

# NOI APLICAȚII DE CALCUL PENTRU IDENTIFICAREA SISTEMELOR

Vasile SIMA

vsima@ici.ro

Florin HARTESCU

flory@ici.ro

Alexandru STANCIU

alex@ici.ro

Institutul Național de Cercetare-Dezvoltare în Informatică - ICI București

**Rezumat:** Se prezintă un set de aplicații pentru identificarea unor sisteme liniare multivariabile. Algoritmii încorporați utilizează tehnici bazate pe subspații (MOESP, N4SID, sau combinația acestora) pentru a determina un model standard cu timp discret, în spațiul stărilor, cât și, optional, matricele de covarianță și matricea predictorului, folosind traiectoriile de intrare și ieșire (I/O). Pentru flexibilitate, sunt oferite aplicații diferite pentru calculul factorului superior triunghiular al unei matrice bloc-Hankel-bloc a datelor I/O (utilizând algoritmi de factorizare QR standard sau rapizi), pentru calculul matricelor sistemului, a matricei predictorului, cât și pentru estimarea stării inițiale a sistemului și simularea acestuia. Aplicațiile sunt încapsulate în containere Docker ce sunt gestionate cu ajutorul platformei Kubernetes. Această soluție asigură o mai bună flexibilitate, securitate sporită și execuție rapidă. Serviciile în curs de implementare sunt parte a unei platforme cloud deschise pentru aplicații de conducere a proceselor.

**Cuvinte cheie:** algoritmi de identificare, sisteme liniare multivariabile, algoritmi numerici, estimarea parametrilor, metode de subspații, descompunerea după valorile singulare, software.

**Abstract:** A set of applications for identification of linear multivariable systems is presented. The incorporated algorithms use subspace-based techniques (MOESP, N4SID, or their combination) to find a standard discrete-time state-space description, and optionally, the covariance matrices and predictor gain matrix, using input and output (I/O) trajectories. For flexibility, separate applications are offered for computing the processed upper triangular factor of the block-Hankel-block matrix of I/O data (using fast or standard QR factorization algorithms), for computing the system matrices, predictor gain matrix, for estimating the initial state of the system, and for its simulation. The applications are encapsulated in Docker containers which are managed by the Kubernetes platform. This ensures greater flexibility, enhanced security, and fast execution. The services to be implemented are part of a cloud-based open platform for process control applications.

**Keywords:** identification algorithms, linear multivariable systems, numerical algorithms, parameter estimation, subspace methods, singular value decomposition, software.

## 1. Introducere

Tehnicile avansate de conducere a sistemelor dinamice necesită existența unor modele matematice adecvate. De aceea, în practică este foarte important să se dispună de instrumente performante de identificare a sistemelor, capabile să determine modele bazate pe înregistrările existente, sau special generate, ale variabilelor cheie ale sistemelor.

Se consideră modelul liniar invariant în timp (LIT) discret, descris în spațiul stărilor, în reprezentarea în inovații,

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k + Ke_k \\ y_k &= Cx_k + Du_k + e_k,\end{aligned}\quad (1)$$

unde  $x_k \in \mathbf{R}^n$  este vectorul de stare  $n$ -dimensional la momentul  $k$ ,  $u_k \in \mathbf{R}^m$  este vectorul de intrare,  $y_k \in \mathbf{R}^l$  este vectorul de ieșire,  $A, B, C$  și  $D$  sunt matrice reale, perechea de matrice  $(A, C)$  este presupusă observabilă,

$\{e_k\}$  este o secvență de zgomot alb, necorelată cu  $\{u_k\}$  și cu starea inițială a sistemului,  $x_1$ , iar  $K$  este matricea predictorului (numită uneori matricea de amplificare Kalman).

În problemele de identificare a sistemelor, ordinul sistemului,  $n$ , și cvadruplul de matrice  $(A, B, C, D)$  trebuie determinate (modulo o transformare de similaritate a sistemului) utilizând secvențele de date de intrare și ieșire,  $\{u_k\}$  și  $\{y_k\}$ ,  $k = 1 : t$  (adică, pentru  $k$  luând valori întregi de la 1 la  $t$ ). În plus, trebuie adesea determinată matricea  $K$  din (1).

Au fost propuse trei abordări bazate pe subspații: MOESP (Multivariable Output Error state SPace) — [1]–[4], N4SID (Numerical algorithm for Subspace State Space System IDentification) — [5,6] și CVA (Canonical Variate Analysis) — [7]. Principala caracteristică a clasei de tehnici MOESP este determinarea unei matrice de observabilitate extinse a părții deterministe a modelului (1). Principala caracteristică a clasei de tehnici N4SID este determinarea unei secvențe de stări

estimate ale sistemului LIT prin intersecția unor (sau proiecția pe niște) spații linie ale unor matrice de tip Hankel construite din datele de intrare-ieșire (I/O) “trecute” și “viitoare”. Tehnicile CVA utilizează, de asemenea, secvența de stare, într-un cadru statistic.

Abordările bazate pe subspații pentru identificarea sistemelor sunt atractive din câteva motive: sunt estimate direct modele în spațiul de stare; nu sunt necesare parametrizări; sunt utilizate doar instrumente robuste de algebră liniară, cum ar fi descompunerea QR și descompunerea după valorile singulare (SVD—singular value decomposition); trebuie ales doar un parametru,  $s$ .

