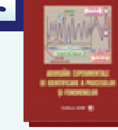


⓫ Identificarea parametrilor fizici

☞ Contextul de lucru

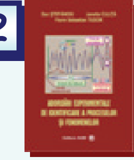
- Dacă un proces poate fi asociat cu un model de sistem liniar în timp continuu **de ordin suficient de mic** (**maxim 3**), atunci există posibilitatea identificării **parametrilor** acestui model **direct din date I/O**.

20 p

Parametri fizici ai procesului
(definiție ad hoc)

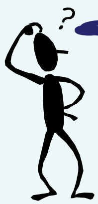
Strategii de identificare

Pe bază de modele I/O	**Pe bază de modele cu reprezentare pe stare**
① Se discretizează funcția de transfer a procesului.	① Se discretizează modelul cu reprezentare pe stare al procesului.
② Se determină relațiile analitice ale parametrilor fizici în funcție de parametrii modelului I/O discret obținut.	② Se determină relațiile analitice ale parametrilor fizici în funcție de parametrii modelului discret pe stare obținut.
③ Se identifică modelul I/O discret din date achiziționate la intrarea şi ieşirea procesului.	③ Se identifică modelul discret pe stare din date achiziționate la intrarea şi ieşirea procesului.
④ Se determină parametrii fizici din parametrii estimați ai modelului de identificare, cu ajutorul relațiilor de la pasul ②.	④ Se determină parametrii fizici din parametrii estimați ai modelului de identificare, cu ajutorul relațiilor de la pasul ②.



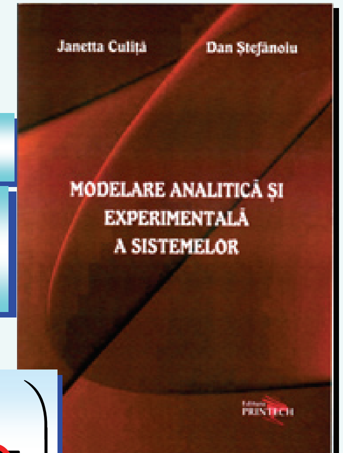
❶❶ Identificarea parametrilor fizici

☞ Contextul de lucru



Discretizare?

Prin tehnici descrise, de exemplu, în

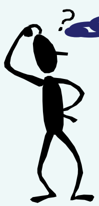


Teoretic, în cazul **funcțiilor de transfer**, se apelează la **Teorema Reziduurilor**.

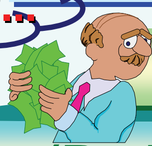
$$\mathrm{H}_d(z) \overset{\text{def}}{=} (z-1) \sum_{s \in \{\text{poli}\}} \mathrm{Rez}\left(\frac{\mathrm{H}(s)}{s} \frac{1}{z - e^{sT_s}} \right)$$

☜ Poate fi dificil!

perioada de eşantionare



Şi atunci?...



Două metode practice de discretizare

Euler (Padé)

$$z \cong 1 + sT_s$$

$$y(nT_s) = y[n] \quad \forall n \in \mathbb{N}$$

$$\dot{y}(t) \cong \frac{y[n+1] - y[n]}{T_s} \quad \forall t \in [nT_s, (n+1)T_s] \quad \forall n \in \mathbb{N}$$

← Derivată atribuită valorii la stînga.

Biliniară (Tustin)

$$s \cong \frac{2}{T_s} \frac{z-1}{z+1}$$

$$y\left(\frac{2n+1}{2} T_s\right) \overset{def}{=} \frac{y[n+1] + y[n]}{2} \quad \forall n \in \mathbb{N}$$

↑ Derivată atribuită valorii mediane.

☝ Regula de atribuire a derivatei nu are importanță prea mare în cazul modelelor **cu parametri constanți**, dar este esențială în cazul modelelor **cu parametri variabili**.

➊➊ Identificarea parametrilor fizici

☞ Contextul de lucru

Exemplul 1 Discretizarea Euler a modelului de ordin 1 cu parametri constanți

① Se pleacă de la reprezentarea Laplace.

$$TsY(s)+Y(s)=KU(s)$$

$$H(s)\overset{def}{=}\frac{Y(s)}{U(s)}=\frac{K}{1+Ts}$$

(funcția de transfer continuă)

② Se trece la reprezentarea Z.

$$H_d(z)\overset{def}{=}\frac{Y(z)}{U(z)}=\frac{KT_s}{T(z-1)+T_s}=\frac{KT_s z^{-1}}{T+\left(T_s-T\right)z^{-1}}$$

③ Se revine la domeniul timp discret.

$$y[n]+ay[n-1]=bu[n-1] \quad \forall n\in\mathbb{N}$$

• Cu a b în funcție de K T T_s.

Exemplul 2 Discretizarea Euler a modelului de ordin 1 cu parametri variabili

① Se particularizează ecuația diferențială pentru momentele de eşantionare.

