

Identificarea sistemelor

Ingineria sistemelor, anul 3
Universitatea Tehnică din Cluj-Napoca

Lucian Buşoniu



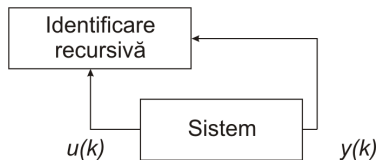
Partea IX

Identificarea recursivă

Conținut

- 1 Introducere și motivare
- 2 Metodele CMMP și ARX recursive
- 3 Metoda recursivă a variabilelor instrumentale

Identificarea recursivă: Idee



Metodele recursive pot funcționa *online*, în paralel cu sistemul.

La fiecare pas k , calculează o nouă *estimare a parametrilor* $\hat{\theta}(k)$ (= un nou model), folosind estimarea anterioară $\hat{\theta}(k-1)$ și noile date disponibile $u(k)$, $y(k)$.

Observație: Pentru a le diferenția de identificarea recursivă, vom numi metodele precedente, care folosesc întregul set de date deodată, metode *offline*.

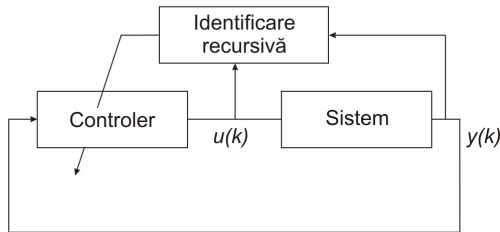
Motivare

Metodele recursive:

- Necesită mai puțină memorie, și mai puțin timp de calcul *pentru fiecare actualizare*, decât algoritmul offline.
- ⇒ Mai ușor de aplicat în timp real.
(timpul total, după setul complet de date, poate fi mai mare)
- Cu modificările potrivite, pot trata sistemele variabile în timp.

Motivare (continuare)

- Dacă modelul este folosit pentru a proiecta un regulator, obținem o schemă de *control adaptiv*:



Control adaptiv bazat pe identificarea recursivă

inițializează modelul și controllerul

loop la fiecare pas $k = 1, 2, \dots$

aplică $u(k)$ folosind controllerul curent, măsoară $y(k)$

actualizează modelul folosind ultimul eșantion $u(k), y(k)$

adaptează/reproiectează controllerul folosind modelul actualizat

end loop

Dezavantaj

Metodele recursive sunt în general o aproximare a metodelor offline
⇒ garanțiile de performanță sunt mai greu de obținut.

Exemplu motivant: Estimarea unui scalar

Model:

$$y(k) = b + e(k) = 1 \cdot b + e(k) = \varphi(k)\theta + e(k)$$

unde $\varphi(k) = 1 \forall k$, $\theta = b$.

Pentru măsurătorile până la (inclusiv) k :

$$y(1) = \varphi(1)\theta = 1 \cdot b$$

...

$$y(k) = \varphi(k)\theta = 1 \cdot b$$

După câteva calcule, soluția este:

$$\hat{\theta}(k) = \frac{1}{k} [y(1) + \dots + y(k)]$$

(estimarea este media măsurătorilor, filtrând zgomotul).

Estimarea unui scalar: Formulare recursivă

Rescriem formula:

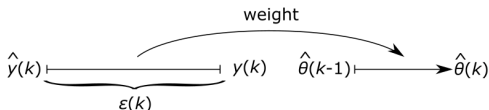
$$\begin{aligned}\hat{\theta}(k) &= \frac{1}{k}[y(1) + \dots + y(k)] \\ &= \frac{1}{k}[(k-1)\frac{1}{k-1}(y(1) + \dots + y(k-1)) + y(k)] \\ &= \frac{1}{k}[(k-1)\hat{\theta}(k-1) + y(k)]\end{aligned}$$