Algoritmii de identificare bazați pe tehnici de subspații pornesc prin construirea, explicită sau implicită, a unei *matrice bloc-Hankel-bloc*  $H$  prin concatenarea unor matrice bloc-Hankel, utilizând (o parte din) datele de I/O disponibile. Pentru tehnica “MOESP cu intrări și ieșiri trecute”,  $H$  este dată de

$$H = [U_{p+1,p+s,N+p}^T \quad U_{1,p,N}^T \quad Y_{1,q+s,N}^T], \quad (2)$$

$H \in \mathbf{R}^{N \times (mp+lq+(m+l)s)}$ , unde  $N = t - \min(p,q) - s + 1$  (uzual  $N \gg mp+lq+(m+l)s$ ),

$$U_{a,b,c} = \begin{bmatrix} u_a & u_{a+1} & u_{a+2} & \cdots & u_c \\ u_{a+1} & u_{a+2} & u_{a+3} & \cdots & u_{c+1} \\ \vdots & \vdots & \vdots & & \vdots \\ u_b & u_{b+1} & u_{b+2} & \cdots & u_{c+b-a} \end{bmatrix}, \quad (3)$$

și similar pentru  $Y_{a,b,c}$ . Pentru N4SID, primele două bloc coloane ale lui  $H$  sunt interschimbate, astfel încât ele pot fi rescrise ca

$U_{1,p+s,N}^T$ . Submatricele  $U_{1,p,N}$  și  $Y_{1,q,N}$  sunt numite părțile de intrare și, respectiv, ieșire “trecute” (past), în timp ce submatricele  $U_{p+1,p+s,N+p}$  și  $Y_{q+1,q+s,N+q}$  sunt numite părțile “viitoare” (future). Ultimele versiuni MATLAB — marcă înregistrată a companiei The MathWorks, Inc. — ale funcției `n4sid` din pachetul de programe de identificare [8] utilizează diferite “orizonturi de predicție”  $s$ ,  $p$  și  $q$ , ale căror valori implicite sunt alese utilizând criteriul de informație Akaike (AIC). Dar algoritmii MOESP și N4SID standard utilizează  $p = q = s$ , unde  $s$  denotă “numărul de blocuri linie”, iar  $s$  satisface uzual  $s \geq n$ . Teoria MOESP presupune că  $s > n$ . În practică,  $s$  este adesea ales mai mare decât ordinul sistemului

$n$ , de pildă,  $s = 2n$ . O valoare mare a lui  $s$  produce uzual rezultate mai precise. Este de asemenea posibil să se aleagă orizonturile de predicție pentru datele trecute mai mici decât  $n$ , de exemplu,  $p = q > n/(m+l)$ . Totuși, cu valori mai mici, se obțin în mod normal rezultate mai proaste decât cele pentru valori mai mari. De aceea, pentru simplitate, se va presupune că  $p = q = s$ , deci

$$H = [U_{s+1,2s,N+s}^T \quad U_{1,s,N}^T \quad Y_{1,2s,N}^T] \quad (\text{MOESP}),$$

$$H = [U_{1,2s,N}^T \quad Y_{1,2s,N}^T] \quad (\text{N4SID}) \quad (4)$$

$H \in \mathbf{R}^{N \times 2(m+l)s}$ , unde  $N = t - 2s + 1$ , iar  $U_{\dots}$  și  $Y_{\dots}$  sunt matrice bloc-Hankel, definite cu formule (3). Primul pas al unei proceduri de identificare bazate pe subspații este efectuarea unei “compresii a datelor” prin calculul unui factor superior triunghiular  $R$  din factorizarea QR a matricei  $H$  din (4),  $H = QR$ ; matricea  $Q \in \mathbf{R}^{N \times 2(m+l)s}$ , satisfăcând  $Q^T Q = I_{2(m+l)s}$ , nu este necesară în calculele următoare. Părți din  $R$  sunt utilizate în continuare pentru a estima  $n$  și matricele sistemului.

Trebuie subliniat că multe rezultate teoretice privind, de pildă, consistența și normalitatea estimațiilor matricelor sistemului au loc asimptotic, pentru  $t \rightarrow \infty$ . De aceea, algoritmii trebuie să fie capabili de a trata volume mari de date, dacă acestea sunt disponibile. Algoritmii bazat pe factorizarea QR standard poate fi prea costisitor, deoarece  $H$  poate avea dimensiuni foarte mari. Algoritmi rapizi și fiabili pentru rezolvarea problemei de identificare a sistemelor au fost propuși în [9]–[13] și au fost implementați în biblioteca SLICOT – Subroutine Library In COntrol Theory [14] (<http://www.slicot.org>) și în pachetul de identificare asociat (în MATLAB), bazat pe tehnici de subspații [15–17]. SLICOT se bazează pe reputatul pachet de programe de algebră liniară LAPACK [18] și pe colecțiile BLAS (Basic Linear Algebra Subprograms), beneficiind astfel de caracteristicile avansate ale arhitecturilor de calculatoare moderne.

