Academia de Studii Economice Bucureşti
Facultatea de Cibernetică, Statistică, Informatică si Economie Matematică

PROIECT

CIBERNETICA ECONOMICA

Tema proiectului:
REGLAREA ŞI POZIŢIONAREA POLILOR

Gr. 1090
Secţia economie-matematică

-2005-

PROIECT. Reglarea si pozitionarea polilor.
Functii importante: place, acker, impulse, reg.

PLACE = Tehnica de pozitionare a polilor.

K= PLACE (A, B, P) calculeazã o matrice de stare, K astfel ca valorile proprii de A– B*K sunt acele specificate în vectorul P. Nici o valoare proprie nu ar trebui sa aiba o multiplicitate mai mare decat cea a numãrului de intrãri.

[K, PREC, MESSAGE] = PLACE(A, B, P)

returneaza PREC, o estimatie a apropierii valorilor proprii de A-B*K care sa se potriveasca cu locatiile specificate P. (PREC masoara acuratetea numarului zecimal, prin numarul de zecimale, in bucla actuala a polilor). Daca vreun pol diferit de 0 contine o eroare mai mare de 10%, fata de locatia dorita, se da un MESSAGE ce contine un mesaj de avertizare.

ACKER = pozitionare a polilor folosind formula lui Ackermann(spor de selectie).

K = ACKER(A, B, P)

calculeaza matricea inversa de spor, K astfel incat, sistemul de intrari:

x = Ax + Bu

ce are legea inversa: u = -K*x , acest sistem sa aiba poli apropiati , la valori specificate in vectorul P.

de exemplu: P = eig (A-B*K)

Nota: Acest algoritm foloseste formula lui Ackermann. Aceasta metoda NU ESTE fiabila (de incredere) din punct de vedere numeric; ea incepe sa se destrame rapid pentru problemele cu ordin mai mare de 10, sau pentru sisteme controlabile slabe (slab controlabile). Un mesaj de avertizare este afisat daca polii fara zerouri (diferiti de zero) sunt calculati cu o eroare mai mare de 10% fata de locatia dorita, specificata in P.

IMPULSE

IMPULSE(SYS) deseneaza raspunsul impuls al modelelor LTI, tip SYS, (aceste modele fiind create fie cu TF, ZPK, sau SS). Pentru modelele multi-input, comenzile impulse independente, sunt aplicate fiecarui canal de input. Perioada de timp si numarul de puncte sunt alese in mod automat. Pentru sistemele continue cu feedthrough direct, pulsul infinit la t=0, nu este luat in seama (este ignorat).

IMPULSE(SYS, TFINAL) simuleaza impulsul raspuns de la momentul t=0 la momentul t = TFINAL. Pentru sistemele cu timp discret, la care nu se specifica momentul final, TFINAL este interpretat ca numar de selectii.

IMPULSE (SYS1, SYS2, ..., T) deseneaza raspunsul modelelor multiplu LTI

SYS 1, SYS2, ... pe un singur grafic. Vectorul timp (T) este optional. Se pot specifica de exemplu, culoarea, tipul de linie, si markerul pentru fiecare sistem in parte.
Ex: impulse (sys1, `r`, sys2, `y--`, sys3, `gx`).

[Y,T] = IMPULSE (SYS)

returneaza output-ul Y si vectorul timp T folosit in simulare. Nici un grafic nu e desenat pe ecran. Daca SYS are NY output-uri si NU inputuri, si in plus, LT=lungimea (T), Y este o matrice de marimea [LT NY NU] unde Y(: , : , j) ne da raspunsul canalului j de input.

[Y, T, X] = IMPULSE (SYS, ...)

returneaza traiectoria X care este un masiv LT * NX * NU ,daca SYS are NX stari.

REG

RSYS = REG (SYS, K, L)

produce un rezultat bazat pe observatii (RSYS), pentru sistemul (dinamic liniar) SYS, presupunand ca toate intrarile sistemului sunt controlabile si toate iesirile sunt observabile (masurabile). Matricile K si L specifica feed-back ul liniar, respectiv castigurile observabile. Pentru sistemul:

x` = Ax + Bu

y = Cx + Du

Rezultatul este:

x_e = [A–BK–LC+LDK] x_e + Ly

u = -K x_e

Acest rezultat ar trebui conectat cu sistemul folosind un feed-back pozitiv. REG se poate folosi in mod similar chiar daca este aplicat unui sistem cu timp discret.

RSYS = REG(SYS, K, L, SENSORS, KNOWN, CONTROLS) se ocupa cu mai multe probleme reglatorii;

- inputurile sistemului sunt: u (controlabile), Ud (intrari cunoscute) si w (inputuri stocastice;
- doar un subset din iesiri (y) sunt masurabile

Subseturile I/O y, Ud si u sunt specificate prin vectorii de index SENSORS, KNOWN, respectiv CONTROLS. Rezultatul RSYS foloseste [Ud;y] ca input pentru a genera comenzile u.

Se pot folosi tehnici de plasare a polilor pentru a afla K si L, sau se pot folosi castigurile LQ si Kalman care sunt date de LQR/DLQR si KALMAN.

Alte functii ce au legatura cu REG:

1. **LQR (Linear –Quadratic Regulator) pentru sisteme cu timp continuu**.

[K, S, E] = LQR(A, B, Q, R, N) calculeaza matricea K, astfel incat legea de feed-back liniar:

$$\mathbf{u = -Kx}$$

minimizeaza functia de cost.

J = Integrala {x'Qx + u'Ru + 2*x'Nu} dt

subiect al dinamicii liniare x`=Ax +Bu.