(deja o formulă recursivă, dar continuăm pentru a înțelege mai mult)

$$\begin{aligned}&= \frac{1}{k}[k\hat{\theta}(k-1) + y(k) - \hat{\theta}(k-1)] \\ &= \hat{\theta}(k-1) + \frac{1}{k}[y(k) - \hat{\theta}(k-1)]\end{aligned}$$

Estimarea unui scalar: Proprietăți

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \frac{1}{k} [y(k) - \hat{\theta}(k-1)]$$



Metoda are multe din proprietățile tehnicilor recursive generale:

- Formulă recursivă: noua estimare $\hat{\theta}(k)$ calculată folosind estimarea anterioară $\hat{\theta}(k-1)$ și noua măsurătoare $y(k)$.
- $[y(k) - \hat{\theta}(k-1)]$ este o *eroare de predicție* $\varepsilon(k)$, fiindcă $\hat{\theta}(k-1) = \hat{b} = \hat{y}(k)$, predicția cu un pas înainte a ieșirii.
- Actualizarea aplică o corecție proporțională cu eroarea $\varepsilon(k)$, ponderată de $\frac{1}{k}$:
 - Când eroarea $\varepsilon(k)$ este mare (sau mică), se efectuează o corecție mare (sau mică).
 - Pondere $\frac{1}{k}$ descrește cu pasul k , ducând la corecții mai mici pe măsură ce estimarea $\hat{\theta}$ se îmbunătățește.

Conținut

- 1 Introducere și motivare
- 2 Metodele CMMP și ARX recursive
 - Metoda CMMP recursivă în cazul general
 - Metoda ARX recursivă
 - Exemplu Matlab
- 3 Metoda recursivă a variabilelor instrumentale

Reamintim: Regresie liniară CMMP

Model:

$$y(k) = \varphi^\top(k)\theta + e(k)$$

Setul de date până la k duce la un sistem liniar de ecuații:

$$y(1) = \varphi^\top(1)\theta$$

$$y(2) = \varphi^\top(2)\theta$$

...

$$y(k) = \varphi^\top(k)\theta$$

După algebră și analiză liniară, soluția CMMP se poate scrie:

$$\hat{\theta}(k) = \left[\sum_{j=1}^k \varphi(j)\varphi^\top(j) \right]^{-1} \left[\sum_{j=1}^k \varphi(j)y(j) \right]$$

CMMP: Formula recursivă

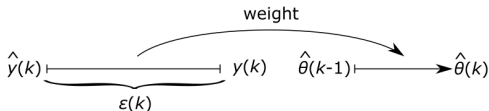
Cu notația $P(k) = \sum_{j=1}^k \varphi(j)\varphi^T(j)$:

$$\begin{aligned}\hat{\theta}(k) &= P^{-1}(k) \left[\sum_{j=1}^k \varphi(j)y(j) \right] \\ &= P^{-1}(k) \left[\sum_{j=1}^{k-1} \varphi(j)y(j) + \varphi(k)y(k) \right] \\ &= P^{-1}(k) \left[P(k-1)\hat{\theta}(k-1) + \varphi(k)y(k) \right] \\ &= P^{-1}(k) \left[[P(k) - \varphi(k)\varphi^T(k)]\hat{\theta}(k-1) + \varphi(k)y(k) \right] \\ &= \hat{\theta}(k-1) + P^{-1}(k) \left[-\varphi(k)\varphi^T(k)\hat{\theta}(k-1) + \varphi(k)y(k) \right] \\ &= \hat{\theta}(k-1) + P^{-1}(k)\varphi(k) \left[y(k) - \varphi^T(k)\hat{\theta}(k-1) \right]\end{aligned}$$

CMMP: Proprietăți

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P^{-1}(k)\varphi(k) \left[y(k) - \varphi^T(k)\hat{\theta}(k-1) \right]$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + W(k)\varepsilon(k)$$



- Formulă recursivă: noua estimare $\hat{\theta}(k)$ calculată pornind de la estimarea anterioară $\hat{\theta}(k-1)$ și noua măsurătoare $y(k)$.
- $\left[y(k) - \varphi^T(k)\hat{\theta}(k-1) \right]$ este o eroare de predicție $\varepsilon(k)$, fiindcă $\varphi^T(k)\hat{\theta}(k-1) = \hat{y}(k)$ este predicția cu un pas înainte folosind vectorul anterior de parametri.
- $W(k) = P^{-1}(k)\varphi(k)$ este un vector de ponderi: elementele lui P cresc pe măsură ce pasul k crește, așadar $W(k)$ descrește.