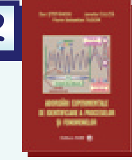
$$T\left(nT_s\right)\dot{y}\left(nT_s\right)+y\left(nT_s\right)=K\left(nT_s\right)u\left(nT_s\right) \quad \forall n\in\mathbb{N}$$

② Se aplică regula de atribuire a derivatei.

$$T[n]\frac{y[n+1]-y[n]}{T_s}+y[n]=K[n]u[n] \quad \forall n\in\mathbb{N}$$

③ Se exprimă ecuația cu diferențe corespunzătoare.

$$\frac{T[n]}{T_s}y[n+1]+\left(1-\frac{T[n]}{T_s}\right)y[n]=K[n]u[n] \quad \forall n\in\mathbb{N}^*$$



❶❶ Identificarea parametrilor fizici

☞ Contextul de lucru

- Un proces care poate fi asociat unui model destul de simplu (în timp continuu) este **motorul de curent continuu**.

$$\mathrm{H}(s) \overset{\text{def}}{=} \frac{K}{s(1+Ts)}$$

Sistem de ordin 2 fără întîrziere.

parametri fizici necunoscuți

☝ Constanți sau variabili.

$$K = K_0 = 4$$
$$T = T_0 = 0.5$$

$$K(t) = K_0\left(3\frac{t^2}{T_{\max}^2} - 3\frac{t}{T_{\max}} + 1\right)$$

$$T(t) = T_0\left(1 + \frac{1}{2}\sin\frac{10\pi t}{T_{\max}}\right)$$

$$\forall t \in [0, T_{\max}]$$

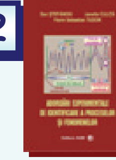
momentul final

Discretizare Euler

$$\mathrm{H}_d(z) \overset{\text{def}}{=} \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$K = \frac{b_1 + b_2}{T_s(1-a_2)} \qquad T = T_s\frac{a_2 b_1 + b_2}{(1-a_2)(b_1+b_2)}$$

☝ Dacă este mare, parametrii sunt aproximativ constanți la începutul perioadei de variație.



❶❶ Identificarea parametrilor fizici

☞ Contextul de lucru

Discretizarea sistemului de ordin 2 cu parametri varibili?

În realitate, modelul (simplificat) al motorului se descompune în **două sisteme de ordin 1 conectate în paralel**.

Discretizarea se efectuează separat pentru fiecare sistem de ordin 1, apoi rezultatele se adună.

Dar în cazul reprezentării pe stare?

parametri fizici necunoscuți (constanți sau variabili)

$$\begin{cases} \dot{\mathbf{x}}(t) = \begin{bmatrix} \theta_1 & 0 \\ \alpha & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \theta_2 \\ 0 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 0 & \beta \end{bmatrix} \mathbf{x}(t) \end{cases} \quad \forall t \in \mathbb{R}_+$$

parametri liber aleşi (constanți)

Discretizare

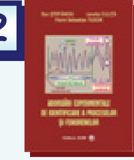
$$\begin{cases} \mathbf{x}[n+1] = \mathbf{A}\,\mathbf{x}[n] + \mathbf{B}\,u(t) + \mathbf{F}\,w[n] \\ y[n] = \mathbf{C}\,\mathbf{x}[n] + \cancel{\mathbf{D}\,u[n]} + E\,v[n] \end{cases} \quad \forall n \in \mathbb{N}$$

zgomote adăugate pentru a forma modelul de identificare

• Cu $\mathbf{A}$ $\mathbf{B}$ $\mathbf{C}$ în funcție de θ_1 θ_2 α β T_s.

Mai mult

$$T = -\frac{1}{\theta_1} \qquad K = -\frac{\alpha\beta\theta_2}{\theta_1}$$



❶❶ Identificarea parametrilor fizici

☞ Contextul de lucru

Obiectiv

- Identificarea parametrilor fizici ai motorului de curent continuu, prin ambele strategii.

Există date I/O?

Se pot genera cu ajutorul unei funcții proiectate special.

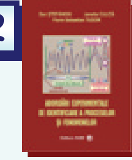
```
>> [D,V,P] = gdata_DCeng(cv,K0,T0,Tmax,Ts,U,lambda);
```

```
Inputs:       cv      # flag indicating the type of process:
                         cv=0 -> constant parameters (default)
                         cv=1 -> variable parameters
              K0      # constant gain (4, by default)
              T0      # constant time constant (0.5 s, by default)
              Tmax    # duration of simulation period
                        (80 s, by default)
              Ts      # sampling period
                        (0.1 s, by default)
              U       # Amplitude of square wave input
                        (0.5, by default)
              lambda  # standard deviation of white noise
                        (1, by default)
                        if null, a noise free process
                        is considered

Outputs:      D       # IDDATA object representing the
                        I/O generated data
              V       # IDDATA object representing the
                        I/O noise generated data
                        (white noise as input, colored noise
                        as output)
              P       # LTI-TF object representing the
                        process that provided the data
                        (see P.num, P.den for transfer
                        function polynomials in case the
                        parameters K and T are constant;
                        otherwise, P.num{1}=K(t), while
                        P.den{1}=T(t))
```

☝ Există numeroase conexiuni între cutia de instrumente IDENT și cutia de instrumente CONTROL.