Pe baza experienței anterioare, au fost elaborate aplicații de identificare operând sub sisteme de operare Windows sau Linux (Ubuntu), cu arhitecturi pe 32 sau 64 biți. Este posibil să se utilizeze fie abordarea MOESP, fie N4SID, fie o abordare combinată: se estimează matricele  $A$  și  $C$  folosind abordarea MOESP și matricele  $B$  și  $D$  folosind abordarea N4SID. Se

pot prelucra secvențial seturi de date I/O multiple și există mai multe opțiuni de calcul. Aplicațiile sunt încapsulate în containere Docker (<https://docker.io>) ce sunt gestionate cu ajutorul platformei Kubernetes (<https://kubernetes.io>). Această soluție asigură o mai mare flexibilitate, securitate sporită și execuție rapidă. Serviciile în curs de implementare sunt parte a unei platforme cloud deschise pentru sisteme de conducere a proceselor.

## 2. Prelucrarea eficientă a datelor pentru identificare bazată pe subspații

Deoarece matricea  $H$  este puternic structurată, prelucrarea eficientă a datelor este posibilă prin utilizarea structurii problemei. Sunt folosite două astfel de tehnici, bazate pe algoritmi rapizi de factorizare Cholesky și factorizare QR, care exploatează structura specială a matricei  $H$  din (4).

Algoritmul de factorizare Cholesky pentru compresia datelor ([12,13] și referințele citate) construiește întâi matricea de intercorelație  $W=H^T H$ , iar apoi factorizează  $W$ , presupunând că este pozitiv definită (ceea ce este aproape întotdeauna adevărat în practică, datorită prezenței zgomotelor). Pentru prelucrarea standard utilizând abordarea N4SID, matricele bloc-Hankel corespunzătoare intrărilor și ieșirilor sunt  $H_u = U_{1,2s,N}^T$  și  $H_y = Y_{1,2s,N}^T$ , iar  $H=[H_u H_y]$ . Definițiile de mai sus pot fi extinse la seturi multiple de date. De fapt, presupunând că ultimul set de prelucrat este definit de  $H$ , atunci  $W = \tilde{W} + H^T H$ , unde  $\tilde{W}$  corespunde seturilor de date deja prelucrate. Evident,  $\tilde{W}=0$ , dacă se prelucrează primul (sau singurul) set de date. Se pot scrie formule economice, exploatând structura, pentru calculul matricei bloc simetrice  $W$  [12, 13]. Într-adevăr, pentru abordarea N4SID, se definesc  $W_{uu} = \tilde{W}_{uu} + H_u^T H_u$ ,  $W_{uy} = \tilde{W}_{uy} + H_u^T H_y$  și  $W_{yy} = \tilde{W}_{yy} + H_y^T H_y$ , unde fiecare bloc constă din  $2s \times 2s$  submatrice de dimensiuni  $m \times m$ ,  $m \times l$  și, respectiv,  $l \times l$ . Notând  $W_{uu}^{i,j}$  submatricea  $(i,j)$  a lui  $W_{uu}$  ( $j=1:2s$ ,  $i=1:2s$ ) și similar pentru  $W_{uy}$  și  $W_{yy}$ , atunci prima bloc linie a matricei  $W_{uu}$  este dată de formula (5), iar liniile următoare se

determină recursiv, folosind (5) și (6),

$$W_{uu}^{1,j} = \tilde{W}_{uu}^{1,j} + u_1 u_j^T + u_2 u_{j+1}^T + \dots + u_N u_{j+N-1}^T, j = 1:2s \quad (5)$$

$$W_{uu}^{i+1,j+1} = \tilde{W}_{uu}^{i+1,j+1} - \tilde{W}_{uu}^{i,j} + W_{uu}^{i,j} + u_{i+N} u_{j+N}^T - u_i u_j^T, j = 1:2s-1, i = 1:j \quad (6)$$

Similar, se calculează  $W_{yy}^{i,j}$ , înlocuindu  $u$  cu  $y$ , cât și  $W_{uy}^{i,j}$ , folosind diade  $uy^T$ .

Fie acum  $W$  matricea calculată așa cum s-a prezentat mai sus, rescrisă sub forma,

$$W := \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{12}^T & W_{22} & W_{23} \\ W_{13}^T & W_{23}^T & W_{yy} \end{bmatrix}, \quad (7)$$

cu  $W_{11}$ ,  $W_{12}$ ,  $W_{22} \in \mathbf{R}^{ms \times ms}$ ,  $W_{13}$ ,  $W_{23} \in \mathbf{R}^{ms \times 2ls}$ . Atunci, matricea corespunzătoare pentru abordarea MOESP,  $W_M$ , este dată de formula

$$W_M := \begin{bmatrix} W_{22} & W_{12}^T & W_{23} \\ W_{12} & W_{11} & W_{13} \\ W_{23}^T & W_{13}^T & W_{yy} \end{bmatrix}. \quad (8)$$

Evident, toate submatricele, exceptând  $8s$ , sunt calculate utilizând formule de actualizare, de forma (6), care sunt mult mai economice (pentru  $N$  mare) decât (5). În final, se aplică algoritmul de factorizare Cholesky matricei calculate  $W$  (sau  $W_M$ ). În acele cazuri rare în care acest algoritm rapid eșuează, se utilizează automat factorizarea QR standard.

