} \quad L_{22}] W_p \quad (2.13)$$

e uma estimativa das matrizes X_i e X_{i+1} é dada por $X_i = [x_k \quad \cdots \quad x_{k+N-2}]$ e $X_{i+1} = [x_{k+1} \quad \cdots \quad x_{k+N-1}]$. 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 , Y_f e 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) Determinar Γ_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

3.1. Caso SISO

Em [9] e [11] 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: $a = [4.40 \ 1 \ 0 \ 0 \ 0; -8.09 \ 0 \ 1 \ 0 \ 0; 7.83 \ 0 \ 0 \ 1 \ 0; -4 \ 0 \ 0 \ 0 \ 1; 0.86 \ 0 \ 0 \ 0 \ 0]$; $b = [0.00098; 0.01299; 0.01859; 0.0033; -0.00002]$; $c = [1 \ 0 \ 0 \ 0 \ 0]$ e $d=0$. Dadas as matrizes do sistema, o passo seguinte é coletar os dados de entrada e saída. Isto é feito 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 usados na identificação são mostrados na figura 1. O sinal de excitação empregado foi o PRBS (Pseudo-Random Binary Sequence), que é amplamente usado na identificação de sistemas lineares.

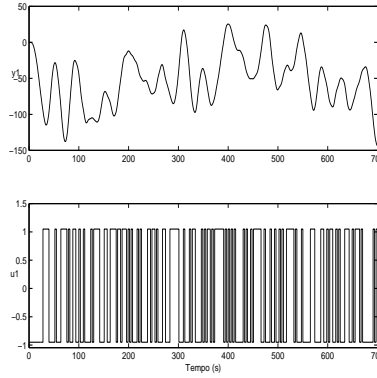


Figure 1: Sinais de saída e entrada usados na identificação.

Agora deve-se encontrar o melhor modelo que simule o processo. Neste trabalho são usados os algoritmos N4sid1, N4sid2 e MOESP, onde N4sid1 denota aqui o algoritmo determinístico 1 (empregando estados), N4sid2 corresponde ao algoritmo determinístico 2 (empregando matriz de observabilidade estendida) [8] e MOESP representa 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. Foram ainda empregados os algoritmos N4SID, MPVerh e PEM (Prediction Error Methods). O N4SID e o PEM se encontram implementados no Toolbox do Matlab e o 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 aqui usados são média relativa do erro quadrático (MRSE), média da variância relativa (MVAF) e o FIT, os quais são definidos como:

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

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

$$FIT(\%) = \left(1 - \frac{\|y - \hat{y}\|_2}{\|y - \text{mean}(y)\|_2}\right) \cdot 100 \quad (3.3)$$

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 SI (System Identification) Toolbox do Matlab. 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.

Analisando-se os valores da tabela 1, todos os modelos tiveram um bom desem-

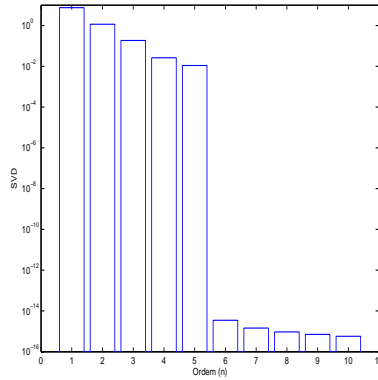


Figure 2: Valores singulares para o modelo MON1.

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	$2296.x10^{-7}$	100
MON1	0,203	100	$9629.x10^{-7}$	100
N4SID	0,359	100	0,0031	100
MPVerh	0,250	100	$9,97.x10^{-9}$	100
PEM	1,187	100	0,0044	100

penho em termos de validação. Verifica-se que o tempo de processamento para a obtenção do modelo é menor para MON1, N4sid2 e MOESP. 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.

3.2. Caso MIMO

Para avaliar o desempenho do algoritmo MON1 para o caso MIMO (Multiple-Input and Multiple Output), é usado um modelo em espaço de estados com matrizes

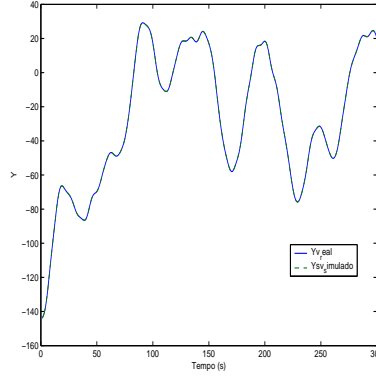


Figure 3: Comparação das respostas do processo real (linha contínua) versus modelo (linha tracejada).