- Funcții **MATLAB** utilizate:
  - `tf`
  - `c2d`
  - `lsim`
  - `iddata`



❶❶ Identificarea parametrilor fizici

☞ Probleme de simulare

Problema 12.1 (Identificarea parametrilor constanți)

Mini-simulatorul **`ISLAB_12A`** demonstrează maniera în care pot fi identificați parametrii fizici K şi T ai motorului de current continuu, în cazul în care aceştia sunt constanți. Pentru generarea datelor, a fost utilizată rutina **`gdata_DCeng`**. Identificarea modelului discretizat, de tip ARX sau OE, s-a efectuat cu ajutorul funcțiilor MATLAB **`arx`** (care se bazează pe MCMMP) respectiv **`oe`** (care se bazează pe MMEP). Utilizatorul poate selecta de la consolă tipul de model discret adoptat. După identificare, mini-simulatorul afişează estimațiile celor doi parametri fizici şi o serie de grafice, după cum urmează: intrarea şi ieşirea măsurate, respectiv ieşirea măsurată versus ieşirea simulată.

a. Rulați de cîteva ori mini-simulatorul **`ISLAB_12A`**, în scopul efectuării unei analize comparative între modelele de identificare ARX şi OE, din punctul de vedere al preciziei de estimare a parametrilor fizici ai motorului de curent continuu. Explicați de ce ieşirea simulată are un aspect atît de neted.

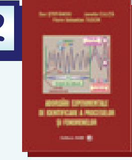
b. Modificați mini-simulatorul **`ISLAB_12A`**, adăugînd graficul diferenței dintre datele de ieşire măsurate şi cele simulate. Această diferență este, de fapt, un *zgomot colorat*, v. Pe grafic, se va indica şi dispersia acestui zgomot, notată prin σ^2. Noul mini-simulator se va numi **`ISLAB_12B`**. Reluați punctul a. cu noul mini-simulator şi comparați dispersiile zgomotului v pentru cele două modele de identificare (ARX şi OE).

➊➊ Identificarea parametrilor fizici

☞ Probleme de simulare

Problema 12.1 (Identificarea parametrilor constanți)

c. Completați mini-simulatorul **ISLAB_12B** cu o secțiune dedicată identificării zgomotului colorat v, care, de fapt, corupe datele achiziționate la ieşire. Pentru aceasta, se vor folosi modele de tip ARMA, numai în cazul în care modelul filtrului util a fost de tip OE. Pentru filtrul ARX, aşa cum se cunoaşte, modelul filtrului de zgomot este de tip AR şi corespunde polinomului A al modelului ARX. Vor fi testate 10 modele ARMA, 5 modele AR şi 5 modele MA, cu indici structurali între 1 şi 20, dintre care se va alege cel mai bun, din punctul de vedere al minimului dispersiei zgomotului alb rezultat, e, dispersie notată prin λ^2. (Se recomandă generarea pseudo-aleatoare a indicilor structurali în gama 1:20.) Pentru a identifica modelele de tip AR, se va apela funcția MATLAB **levinson**, (care se bazează pe *Algoritmul Levinson-Durbin*). Pentru modelele ARMA şi MA se va apela cunoscuta funcție MATLAB **armax**. Adăugați mini-simulatorului **ISLAB_12B** încă două grafice: unul în care se compară ieşirea măsurată cu cea simulată obținută prin adăugarea zgomotului colorat estimat şi altul în care se afişează variația zgomotului alb estimat rezultat. Pe primul grafic, se vor afişa indicii structurali ai modelului de zgomot, iar pe al doilea se va preciza λ^2. Aceste grafice se vor afişa atît pentru cazul OE, cît şi pentru cazul ARX. Care dintre modelele OE cu filtru de zgomot şi ARX vi se pare cel mai bun pentru simularea procesului în condiții de zgomot? Argumentați raspunsul prin rulări ale mini-simulatorului **ISLAB_12B** modificat şi grafice ilustrative. Observați şi indicați diferențele dintre cele două zgomote: colorat (v) şi alb (e).