Se consideră acum algoritmul rapid de factorizare QR pentru abordarea N4SID. Se definește matricea de deplasare  $Z = \text{diag}(Z_u, Z_y)$ , unde  $Z_u$  ( $Z_y$ ), este o matrice bloc  $2s \times 2s$  cu blocuri  $m \times m$  ( $l \times l$ ), toate zero exceptând blocurile identitate de pe supradiagonală,

$$Z_u = \begin{bmatrix} 0_m & I_m & 0_m & 0_m \\ 0_m & 0_m & \ddots & \vdots \\ \vdots & \ddots & \ddots & I_m \\ 0_m & 0_m & \dots & 0_m \end{bmatrix},$$

$$Z_y = \begin{bmatrix} 0_l & I_l & 0_l & 0_l \\ 0_l & 0_l & \ddots & \vdots \\ \vdots & \ddots & \ddots & I_l \\ 0_l & 0_l & \dots & 0_l \end{bmatrix},$$

În [13] s-a demonstrat că matricea simetrică  $\nabla W = W - Z^T W Z$ , numită *deplasarea* lui  $W$ , are o factorizare de rang mic, anume  $\nabla W = G^T \Sigma G$ , cu  $\Sigma = \text{diag}(I_p, -I_q)$ , unde  $p=q=m+l+1$ , deci  $\nabla W$  are rangul cel mult  $2(m+l+1)$ . În consecință,  $\nabla W$  poate fi scrisă în forma factorizată menționată, unde  $G \in \mathbf{R}^{2(m+l+1) \times 2(m+l)s}$ . Matricea  $G$  este numită *generatorul* lui  $W$ . Sunt disponibile tehnici eficiente pentru a construi generatorii. În calculele practice, se utilizează *algoritmul Schur generalizat* pentru a găsi toate liniile factorului Cholesky al lui  $W$ ,  $W = \tilde{W} + H^T H$ . Sunt implicate transformări Householder și rotații hiperbolice [11].

Pentru abordarea MOESP, se utilizează același algoritm ca și pentru N4SID, dar primele două bloc coloane ale factorului superior triunghiular rezultat  $R$  sunt interschimbate și apoi retriunghiularizate, exploatând structura.

Opțional, prelucrarea datelor I/O pentru determinarea matricei  $R$  poate fi efectuată secvențial, pe seturi de date. Matricele  $A$  și  $C$  sunt obținute folosind o estimare a unei matrice de observabilitate extinsă,  $\Gamma_s$ , calculată, la rândul ei, din factorii  $U$  și  $\Sigma$  ai descompunerii după valorile singulare a unei matrice construită din  $R$  (a se vedea, de pildă, [13] și referințele citate). Se utilizează proprietatea de invarianță la deplasare a matricei  $\Gamma_s$ . (Abordarea N4SID folosește proiecția oblică  $O_s$  a ieșirilor „viitoare” peste intrările și ieșirile „trecute”. Ordinul sistemului este rangul matricei  $O_s$ .) Se folosesc apoi un algoritm QR eficient, exploatând structura, și un algoritm bazat pe produsul Kronecker, pentru calcularea matricelor  $B$  și  $D$  [13]. Matricele de covarianță sunt determinate folosind reziduul unei probleme de tipul celor mai mici pătrate (CMMP). Apoi, matricea predictorului,  $K$ , se obține prin rezolvarea unei ecuații algebrice Riccati discrete pentru problema de filtrare corespunzătoare. Starea inițială se calculează, de asemenea, rezolvând o problemă CMMP având ca date matricele sistemului și traiectoriile de intrare și ieșire. Detalii sunt date, de exemplu, în [9, 10, 12, 13] și în referințele citate acolo.

### 3. Aplicațiile de calcul pentru identificare

Au fost elaborate patru aplicații pentru identificarea sistemelor LIT multivariabile discrete. Prima aplicație, denumită *corder*, determină factorul superior triunghiular al matricei bloc-Hankel-bloc  $H$  a datelor I/O, utilizând fie algoritmul de factorizare QR standard, fie unul dintre algoritmi rapizi prezentați anterior. Datele de I/O pot fi furnizate opțional secvențial, pe seturi, independente sau nu. Aplicația determină factorul  $R$  și îl prelucrează, în funcție de opțiunile specificate, pentru a fi utilizat în aplicația următoare. Folosind valorile singulare, se propune o valoare adecvată a ordinului  $n$  al modelului.

A doua aplicație, *csident*, utilizează factorul  $R$  (prelucrat) la calculul matricelor modelului. Opțional, se determină, de asemenea, matricele de covarianță și matricea predictorului,  $K$ . Se poate specifica orice ordin  $n < s$ , pentru același factor  $R$ , ceea ce permite efectuarea unor experimente multiple.

A treia aplicație — *cfindx0*, folosită opțional, calculează o estimare  $\hat{x}_1$  a stării inițiale  $x_1$ , folosind modelul calculat și traiectoriile de intrare și ieșire.