Calculul recursiv al matricii inverse

Formula anterioară necesită inversarea unei matrici, $P^{-1}(k)$, o operație costisitoare.

Matricea P se scrie ușor recursiv: $P(k) = P(k-1) + \varphi(k)\varphi^T(k)$, dar această formă nu ne ajută direct; matricea încă trebuie inversată.

Formula **Sherman-Morrison** actualizează recursiv direct inversa P^{-1} :

$$P^{-1}(k) = P^{-1}(k-1) - \frac{P^{-1}(k-1)\varphi(k)\varphi^T(k)P^{-1}(k-1)}{1 + \varphi^T(k)P^{-1}(k-1)\varphi(k)}$$

Exercițiu: Demonstrați formula Sherman-Morrison! (algebră liniară)

Algoritm

Metoda CMMP recursivă

inițializează $\hat{\theta}(0)$, $P^{-1}(0)$

loop la fiecare pas $k = 1, 2, \dots$

măsoară $y(k)$, formează vectorul de regresori $\varphi(k)$

calculează eroarea de predicție $\varepsilon(k) = y(k) - \varphi^T(k)\hat{\theta}(k-1)$

actualizează $P^{-1}(k) = P^{-1}(k-1) - \frac{P^{-1}(k-1)\varphi(k)\varphi^T(k)P^{-1}(k-1)}{1 + \varphi^T(k)P^{-1}(k-1)\varphi(k)}$

calculează ponderile $W(k) = P^{-1}(k)\varphi(k)$

actualizează parametrii $\hat{\theta}(k) = \hat{\theta}(k-1) + W(k)\varepsilon(k)$

end loop

Fiecare model actualizat poate fi folosit în diverse scopuri, de ex. aproximare

Inițializare

Metoda necesită vectorul inițial de parametri $\hat{\theta}(0)$, și inversa inițială $P^{-1}(0)$.

Valori tipice fără cunoștințe a priori:

- $\hat{\theta}(0) = [0, \dots, 0]^T$ - un vector de n zerouri.
- $P^{-1}(0) = \frac{1}{\delta} I$, cu δ un număr mic, de ex. 10^{-3} .
Valoarea inițială corespunzătoare a lui P este $P(0) = \delta I$.

Intuiție: P mic corespunde unui grad de încredere mic în parametrii inițiali, ducând la valori mari în $P^{-1}(0)$. Așadar, ponderile $W(k)$ sunt mari inițial, și se efectuează corecții mari ale parametrilor $\hat{\theta}$.

Dacă sunt disponibile valori a priori pentru $\hat{\theta}$, ele se pot folosi la inițializarea $\hat{\theta}(0)$, și δ poate fi mărit corespunzător. Interpretarea este un grad de încredere mai mare în estimarea inițială, ducând la corecții mai mici.

Verificare: Recuperarea cazului scalar

$$y(k) = b + e(k) = \varphi(k)\theta + e(k), \text{ unde } \varphi(k) = 1, \theta = b$$

Luăm $\hat{\theta}(0) = 0, P^{-1}(0) \rightarrow \infty$. Folosind Sherman-Morrison, avem:

$$P^{-1}(k) = P^{-1}(k-1) - \frac{(P^{-1}(k-1))^2}{1 + P^{-1}(k-1)} = \frac{P^{-1}(k-1)}{1 + P^{-1}(k-1)}$$

$$P^{-1}(1) = 1$$

$$P^{-1}(2) = \frac{1}{2}$$

...