aleatórias. A função *dss* do Matlab 6.5 permite gerar um modelo em espaço de estados, com matrizes do sistema (A , B , C , D) na forma aleatória. A ordem escolhida do modelo é 4, que é igual ao posto de A . O seguinte algoritmo gerou um sistema MIMO de duas entradas e duas saídas: $M=idss(dss(4,2,2)); [A,B,C,D]=th2ss(M)$. As matrizes do sistema são dadas por:

$$A = \begin{pmatrix} 0,2075 & 0,6037 & 0,0467 & 0,3296 \\ 0,0647 & -0,1354 & 0,8093 & 0,2532 \\ -0,6549 & 0,2657 & -0,1127 & 0,4057 \\ -0,2056 & -0,5368 & -0,0829 & 0,3546 \end{pmatrix}$$

$$B = \begin{pmatrix} 0,1542 & 0 \\ -0,3429 & -0,9224 \\ -0,4785 & 1,5295 \\ 0,5955 & -0,0031 \end{pmatrix}$$

$$C = \begin{pmatrix} -0,2141 & 0 & -0,9223 & 0 \\ -2,4532 & 0 & 0,0617 & -0,2908 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 2,1346 \\ -0,5508 & 0,5446 \end{pmatrix}$$

Obtidas as matrizes do sistema, o passo seguinte é coletar os dados de entrada e saída do sistema determinístico. As amplitudes do sinal de entrada devem ser tão pequenas quando possível, sem comprometer o funcionamento adequado do sistema, nem levar o sistema a operar em outra faixa. O sinal de entrada para o sistema M é um PRBS, o qual é persistentemente excitante e o sinal de saída é gerado através da função *sim* do Matlab. Isto é dado pelo seguinte algoritmo: $ut = idinput([1000, 2], 'prbs', [0 0.3]);$ (sinal de entrada); $yt = sim(M, ut);$ (sinal de saída). Para o modelo M com sinal de entrada ut , foram coletados 1000 dados, dos quais 700 foram aplicados para identificação e o restante para validação. O passo seguinte é encontrar o melhor modelo que simule o processo M , para os algoritmos citados acima. Isto é mostrado na tabela 2. A ordem $n = 6$ do sistema é dada pelos valores singulares mais significativos da matriz L_{32} . Analisando-se os valores da tabela 2, todos os modelos tiveram um bom desempenho em termos de validação. Verifica-se que o tempo de processamento para a obtenção do modelo é menor

Table 2: Resultados numéricos do desempenho dos algoritmos.

Algoritmo	Tempo (s)	FIT (%)	MRSE (%)	MVAF (%)
N4sid1	0,328	100	$14,25 \cdot 10^{-13}$	100
N4sid2	0,328	100	$9,252 \cdot 10^{-13}$	100
MOESP	0,266	100	$4,946 \cdot 10^{-13}$	100
MON1	0,268	100	$9,881 \cdot 10^{-13}$	100
N4SID	0,844	100	0,0274	100
MPVerh	0,297	100	$1,88 \cdot 10^{-13}$	100
PEM	1,844	100	0,0274	100

para MOESP, MON1 e MPVerh. Com o objetivo de visualizar o desempenho do algoritmo proposto, optou-se pelo método MON1 para identificar o processo. A figura 4 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. Da figura 4 pode-se observar que a resposta

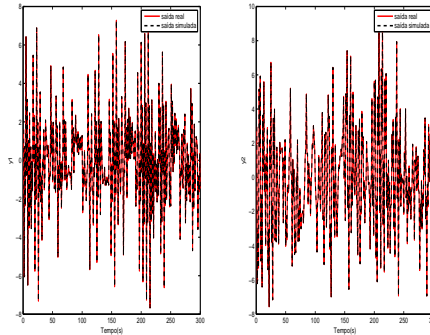


Figure 4: Comparação das respostas do processo real (linha contínua) versus modelo (linha tracejada).

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. As matrizes do modelo obtido pelo algoritmo MON1 são dadas por:

$$A = \begin{pmatrix} -0,8458 & -0,3721 & -0,0488 & -0,1807 & 0,0420 & -0,1142 \\ -0,2352 & 0,4636 & 0,7839 & 0,2699 & 0,0919 & 0,0325 \\ 0,0394 & -0,7524 & 0,3641 & 0,4447 & -0,1109 & 0,0103 \\ 0,0525 & 0,0675 & -0,0113 & 0,3321 & -0,1097 & -0,3361 \\ 0,0000 & 0,0000 & 0,0000 & -0,0001 & 1,0018 & 0,0082 \\ -0,0000 & -0,0000 & -0,0000 & 0,0000 & -0,0004 & 0,9982 \end{pmatrix}$$

$$B = \begin{pmatrix} -0,1814 & 3,2799 \\ 0,1236 & 0,3198 \\ -0,0525 & -1,5857 \\ -1,0032 & 0,5198 \\ -0,0001 & 0,0001 \\ 0,0000 & -0,0000 \end{pmatrix} \quad D = \begin{pmatrix} -0,0000 & 2,1346 \\ -0,5508 & 0,5446 \end{pmatrix}$$

$$C = \begin{pmatrix} -0,3505 & 0,2081 & 0,1006 & -0,3232 & -0,2805 & 0,0406 \\ -0,3448 & 0,2530 & -0,4933 & 0,6984 & -0,0306 & 0,0914 \end{pmatrix}.$$

A coleta de dados do processo real está sempre afetada por ruído. Sistemas puramente determinísticos têm interesse apenas acadêmico. Então se agregou ruído aos dados de entrada e saída. Neste caso, deseja-se saber como funciona o método de identificação por subespaços para o modelo M na presença de ruído.