A patra aplicație, *csimul*, calculează traiectoria ieșirii modelului estimat, cu sau fără predictor, cu o stare inițială specificată (de pildă,  $\hat{x}_1$ ) și determină unele metrici pentru evaluarea calității modelului, cum ar fi eroarea relativă sau VAF — “Variance Accounted For” — pentru fiecare variabilă de ieșire. Cu cât o valoare VAF este mai aproape de 1, cu atât traiectoria variabilei de ieșire estimate asociate este mai precisă. Modelul estimat se poate scrie sub forma

$$\begin{aligned}\hat{x}_{k+1} &= \hat{A} \hat{x}_k + \hat{B} u_k + K(\hat{y}_k - y_k), \\ \hat{y}_k &= \hat{C} \hat{x}_k + \hat{D} u_k, \quad k = 1:t,\end{aligned}$$

unde  $\hat{A}$ ,  $\hat{B}$ ,  $\hat{C}$ ,  $\hat{D}$  și  $K$  sunt matricele calculate de *csident*. Dacă nu s-a optat pentru determinarea matricei  $K$ , atunci se folosește  $K=0$ .

Aplicațiile implementate citesc opțiunile și alți parametri necesari execuției, inclusiv numele fișierelor care trebuie citite sau scrise, dintr-un alt fișier cu nume și structură predefinite. Aceste fișiere „machetă” sunt generate automat utilizând o interfață Web, prin completarea informației cerute în câmpurile corespunzătoare. Un exemplu este dat în Figura 1 pentru aplicația care calculează

matrice și indicii elementelor acestora, pentru valoarea 2, sau doar a valorilor numerice, pentru valoarea 3 a opțiunii. În primul caz, rezultatele pot fi inspectate și înțelese de utilizator, sau transferate ușor unui alt mediu de calcul, de pildă, MATLAB. În ultimul caz, rezultatele pot fi folosite direct de aplicația următoare din fluxul de lucru.

Parameter	Value
Output file name (give it if Unit number is not 0 or 6)	'order_evaporator.out'
Unit number for the output device (not 11 or 12)	13
Method METH to use ('M': MOESP, 'N': N4SID)	'M'
Algorithm ALG to use ('C': Cholesky, 'F': Fast QR, or 'Q': QR);	'C'
Processing option BATCH ('O': One, 'F': First, 'T': Intermediate, or 'L': Last)	'O'
Continuation option CONCT ('N': No, or 'C': Continuation)	'N'
Value for the number of block rows, S (S > N)	10
Value for the number of samples per data block, T	6305
Value for the number of system inputs, M (M >= 0)	3
Value for the number of system outputs, P (P > 0)	3
File containing the input/output data trajectories	'evaporator.dat'
Values for TOL1 and TOL2 tolerance parameters	0.0 - 1.0
Printing option (0: No matrices, 1: I/O, 2: Detailed O, 3: Compressed O)	3

\* All fields are compulsory

**Figura 1. Macheta pentru specificarea opțiunilor și parametrilor aplicației de determinare a ordinului modelului**

ordinul sistemului și factorul superior triunghiular prelucrat al factorizării QR a matricei  $H$ . Aplicația citește fișierul conținând traiectoriile de intrare/ieșire, al cărui nume este specificat în antepenultimul câmp al machetei corespunzătoare. Rezultatele sunt scrise fie la un terminal, fie într-un fișier al cărui nume este indicat în primul câmp al machetei. Modul de afișare este specificat în ultimul câmp. Valoarea 1 a opțiunii de afișare determină listarea atât a datelor de intrare, cât și a rezultatelor calculate. Valorile 2 sau 3 implică afișarea doar a rezultatelor, cu antete, nume

Implementarea aplicațiilor utilizând servicii cloud este proiectată pe baza platformei Kubernetes ce are rolul de orchestrare a execuției containerelor Docker.

Aplicațiile prezentate anterior sunt încapsulate în cadrul unor containere ce sunt grupate într-un Pod — concept specific Kubernetes, astfel încât să poată comunica între ele în mod direct. Pentru transferul datelor de intrare și de ieșire se utilizează o interfață de tip REST (Representational State Transfer), cu scopul de a se crea un serviciu gestionat de către Kubernetes pentru întreg fluxul de lucru al aplicațiilor, care este prezentat în figura 2.