$$P^{-1}(k) = \frac{1}{k}$$

Avem $\varepsilon(k) = y(k) - \hat{\theta}(k-1)$ și $W(k) = P^{-1}(k) = \frac{1}{k}$, ducând la:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + W(k)\varepsilon(k) = \hat{\theta}(k-1) + \frac{1}{k}[y(k) - \hat{\theta}(k-1)]$$

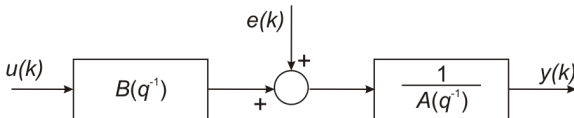
⇒ a fost recuperată formula din cazul scalar.

Conținut

- 1 Introducere și motivare
- 2 Metodele CMMP și ARX recursive
 - Metoda CMMP recursivă în cazul general
 - Metoda ARX recursivă
 - Exemplu Matlab
- 3 Metoda recursivă a variabilelor instrumentale

Reamintim: Model ARX

$$A(q^{-1})y(k) = B(q^{-1})u(k) + e(k)$$



ARX ca regresie liniară:

$$\begin{aligned} y(k) &= -a_1y(k-1) - a_2y(k-2) - \dots - a_{na}y(k-na) \\ &\quad b_1u(k-1) + b_2u(k-2) + \dots + b_{nb}u(k-nb) + e(k) \\ &= [-y(k-1) \quad \dots \quad -y(k-na) \quad u(k-1) \quad \dots \quad u(k-nb)] \\ &\quad \cdot [a_1 \quad \dots \quad a_{na} \quad b_1 \quad \dots \quad b_{nb}]^T + e(k) \\ &=: \varphi^T(k)\theta + e(k) \end{aligned}$$

Vector de regresori: $\varphi \in \mathbb{R}^{na+nb}$, ieșirile și intrările precedente.

Vector de parametri: $\theta \in \mathbb{R}^{na+nb}$, coeficienții polinoamelor

$$a_1, a_2, \dots, a_{na} \text{ și } b_1, b_2, \dots, b_{nb}.$$

ARX recursiv

Instanțiem metoda CMMP recursivă din cazul general, pentru forma de regresie liniară a modelului ARX.

ARX recursiv

inițializează $\hat{\theta}(0)$, $P^{-1}(0)$

loop la fiecare pas $k = 1, 2, \dots$

măsoară $u(k)$, $y(k)$

formează vectorul de regresori

$$\varphi(k) = [-y(k-1), \dots, -y(k-na), u(k-1), \dots, u(k-nb)]^T$$

calculează eroarea de predicție $\varepsilon(k) = y(k) - \varphi^T(k)\hat{\theta}(k-1)$

actualizează $P^{-1}(k) = P^{-1}(k-1) - \frac{P^{-1}(k-1)\varphi(k)\varphi^T(k)P^{-1}(k-1)}{1 + \varphi^T(k)P^{-1}(k-1)\varphi(k)}$

calculează ponderile $W(k) = P^{-1}(k)\varphi(k)$

actualizează parametrii $\hat{\theta}(k) = \hat{\theta}(k-1) + W(k)\varepsilon(k)$

end loop

Observație: leșirile mai vechi de na pași, și intrările mai vechi de nb pași, nu sunt folosite și pot fi “uite”, reducând necesarul de memorie al algoritmului.

Garanție de performanță: Idee

Cu $\hat{\theta}(0) = 0$, $P^{-1}(0) = \frac{1}{\delta} I$ și $\delta \rightarrow 0$, ARX recursiv la pasul k este echivalent cu ARX offline pe setul de date $u(1), y(1), \dots, u(k), y(k)$.

⇒ aceeași garanție ca ARX offline:

Teoremă

Date fiind ipotezele potrivite (incluzând existența parametrilor corecți θ_0), identificarea ARX este **consistentă**: parametrii estimați $\hat{\theta}$ converg la cei corecți θ_0 când $k \rightarrow \infty$.