❶❶ Identificarea parametrilor fizici

☞ Probleme de simulare

Program existent | **ISLAB_12A**

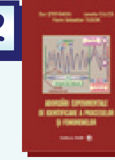
Problema 12.1 (Identificarea parametrilor constanți)

d. Proiectați şi implementați mini-simulatorul **ISLAB_12C**, similar lui **ISLAB_12B**, cu ajutorul căruia cei doi parametri fizici K şi T să fie identificați folosind reprezentarea pe stare. Identificarea modelului pe stare, se poate realiza cu funcțiile MATLAB **pem** sau **n4sid**. Mini-simulatorul va afişa aceleaşi informații ca şi **ISLAB_12B** (inclusiv graficele analizei zgomotului perturbator). În urma testelor, care dintre funcțiile **pem** şi **n4sid** credeți că este mai performantă? Ce influență au constantele α şi β alese liber? (Pentru a răspunde la această întrebare, trebuie să efectuați mai multe teste, în urma cărora să alegeți valorile cele mai convenabile ale celor două constante.)

- Notă: Funcțiile **pem** şi **n4sid** oferă şi matricea zgomotului endogen (intern) al procesului. Se va ține cont de aceasta. Ele nu oferă însă un model al zgomotului (exogen), care corupe datele măsurate la ieşire. Pentru acest zgomot, este necesară în continuare identificarea unui model din clasa ARMA, ca şi în cazul modelului OE.

e. Efectuați o analiză comparativă între cele două strategii de identificare a parametrilor fizici K şi T, indicînd avantajele şi dezavantajele fiecăreia dintre ele.

Programe ce trebuie proiectate | **ISLAB_12B** | **ISLAB_12C**



➊➊ Identificarea parametrilor fizici

☞ Probleme de simulare

Înainte de a rula mini-simulatoarele, trebuie executate comenzile:

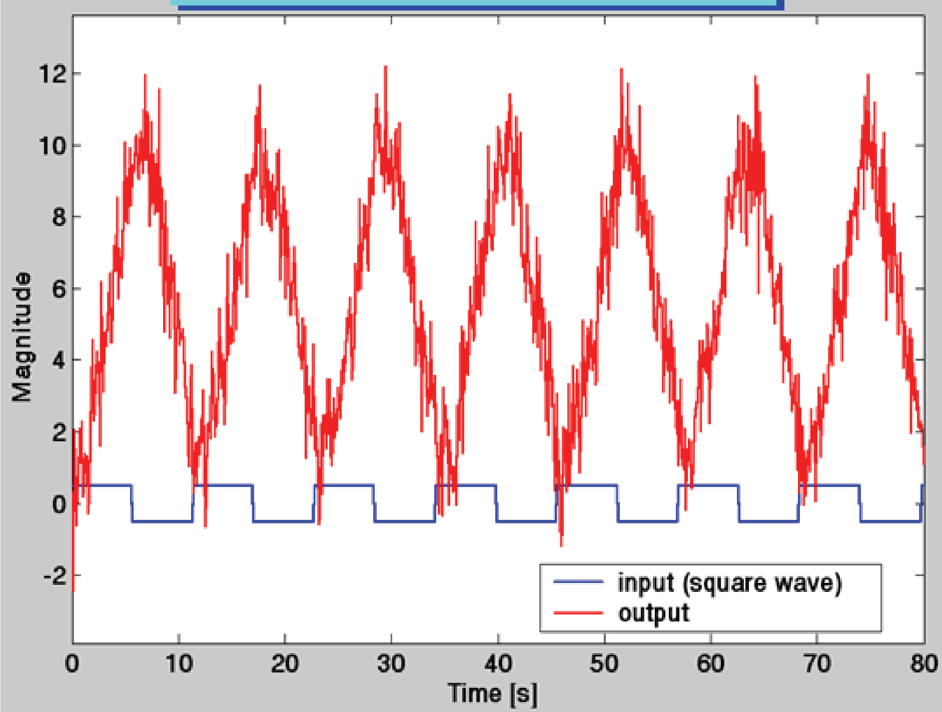
```
> global FIG
> FIG = 1 ;
```

Ce afişează mini-simulatorul ISLAB_12A

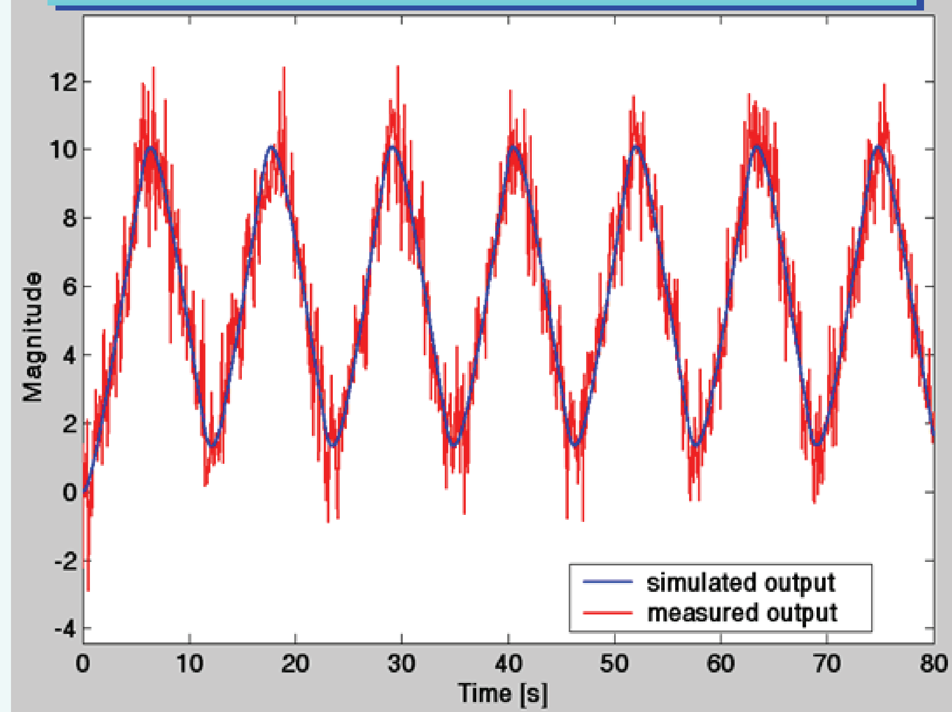
La consolă

```
<ISLAB_12A>: Physical parameters:
          True        Estimated
   K:    4.0000        4.0597
   T:    0.5000        0.4412
```

Intrarea şi ieşirea măsurate



Ieşirea simulată versus ieşirea măsurată



➊➊ Identificarea parametrilor fizici

☞ Probleme de simulare

Problema 12.2 (Identificarea parametrilor variabili)

Mini-simulatorul **ISLAB_12D** este dedicat identificării parametrilor fizici K şi T ai motorului de current continuu, în cazul în care ei variază în timp. Datele de identificare au fost generate tot cu ajutorul rutinei **gdata_DCeng**. Identificarea modelului discretizat, de tip ARX sau OE, ambele adaptive, s-a efectuat cu ajutorul funcțiilor MATLAB **rarx** (care se bazează pe MCMMP-R) respectiv **roe** (care se bazează pe MMEP-R). Utilizatorul poate selecta de la consolă tipul de model discret adoptat. După identificare, mini-simulatorul afişează serie de grafice, după cum urmează: intrarea şi ieşirea măsurate, ieşirea măsurată versus ieşirea simulată, variațiile parametrilor variabili şi modul lor de urmărire, după identificare.

a. Rulați de cîteva ori mini-simulatorul **ISLAB_12D**, în scopul efectuării unei analize comparative între modelele de identificare ARX şi OE, din punctul de vedere al capacității de urmărire a parametrilor fizici ai motorului de curent continuu. De ce credeți că ieşirea simulată este relativ diferită de cea măsurată?

b. Modificați mini-simulatorul **ISLAB_12D**, adăugînd următoarele grafice: diferența dintre datele de ieşire măsurate şi cele simulate, diferențele dintre variațiile parametrilor adevărați şi cele ale parametrilor estimați. Aceste diferențe sunt, *zgomote colorate*: v_y, v_K şi v_T. Pe grafic, se vor indica şi dispersiile acestor zgomote, notate prin σ_v^2, σ_K^2, respectiv σ_T^2. Noul mini-simulator se va numi **ISLAB_12E**. Reluați punctul a. cu noul mini-simulator şi comparați dispersiile zgomotelor pentru cele două modele de identificare (ARX şi OE).

❶❶ Identificarea parametrilor fizici

☞ Probleme de simulare

Problema 12.2 (Identificarea parametrilor variabili)

c. Completați mini-simulatorul **ISLAB_12E** cu o secțiune dedicată identificării zgomotului colorat v_y, care, de fapt, corupe datele achiziționate la ieşire. Pentru aceasta, se vor folosi modele de tip ARMA adaptive, numai în cazul în care modelul filtrului util a fost de tip OE. Pentru filtrul ARX, aşa cum se cunoaşte, modelul filtrului de zgomot este de tip AR şi corespunde polinomului A al modelului ARX. Vor fi testate 10 modele ARMA, 5 modele AR şi 5 modele MA, cu indici structurali între 1 şi 20, dintre care se va alege cel mai bun, din punctul de vedere al minimului dispersiei zgomotului alb rezultat, e, dispersie notată prin λ^2. (Se recomandă generarea pseudo-aleatoare a indicilor structurali în gama 1:20.) Pentru a identifica modelele de tip ARMA, se va apela funcția MATLAB **rarmax**, (care se bazează pe MMEP-R). Adăugați mini-simulatorului **ISLAB_12E** încă două grafice: unul în care se compară ieşirea măsurată cu cea simulată obținută prin adăugarea zgomotului colorat estimat şi altul în care se afişează variația zgomotului alb estimat rezultat. Pe primul grafic, se vor afişa indicii structurali ai modelului de zgomot, iar pe al doilea se va preciza λ^2. Aceste grafice se vor afişa atît pentru cazul OE, cît şi pentru cazul ARX. Care dintre modelele OE cu filtru de zgomot şi ARX vi se pare cel mai bun pentru simularea procesului în condiții de zgomot? Argumentați raspunsul prin rulări ale mini-simulatorului **ISLAB_12E** modificat şi grafice ilustrative. Observați şi indicați diferențele dintre cele două zgomote: colorat (v_y) şi alb (e).