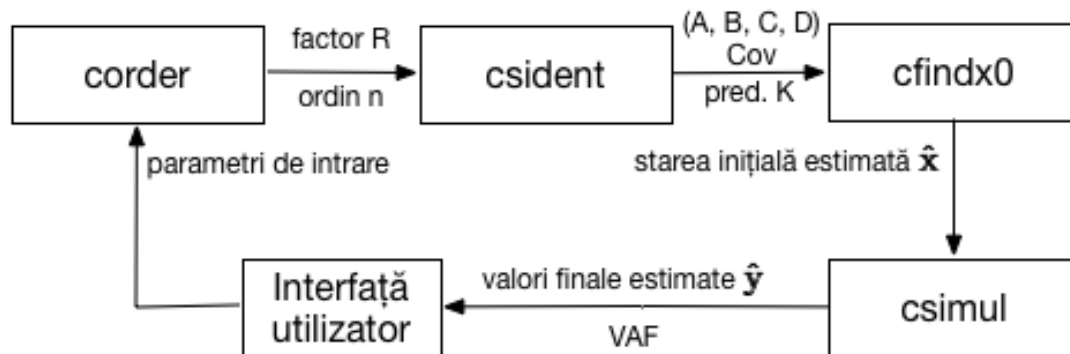


Figura 2. Fluxul de lucru al aplicațiilor de identificare

Aplicațiile de calcul pentru identificare sunt executate prin intermediul unui wrapper, care are rolul de a gestiona transferul datelor de intrare și de ieșire. Aceste programe wrapper trebuie să se conecteze la un broker de mesaje — Redis (<https://redis.io>) — pentru a putea trimite sau primi datele necesare. Adresa la care trebuie să se conecteze, canalul utilizat, precum și toate celelalte informații necesare pentru execuție sunt generate de către un program destinat a configura și a pune în execuție containerele Docker în cadrul platformei Kubernetes. Acest program folosește un fișier cu descrierea sistemului, în care sunt prezentate componentele, conexiunile între acestea și valorile parametrilor utilizați.

#### 4. Rezultate numerice

Aplicațiile elaborate au fost testate pe câteva exemple din colecția DAISY (Database for Identification of Systems) [19], disponibilă public la adresa Internet <http://home.esat.kuleuven.be/~smc/daisy/>. Colecția conține peste 20 seturi de date: procese industriale, sisteme mecanice, sisteme biomedicale, sisteme de mediu, sisteme termice etc. În continuare se prezintă unele rezultate obținute pentru exemplul unui cuptor de sticlă. Numeroase alte rezultate și comparații ale algoritmilor de identificare bazați pe tehnici de subspații au fost publicate, de pildă, în [12,13,16,20] și în referințele citate.

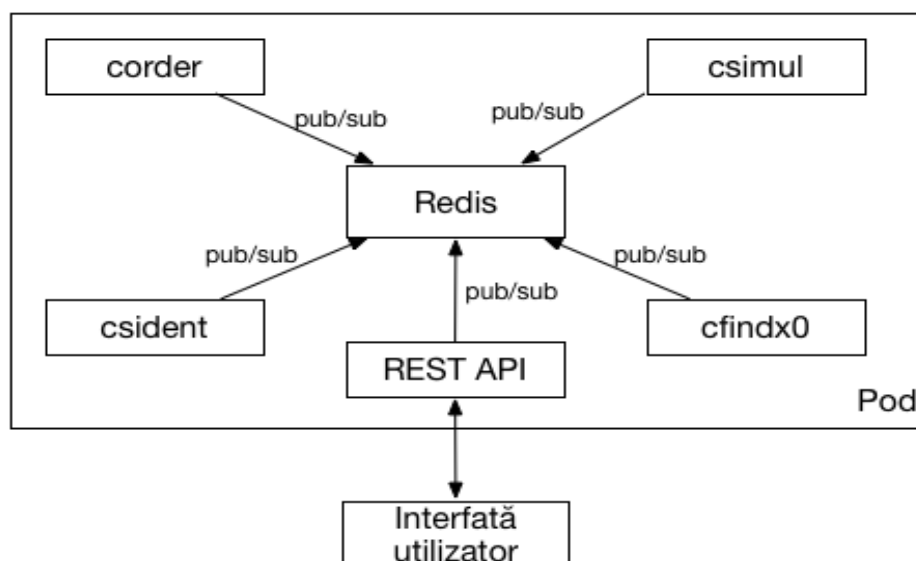


Figura 3. Conectarea aplicațiilor cu ajutorul unui broker de mesaje și execuția acestora în cadrul unui Pod Kubernetes

Datele disponibile pentru cuptorul de sticlă sunt 1247 eșantioane pentru 6 variabile de ieșire și 3 variabile de intrare. Variabilele de ieșire sunt temperaturile măsurate de șase senzori plasați într-o secțiune a cuptorului. Două dintre variabilele de intrare se referă la încălzire, iar a treia, la răcire.

S-au folosit toate cele patru aplicații menționate. S-a ales numărul de linii bloc ca fiind  $s = 10$  și numărul de variabile de stare (ordinul sistemului) ca fiind  $n = 5$ . S-a utilizat algoritmul Cholesky rapid. Cu o aproximație de trei cifre semnificative, polii sistemului determinat sunt  $0,662$ ,  $0,880$ ,  $0,940$ ,  $0,983 \pm 0,0804j$ , unde  $j$  este unitatea pur imaginară. Așadar, sistemul estimat este stabil.

Polii sistemului cu predictor sunt  $0,634 \pm 0,171j$ ,  $0,566 \pm 0,157j$ ,  $0,666$ . În Figura 4 sunt reprezentate traiectoriile variabilelor de ieșire,  $y_i$ , și a ieșirilor estimate,  $\hat{y}_i$ , fără predictor (notate  $Ye_i$ , în coloana din stânga) și cu predictor (notate  $YeK_i$ , în coloana din dreapta). Traiectoriile originale, date, sunt reprezentate cu linie roșie, iar traiectoriile estimate cu modelul obținut sunt reprezentate cu linie albastră. Deasupra fiecărei reprezentări grafice se indică o măsură a preciziei obținute, anume eroarea relativă, definită de formula  $\frac{\|y_i - \hat{y}_i\|}{\max(1, \|y_i\|)}$ ,  $i = 1:6$ . Evident, folosirea predictorului îmbunătățește precizia.