Inițializarea lui P în acest fel asigură egalitatea inverselor cu cele calculate în cazul offline.

Conținut

- 1 Introducere și motivare
- 2 Metodele CMMP și ARX recursive**
 - Metoda CMMP recursivă în cazul general
 - Metoda ARX recursivă
 - Exemplu Matlab**
- 3 Metoda recursivă a variabilelor instrumentale

Sistem

Pentru a ilustra metoda ARX recursivă, luăm un sistem cunoscut:

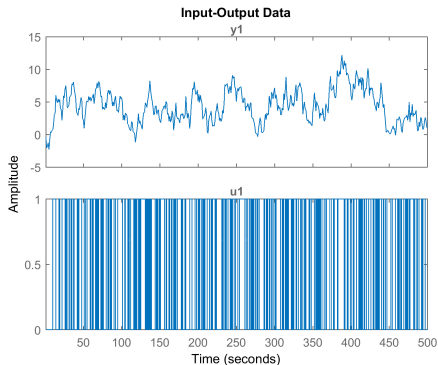
$$y(k) + ay(k - 1) = bu(k - 1) + e(k), \quad a = -0.9, b = 1$$

(Söderström & Stoica)

```
system = idpoly([1 -0.9], [0 1]);
```

Datele de identificare sunt obținute în simulare:

```
sim(system, u, 'noise');
```



ARX recursiv

```
model = rarx(id, [na, nb, nk], 'ff', 1, th0, Pinv0);
```

Argumente:

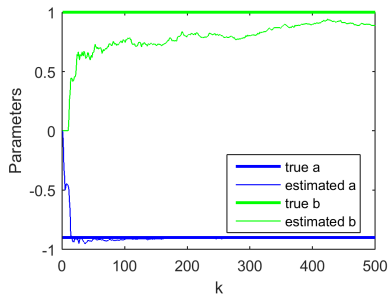
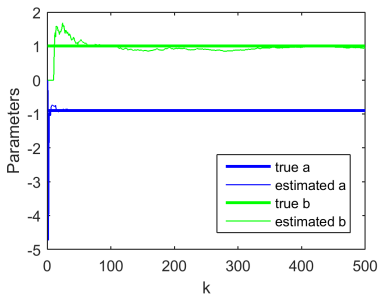
- 1 Datele de identificare.
- 2 Vector conținând ordinele polinoamelor A și B și întârzierea nk .
- 3 'ff', 1 selectează varianta de algoritm prezentată în curs.
- 4 th0 este vectorul inițial de parametri.
- 5 Pinv0 este inversa inițială $P^{-1}(0)$.

Rezultate

$$na = nb = nk = 1$$

$P^{-1}(0) = \frac{1}{\delta}I$, stânga: $\delta = 10^{-3}$, dreapta: $\delta = 100$.

În toate experimentele, $\theta_0 = [0, 0]^T$



Concluzii:

- Converge la valorile corecte ale parametrilor...
- ...mai încet când δ este mai mare (în acest caz, gradul mare de încredere este greșit alocat unei soluții inițiale incorecte)

Perturbație colorată

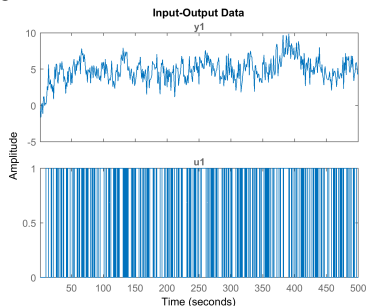
Luăm acum un sistem în forma *eroare de ieșire* (care nu satisface așadar structura ARX cu zgomot alb):

$$y(k) = \frac{bq^{-1}}{1 + fq^{-1}}u(k) + e(k), \quad f = -0.9, b = 1$$

(Söderström & Stoica)

```
system = idpoly([], [0 1], [], [], [1 -.9]);
```

Date de identificare:

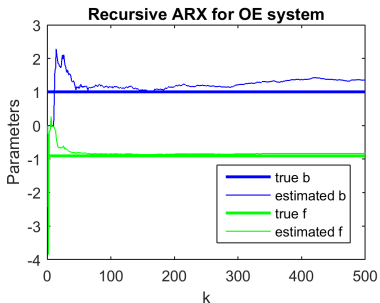


Rezultate

$nf = nb = nk = 1$, deci încercăm un ARX de ordinul 1:

$$y(k) + fy(k-1) = bu(k-1) + e(k)$$

Luăm și $P^{-1}(0) = \frac{1}{\delta}I$, $\delta = 10^{-3}$.



Concluzie: Nu converge la valorile corecte! – datorită perturbației colorate (sistemul nu este în clasa de modele considerată). La fel ca la cursul de VI, algoritmul nu este consistent.

Alți algoritmi de tip MEP recursivi

În Matlab sunt disponibile și alte variante recursive de MEP, de ex.:

- ARMAX, `rarmax`
- eroare de ieșire, `roe`
- modele în forma generală de la minimizarea erorii de predicție, `rpem`

Conținut

- 1 Introducere și motivare
- 2 Metodele CMMP și ARX recursive
- 3 Metoda recursivă a variabilelor instrumentale
 - Metoda VI recursivă
 - Exemplu Matlab

Reamintim: Metoda variabilelor instrumentale

Metoda VI găsește vectorul de parametri:

$$\hat{\theta} = \left[\frac{1}{N} \sum_{k=1}^N Z(k) \varphi^T(k) \right]^{-1} \left[\frac{1}{N} \sum_{k=1}^N Z(k) y(k) \right]$$

care este soluția sistemului de ecuații:

$$\left[\frac{1}{N} \sum_{k=1}^N Z(k) [\varphi^T(k) \theta - y(k)] \right] = 0$$

Vectorul de regresori:

$$\varphi(k) = [-y(k-1), \dots, -y(k-na), u(k-1), \dots, u(k-nb)]^T,$$

iar vectorul VI este de obicei:

$$Z(k) = [-x(k-1), \dots, -x(k-na), u(k-1), \dots, u(k-nb)]^T.$$

Cum vectorul VI $Z(k)$ este necorelat cu perturbația, metoda VI funcționează pentru perturbații colorate.

Formulare recursivă (1)

Putem elimina împărțirea cu numărul N de puncte din setul de date, obținând:

$$\hat{\theta} = \left[\sum_{k=1}^N Z(k)\varphi^T(k) \right]^{-1} \left[\sum_{k=1}^N Z(k)y(k) \right]$$

Împărțirea era necesară pentru obținerea mediilor în VI offline, fiindcă sumele ar fi fost foarte mari pentru multe date, ducând la probleme numerice. Metoda VI recursivă va aduna termenii unul câte unul și va fi așadar mai stabilă d.p.d.v. numeric.

Rescriem ecuația pentru cazul recursiv:

$$\hat{\theta}(k) = P^{-1}(k) \left[\sum_{j=1}^k Z(j)y(j) \right]$$

unde $P(k) = \sum_{j=1}^k Z(j)\varphi^T(j)$.

Formulare recursivă (2)

Cu această definiție P , formula recursivă se găsește ca și pentru CMMP:

$$\begin{aligned}\hat{\theta}(k) &= P^{-1}(k) \left[\sum_{j=1}^k Z(j)y(j) \right] \\ &= P^{-1}(k) \left[\sum_{j=1}^{k-1} Z(j)y(j) + Z(k)y(k) \right] \\ &= P^{-1}(k) \left[P(k-1)\hat{\theta}(k-1) + Z(k)y(k) \right] \\ &= P^{-1}(k) \left[[P(k) - Z(k)\varphi^T(k)]\hat{\theta}(k-1) + Z(k)y(k) \right] \\ &= \hat{\theta}(k-1) + P^{-1}(k) \left[-Z(k)\varphi^T(k)\hat{\theta}(k-1) + Z(k)y(k) \right] \\ &= \hat{\theta}(k-1) + P^{-1}(k)Z(k) \left[y(k) - \varphi^T(k)\hat{\theta}(k-1) \right]\end{aligned}$$