❶❶ Identificarea parametrilor fizici

☞ Probleme de simulare

Program existent ISLAB_12D

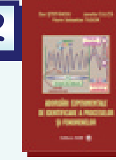
Problema 12.2 (Identificarea parametrilor variabili)

d. Proiectați şi implementați mini-simulatorul **ISLAB_12F**, similar lui **ISLAB_12E**, cu ajutorul căruia cei doi parametri fizici K şi T să fie identificați folosind reprezentarea pe stare. Identificarea modelului pe stare, se poate realiza cu funcția MATLAB **rpem**. Mini-simulatorul va afişa aceleaşi informații ca şi **ISLAB_12E** (inclusiv graficele analizei zgomotului perturbator). Ce influență au constantele α şi β alese liber? (Pentru a răspunde la această întrebare, trebuie să efectuați mai multe teste, în urma cărora să alegeți valorile cele mai convenabile ale celor două constante.)

- Notă: Funcția **rpem** oferă şi matricea zgomotului endogen (intern) al procesului. Se va ține cont de aceasta. Ea nu oferă însă un model al zgomotului (exogen), care corupe datele măsurate la ieşire. Pentru acest zgomot, este necesară în continuare identificarea unui model din clasa ARMA, ca şi în cazul modelului OE.

e. Efectuați o analiză comparativă între cele două strategii de identificare a parametrilor fizici variabili K şi T, indicînd avantajele şi dezavantajele fiecăreia dintre ele.

Programe ce trebuie proiectate: ISLAB_12E ISLAB_12F



➊➊ Identificarea parametrilor fizici

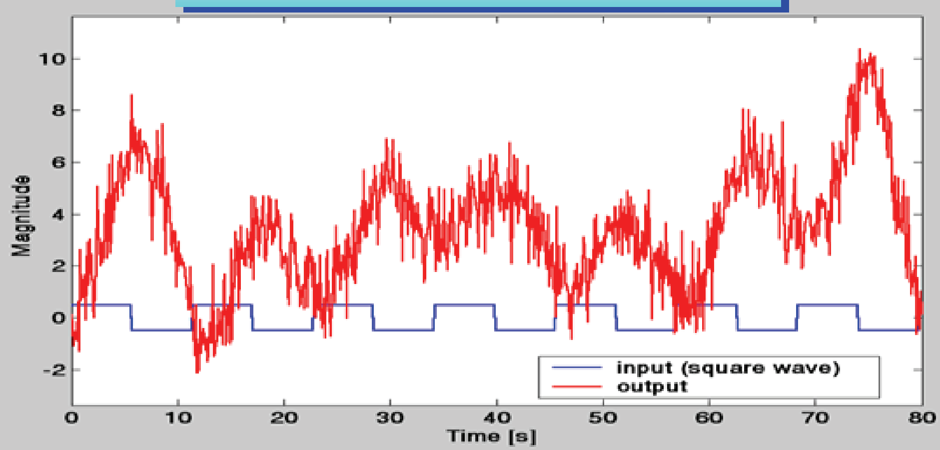
☞ Probleme de simulare

Înainte de a rula mini-simulatoarele, trebuie executate comenzile:

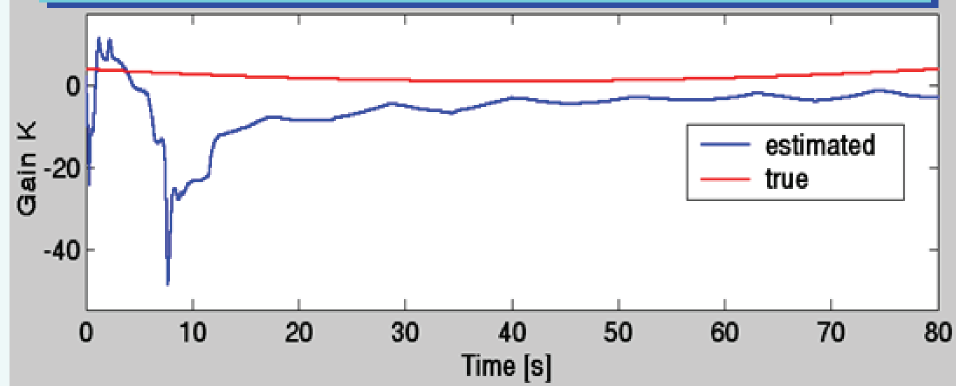
```
> global FIG
> FIG = 1 ;
```

Ce afişează mini-simulatorul ISLAB_12D

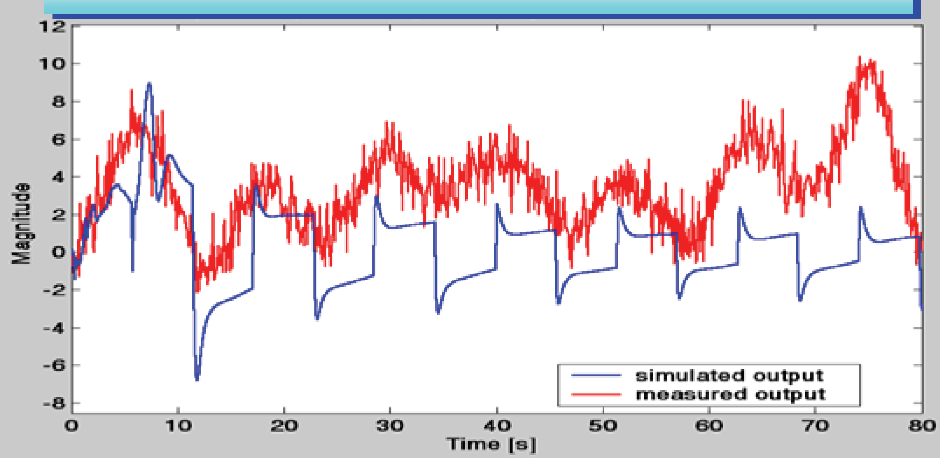
Intrarea şi ieşirea măsurate



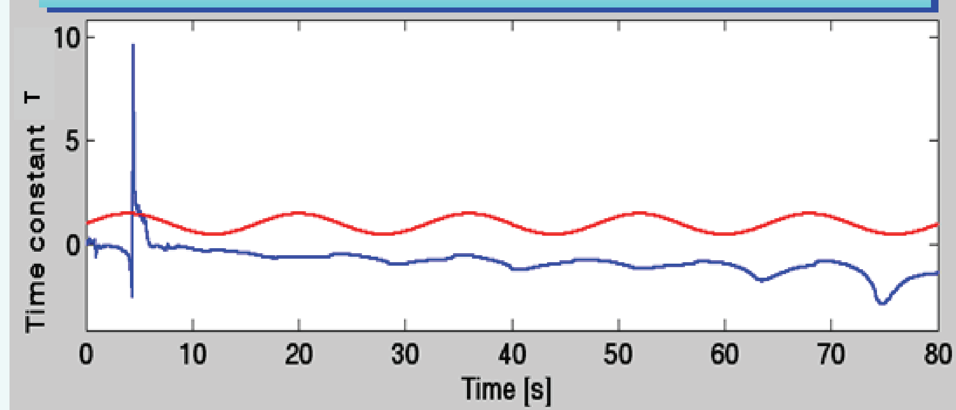
Performanța de urmărire a parametrului K



Ieşirea simulată versus ieşirea măsurată



Performanța de urmărire a parametrului T



❶❶ Identificarea parametrilor fizici

☞ Probleme de simulare

Problema 12.3 (Identificarea parametrilor paraziți, în vederea diagnozei)

Se consideră că un model posibil al mototului de current continuu este următorul, care pune în evidență existența unei constate de timp parazite $T_p \ll T$, apărute în urma uzurii:

$$H(s) = \frac{K}{s(1+Ts)\left(1+T_p s\right)}.$$

De exemplu, $T_p \cong T/10$. Se pune problema identificării celor 3 parametri fizici K, T şi T_p din date I/O măsurate, în scopul diagnosticării motorului.

a. Folosind tehnica descompunerii în fracții simple şi discretizarea de tip Euler-Padé, deduceți expresiile celor 3 parametri fizici în funcție de parametrii modelului de identificare:

$$H_d(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}.$$

Implementați relațiile obținute într-o funcție numită **`Hz2Hs_DCeng`**.

❶❶ Identificarea parametrilor fizici

☞ Probleme de simulare

Problema 12.3 (Identificarea parametrilor paraziți, în vederea diagnozei)

b. Modificați funcția de generare a datelor `gdata_DCeng`, astfel încît să se poată obține date prin intermediul sistemului continuu cu o constantă parazită de cel puțin 10 ori mai mică decît constanta principală. Credeți că intrarea de tip pulsuri dreptunghiulare este suficient de persistentă pentru a identifica toți parametrii fizici (inclusiv constanta parazită)? Pentru siguranță, oferiți posibilitatea utilizatorului de a alege încă două intrări persistente: una formată tot din pulsuri dreptunghiulare dar cu durate ale pulsurilor alese pseudo-aleator şi alta de tip SPAB, în care cele două valori ale semnalului sunt chiar ale pulsurilor dreptunghiulare (`-U` şi `+U`). Denumiți noua funcție `IOdata_DCeng`.