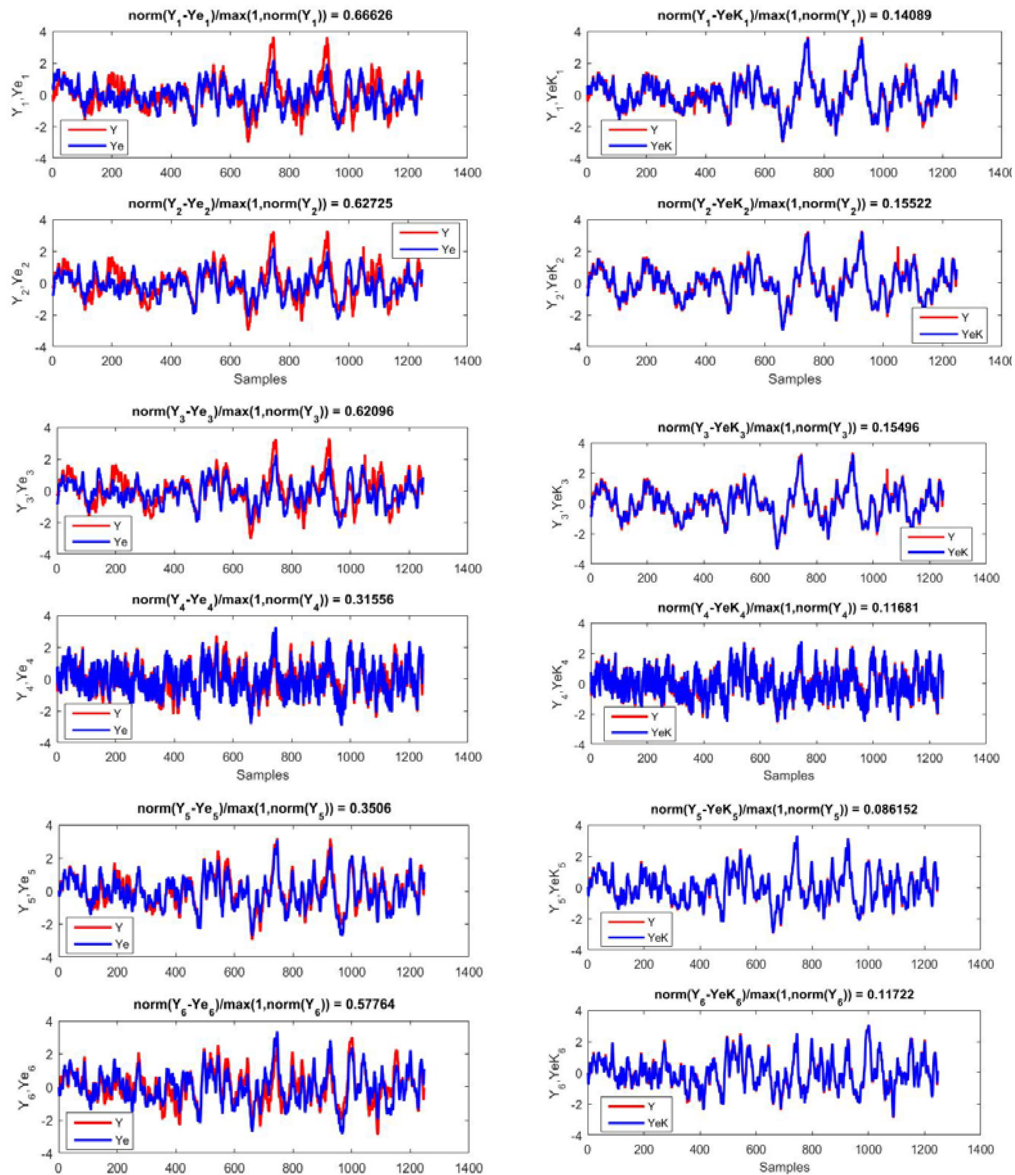


Figura 4. Traiectoriile ieșirilor,  $y_i$ , și a ieșirilor estimate,  $\hat{y}_i$ , fără predictor (notate  $Ye_i$ , în coloana din stânga) și cu predictor (notate  $YeK_i$ , în coloana din dreapta),  $i=1:6$

## 5. Concluzii

Articolul prezintă succint algoritmi utilizați pentru implementarea unor aplicații pentru identificarea modelelor matematice ale unor sisteme dinamice liniare. Modelele sunt cu timp discret, multivariabile, invariante în timp și folosind variabile de stare. Datele de intrare sunt traiectoriile variabilelor de ieșire și de intrare. Diverse opțiuni permit alegerea unor algoritmi rapizi și a unor parametri de calcul. Opțiunile sunt specificate printr-o interfață Web. Aplicațiile sunt încapsulate în containere Docker și pot fi executate local sau într-un cloud. Se prezintă rezultatele obținute la identificarea unui cuptor de sticlă.

## Mulțumiri

Această lucrare a fost realizată cu suport parțial din Programul Național de Cercetare PNII, proiectul 257/2014: Arhitectură cloud pentru o bibliotecă deschisă de blocuri funcționale logice reutilizabile pentru sisteme optimizate – CALCULOS.