Formulare recursivă (3)

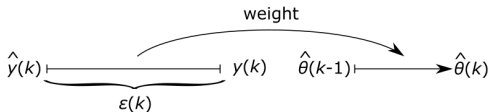
Formula finală:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P^{-1}(k)Z(k) \left[y(k) - \varphi^T(k)\hat{\theta}(k-1) \right]$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + W(k)\varepsilon(k)$$

cu Sherman-Morrison pentru actualizarea inversei:

$$P^{-1}(k) = P^{-1}(k-1) - \frac{P^{-1}(k-1)Z(k)\varphi^T(k)P^{-1}(k-1)}{1 + \varphi^T(k)P^{-1}(k-1)Z(k)}$$



Metoda VI recursivă: Algoritm

VI recursiv

inițializează $\hat{\theta}(0)$, $P^{-1}(0)$

loop la fiecare pas $k = 1, 2, \dots$

măsoară $y(k)$, formează $\varphi(k)$ și vectorul VI

$$Z(k) = [-x(k-1), \dots, -x(k-na), u(k-1), \dots, u(k-nb)]^T$$

calculează eroarea de predicție $\varepsilon(k) = y(k) - \varphi^T(k)\hat{\theta}(k-1)$

actualizează $P^{-1}(k) = P^{-1}(k-1) - \frac{P^{-1}(k-1)Z(k)\varphi^T(k)P^{-1}(k-1)}{1 + \varphi^T(k)P^{-1}(k-1)Z(k)}$

calculează ponderile $W(k) = P^{-1}(k)Z(k)$

actualizează parametrii $\hat{\theta}(k) = \hat{\theta}(k-1) + W(k)\varepsilon(k)$

end loop

Table of contents

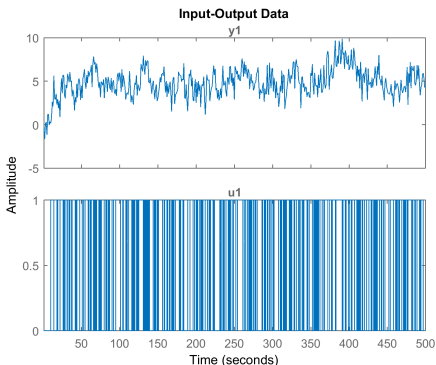
- 1 Introducere și motivare
- 2 Metodele CMMP și ARX recursive
- 3 Metoda recursivă a variabilelor instrumentale
 - Metoda VI recursivă
 - Exemplu Matlab

Sistem OE

Considerăm același sistem OE pe care l-am folosit în exemplul ARX:

$$y(k) = \frac{bq^{-1}}{1 + fq^{-1}}u(k) + e(k), \quad f = -0.9, b = 1$$

cu același set de date:



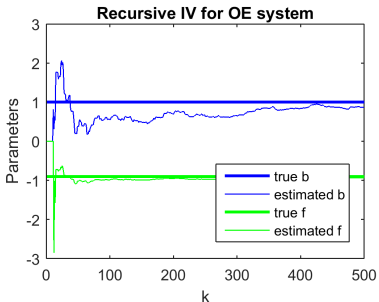
Rezultat cu metoda VI recursivă

Folosim modelul VI cu perturbație colorată $v(k)$:

$$y(k) + fy(k-1) = bu(k-1) + v(k)$$

Vector de VI: $Z(k) = [u(k-2), u(k-1)]$.

Ca mai sus, $P^{-1}(0) = \frac{1}{\delta}I$, $\delta = 10^{-3}$.



Concluzie: Mai bun decât ARX recursiv, parametrii converg la valorile corecte.