c. Proiectați şi implementați mini-simulatorul `ISLAB_12G`, asemănător lui `ISLAB_12B`, dar cu scopul de a identifica 3 parametri fizici constanți în loc de 2. Efectuați mai multe simulări, cu diferite seturi de date I/O (pentru toate cele 3 tipuri de intrări şi cu un set crescător de valori ale parametrului `lambda`, adică pentru valori descrescătoare ale raportului semnal-zgomot, SNR). Comentați toate rezultatele obținute şi puneți în evidență condițiile în care poate fi identificată, cu precizie satisfăcătoare, constanta de timp parazită a motorului.

d. Cum credeți că s-ar putea efectua o diagnoză de defect în funcție de valorile identificate ale constantei parazite şi SNR? Imaginați un mic scenariu, în care să fie puse în evidență 5 nivele de uzură (de la incipient la sever), în funcție de valori ale constantei parazite şi SNR. (Este cunoscut faptul că, odată cu uzura, scade şi SNR.) Cînd ar trebui ca motorul de curent continuu să fie oprit în vederea reparării?

Program şi rutine ce trebuie proiectate

`Hz2Hs_DCeng` | `IOdata_DCeng` | `ISLAB_12G`

❶❶ Identificarea parametrilor fizici

☞ Probleme de simulare

Notă privind punctul a. al Problemei 12.3

• Formula **ideală** de calcul a funcției de transfer discrete din cea continuă este următoarea:

$$\mathrm{H}_d(z) \overset{\text{def}}{=} (z-1) \sum_{s\in\{\text{poli}\}} \mathrm{Rez}\left(\frac{\mathrm{H}(s)}{s} \frac{1}{z - e^{sT_s}} \right)$$

$$e^{sT_s} \cong 1 + sT_s$$

aproximarea Euler (Padé)

☝ Se poate calcula, folosind descompunerea în fracții simple.

Exemplul 3 Discretizarea Euler a modelului de ordin 2 cu parametri constanți

$$\mathrm{H}(s) = \frac{\omega_0^2}{s^2 + 2\varsigma\omega_0 s + \omega_0^2}$$

$$\mathrm{H}_d(z) \overset{\text{def}}{=} \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$0 \le \varsigma < 1$ (poli complecşi)

$$\begin{cases} a_1 = -2\beta_{\mathbb{C}}\gamma \\ a_2 = \gamma^2 \end{cases}$$

$$\begin{cases} b_1 = 1 - \gamma\left(\beta_{\mathbb{C}} + \alpha_{\mathbb{C}}\varsigma\dfrac{\omega_0}{\omega_\varsigma}\right) \\ b_2 = \gamma^2 + \gamma\left(\beta_{\mathbb{C}} + \alpha_{\mathbb{C}}\varsigma\dfrac{\omega_0}{\omega_\varsigma}\right) \end{cases}$$

$$\begin{cases} \alpha_{\mathbb{C}} = \sin\omega_\varsigma T_s \\ \beta_{\mathbb{C}} = \cos\omega_\varsigma T_s \end{cases}$$

$$\begin{cases} \omega_\varsigma = \omega_0\sqrt{\left|1-\varsigma^2\right|} \\ \gamma = e^{-\varsigma\omega_0 T_s} \end{cases}$$

$\varsigma > 1$ (poli reali ≠)

$$\begin{cases} a_1 = -2\beta_{\mathbb{R}}\gamma \\ a_2 = \gamma^2 \end{cases}$$

$$\begin{cases} b_1 = 1 - \gamma\left(\beta_{\mathbb{R}} + \alpha_{\mathbb{R}}\varsigma\dfrac{\omega_0}{\omega_\varsigma}\right) \\ b_2 = \gamma^2 + \gamma\left(\beta_{\mathbb{R}} + \alpha_{\mathbb{R}}\varsigma\dfrac{\omega_0}{\omega_\varsigma}\right) \end{cases}$$

$$\begin{cases} \alpha_{\mathbb{R}} = \mathrm{sh}\,\omega_\varsigma T_s \\ \beta_{\mathbb{R}} = \mathrm{ch}\,\omega_\varsigma T_s \end{cases}$$

$\varsigma = 1$ (poli reali =)

$$\begin{cases} a_1 = -2e^{-\omega_0 T_s} \\ a_2 = e^{-2\omega_0 T_s} \end{cases}$$

$$\begin{cases} b_1 = 1 + (T_s - 1)e^{-\omega_0 T_s} \\ b_2 = e^{-\omega_0 T_s}\left(e^{-\omega_0 T_s} - T_s - 1\right) \end{cases}$$

Cu toate acestea, **dacă polii sunt reali**, discretizarea se poate efectua şi **fără a utiliza Teorema reziduurilor**, pentru fiecare fracție simplă în parte, urmată de adunarea rezultatelor.

☝ Rezultat final uşor diferit.