## BIBLIOGRAFIE

1. **VERHAEGEN, M.:** Subspace model identification. Part 3: Analysis of the ordinary output-error state-space model identification algorithm. *Int. J. Control*, 58(3):555-586, 1993.
2. **VERHAEGEN, M.:** Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30(1):61-74, 1994.
3. **VERHAEGEN, M.; DEWILDE P.:** Subspace model identification. Part 1: The output-error state-space model identification class of algorithms. *Int. J. Control*, 56(5): 1187-1210, 1992.
4. **VERHAEGEN, M.; DEWILDE, P.:** Subspace model identification. Part 2: Analysis of the elementary output-error state-space model identification algorithm. *Int. J. Control*, 56(5): 1211-1241, 1992.
5. **VAN OVERSCHEE, P.; DE MOOR, B.:** N4SID: Two subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75-93, 1994.
6. **VAN OVERSCHEE, P.; DE MOOR, B.:** Subspace Identification for Linear Systems : Theory - Implementation - Applications. Kluwer Academic Publishers, Boston/London/Dordrecht, 1996.
7. **LARIMORE, W.:** System identification, reduced order filtering and modeling via canonical variate analysis. In *Proceedings of the American Control Conference*, San Francisco, CA, USA, pp. 445-451, 1983.
8. **LJUNG, L.:** System Identification Toolbox. For Use with MATLAB. User's Guide. Version 5. The Math-Works, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, 2000.
9. **SIMA, V.:** Algorithms and LAPACK-based software for subspace identification. In *Proceedings of The 1996 IEEE International Symposium on Computer-Aided Control System Design*, September 15-18, 1996, Ritz-Carlton, Dearborn, Michigan, U.S.A., pp. 182-187, 1996.
10. **SIMA, V.:** Subspace-based algorithms for multivariable system identification. *Studies in Informatics and Control*, 5(4):335-344, 1996.
11. **MASTRONARDI, N.; KRESSNER, D.; SIMA, V.; VAN DOOREN, P.; VAN HUFFEL, S.:** A fast algorithm for subspace state-space system identification via exploitation of the displacement structure. *Journal of Comput. Appl. Math.*, 132(1):71-81, 2001.
12. **SIMA, V.; SIMA, D. M.:** High-performance algorithms for linear multivariable system identification. *Romanian Journal of Information Science and Technology*, 6(3-4):345-375, 2003.
13. **SIMA, V.; SIMA, D. M.; VAN HUFFEL, S.:** High-performance numerical algorithms and software for subspace-based linear multivariable system identification. *J. Comput. Appl. Math.*, 170(2):371-397, 2004.
14. **BENNER, P.; MEHRMANN, V.; SIMA, V.; VAN HUFFEL, S.; VARGA, A.:** SLICOT — A subroutine library in systems and control theory. In B. N. Datta,



- editor, Applied and Computational Control, Signals, and Circuits, volume 1, chapter 10, pp. 499-539. Birkhäuser, Boston, 1999.
15. **SIMA, V.; VAN HUFFEL, S.:** Efficient numerical algorithms and software for subspace-based system identification. In Proceedings of the 2000 IEEE International Conference on Control Applications and IEEE International Symposium on Computer-Aided Control Systems Design, September 25-27, 2000, Anchorage Hilton, Anchorage, Alaska, U.S.A., pp. 1-6. Wilson Center for Research & Technology, Xerox Corporation, 800 Phillips Rd. MS 128-56E, Webster, NY 14580, 2000. Omnipress.
  16. **SIMA, V.; VAN HUFFEL, S.:** Performance investigation of SLICOT system identification toolbox. In Proceedings of the European Control Conference, ECC 2001, 4-7 September, 2001, Seminario de Vilar, Porto, Portugal, pp. 3586-3591, 2001.
  17. **SIMA, V.; SIMA, D. M.; VAN HUFFEL, S.:** SLICOT system identification software and applications. In Proceedings of the 2002 IEEE International Conference on Control Applications and IEEE International Symposium on Computer Aided Control System Design, CCA/CACSD 2002, September 18-20, 2002, Scottish Exhibition and Conference Centre, Glasgow, Scotland, U.K., pp. 45-50. Omnipress, 2002.
  18. **ANDERSON, E.; BAI, Z.; BISCHOF, C.; BLACKFORD, S.; DEMMEL, J.; DONGARRA, J.; DU CROZ, J.; GREENBAUM, A.; HAMMARLING, S.; MCKENNEY, A.; SORENSEN, D.:** LAPACK Users' Guide: Third Edition. Software · Environments · Tools. SIAM, Philadelphia, 1999.
  19. **DE MOOR, B.; DE GERSEM, P.; DE SCHUTTER, B.; FAVOREEL, W.:** DAISY: A database for identification of systems. Journal A, Special Issue on CACSD (Computer Aided Control Systems Design), 38:4-5, 1997.
  20. **SIMA, V.:** Efficient data processing for subspace-based multivariable system identification. In Proceedings of IFAC Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP 2004), and IFAC Workshop on Periodic Control Systems (PSYCO 2004), August 30th - September 1st, 2004, "Pacifico Yokohama" Pacific Convention Plaza, Yokohama, Japan, pp. 871-876, 2004.