3.3. Identificação com ruído de baixa (3%) e alta (80%) intensidade para o modelo MIMO M

A forma como foram coletados os dados com ruídos de baixa (3%), $\lambda = 0,03$ e alta intensidade (80%), $\lambda = 0,8$, para o modelo M é dado pelo seguinte algoritmo:

Urt=[ut+ λ *std(ut)*randn(1000,2)];

Yrt=[yt+ λ *std(yt)*randn(1000,2)].

Observa-se que o ruído não afeta a planta, somente afeta os sinais coletados de entradas e saídas. Quando o sistema é perturbado por ruído, muitas vezes encontrar a ordem n do sistema a partir da decomposição de valores singulares não é fácil. Uma alternativa é dada pelo critério de Akaike, para maiores detalhes ver [1]. Neste caso, $n = 4$ foi obtido pelo critério de Akaike. Os resultados da simulação são mostrados nas tabelas 3 e 4, para os casos de baixa e alta intensidade do ruído, respectivamente.

Table 3: Resultados numéricos do desempenho dos algoritmos com ruído de baixa intensidade (3%).

Algoritmo	Tempo (s)	FIT (%)	MRSE (%)	MVAF (%)
N4sid1	0,266	95,24	4,75	99,79
N4sid2	0,250	95,21	4,78	99,79
MOESP	0,266	95,18	4,82	99,78
MON1	0,328	95,32	4,67	99,80
N4SID	1,328	95,21	4,79	99,79
MPVerh	0,281	95,15	4,84	99,78
PEM	1,734	95,22	4,78	99,79

Analisando-se os valores da tabela 3, todos os modelos tiveram bom desempenho em termos de validação. Verifica-se que o tempo de processamento é menor para o algoritmo N4sid2. Analisando-se os valores da tabela 4, o modelo N4sid2 é melhor em termos de validação, verifica-se que o tempo de processamento para obtenção do modelo é menor para o algoritmo MPVerh. Dos resultados da tabelas 3 pode-se

Table 4: Resultados numéricos do desempenho dos algoritmos com ruído de alta intensidade (80%).

Algoritmo	Tempo (s)	FIT (%)	MRSE (%)	MVAF (%)
N4sid1	0,422	21,64	78,30	38,59
N4sid2	0,406	21,90	78,04	38,99
MOESP	0,265	16,60	83,31	30,52
MON1	0,516	21,44	78,50	38,27
N4SID	0,735	21,08	78,80	37,69
MPVerh	0,203	21,37	78,58	38,15
PEM	1,828	21,03	78,91	37,62

observar que os índices de desempenho para todos os algoritmos são quase iguais, o mesmo ocorrendo na tabela 4. A identificação determinística é um caso ideal, no entanto o algoritmo MON1 pode ser ampliado para a identificação de sistemas determinístico-estocásticos, para maiores detalhes ver [4] e [3].

4. Conclusão

O algoritmo MON1 não é um modelo novo, mas sim um modelo alternativo, pois usa as técnicas dos métodos MOESP e N4SID. No método alternativo MON1 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, sete algoritmos de identificação foram aplicados a dois processos em estudo e seus resultados foram comparados via validação cruzada, usando três tipos de critério. O modelo PEM foi o mais lento nos sete casos de identificação.

Abstract. This paper presents an overview of the subspace identification problem in open loop. There are several algorithms that solve this problem (MOESP, DSR, N4SID, CVA). Based on the methods MOESP and N4SID the authors propose an alternative algorithm to identify deterministic systems operating in open loop. Two simulated process, one SISO and one MIMO are used to show the performance of this algorithm.

Keywords. Subspace identification, identification methods, deterministic identification.

References

- [1] H. Akaike, Information theory and an extension of the maximum likelihood principle. In: Second International Symposium on Information Theory, Budapest, Hungary. B.N. Petrov, F. Csaki (Eds.), pp. 267-281, 1973.

- [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] S.D.M. Borjas, "Estudo da identificação por subespaços em malha aberta e fechada e proposta de novos algoritmos", Tese de Doutorado, POLI, USP, São Paulo, SP, 2009.
- [4] 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.*, **12**, No. 3 (2011), 183-194.
- [5] W. Favoreel, B. De Moor, P. Van Overschee, Subspace state space system identification for industrial processes. *Journal of Process Control*, **10**, No.3 (2000), 149-155.
- [6] B. De Moor, P. Van Overschee, W. Favoreel, Algorithms for subspace state space system identification - an overview. In B. Datta (Ed.), Applied and computational control, signal and circuits, **1**, pp. 247-311. Birkhauser: Boston (Chapter 6), 1999.
- [7] 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.
- [8] P. Van Overschee B. De Moor, "Subspace Identification for Linear Systems: Theory, Implementation, Applications", Kluwer Academic Publishers, Dordrecht, 1996.
- [9] 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.
- [10] 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.
- [11] M. Verhaegen, Application of a subspace model identification technique to identify LTI systems operating in closed loop, *Automatica*, **29**, No 4 (1993), 1027-1040.