Matricea N = 0, daca ea este omisa.

$$\mathbf{SA + A'S - (SB+N)R^{-1}(B'S+N') + Q = 0}$$

$$\mathbf{E = EIG\ (A-B*K)}$$

unde: S este solutia ecuatiei Riccati;

E calculeaza valoarile proprii.

2. **DLQR (Linear-Quadratic Regulator design) pentru sisteme cu timp discret**.

[K, S, E] = DLQR(A, B, Q, R, N) calculeaza matricea K de optim, astfel incat legea de feed-back: u[n] = –Kx[n] minimizeaza functia de cost.

J = Sum {x'Qx + u'Ru + 2*x'Nu}

subiect al dinamicii liniare x[n+1] = Ax[n] + Bu[n]

$$\mathbf{A'SA - S - (A'SB+N)(R+B'SB)^{-1}(B'SA+N') + Q = 0}$$

$$\mathbf{E = EIG(A-B*K)}$$

3. ESTIM (Estimator)

EST = ESTIM(SYS, L) returneaza un estimator EST, folosind output-ul L, si sistemul dinamic SYS, presupunand ca toate inputurile din SYS sunt stocastice si toate output-urile sunt masurabile.

Pentru un sistem continuu,

SYS: x`= Ax +Bw

y = Cx +Dw

rezulta estimatorul:

x_e`= [A–LC]x_e + Ly

!y_e! = !C! x_e

!x_e! ! I !

genereaza estimarile x_e si y_e ale lui x, respectiv y. Estim actioneaza similar si aplicat sistemelor cu timp discret.

EST = ESTIM(SYS, L, SENSORS, KNOWN)

se ocupa cu sisteme ceva mai generale SYS, care pot avea intrari atat deterministe, cat si stocastice, si iesiri atat masurabile cat si nemasurabile. Vectorii index

SENSORS si KNOWN specifica output-urile y care sunt masurabile, respectiv inputurile u, care sunt cunoscute. Estimatorul rezultat EST foloseste [u;y] ca intrari, pentru a produce estimarile [y_e; x_e].
Se pot folosi tehnici de pozitionare a polilor (PLACE) pentru a crea observatorul L, sau functia Kalman.

4. **KALMAN. Estimatorul Kalman discret sau continuu**.

[KEST, L, P] = KALMAN (SYS, Qn, Rn, Nn)

returneaza un estimator Kalman KEST, pentru modele continue sau discrete, cu sistemul dinamic SYS. Pentru un model continuu:
x`= Ax + Bu + Gw –> ecuatia de dinamica a variabilelor de stare
y = Cx + Du + Hw + v –> relatiile de iesire
cu inputuri cunoscute u, eroare proceduala w, eroare de masurare v, si covariantele:
E{ww'} = Qn, E{vv'} = Rn, E{wv'} = Nn.

Estimatorul KEST are inputul [u;y] si genereaza estimarile optime y_e, x_e ale lui y, respectiv x, astfel:
x_e` = Ax_e + Bu + L(y – Cx_e – Du)
!y_e! = !C! x_e + !D! u
!x_e! !I! !O!

Sistemul dinamic SYS contine datele modelului (A, [B G], C, [D H]), si Nn, care poate fi =0. Estimatorul Kalman KEST este cu timp continuu, atunci cand SYS este continuu, altfel, cu timp discret. Se returneaza si eroarea de covarianta P. In cazul - timp continuu- cu H=0, P rezolva ecuatia Riccati.

$$\mathbf{AP + PA' - (PC' + G*N)R^{-1}(CP + N'*G') + G*Q*G' = 0}$$

[KEST, L, P] = KALMAN(SYS, Qn, Rn, Nn, SENSORS, KNOWN) se foloseste pentru sisteme mai generale SYS, unde inputurile stocastice [u,w] sunt comasate, si nu toate outputurile sunt masurabile. Vectorii index SENSORS si KNOWN specifica outputurile (y) ale lui SYS care sunt masurabile, si care inputuri u sunt cunoscute. Toate celelalte inputuri se presupune ca sunt stocastice.

SS creeaza un sistem dinamic, sau transforma un model LTI intr-unul dinamic.

SYS = SS(A, B, C, D)

creeaza un model (SS) cu timp continuu, SYS, cu matricile A, B, C, D. Outputul SYS este un obiect SS. Se poate scrie D=0, pentru a se reprezenta matricea zero (cu toate elementele=0) de dimensiuni dorite.

SYS = SS(A, B, C, Ts) creeaza un model (SS) cu timp discret cu timpul Ts (daca timpul este nedeterminat se poate scrie Ts= -1);

SYS = SS creeaza un obiect SS nedefinit.

SYS = SS(D) specifica o matrice statica D.

*Listei de inputuri i se pot adauga parametrii: 'PropertyName1' ,'PropertyValue1', ...

pentru diferitele proprietati ale modelelor SS (se poate tasta LTIPROPS pentru detalii). Pentru a transmite proprietatile unui model LTI catre unul SYS,(dintr-un model LTI existent) se poate folosi sintaxa SYS = SS(A, B, C, D, REFSYS).

Matricile modelului dinamic:

Se pot creea matrici pentru modelele SS folosind matrici ND pentru A, B, C, D.

Primele doua dimensiuni ale A, B, C, D determina numarul de stari, inputuri, si outputuri, iar celelalte dimensiuni specifica marimea matricii.

De exemplu, daca A, B, C, D sunt matrici 4D si ultimele 2 dimensiuni au marimile 2 si 5, atunci:

SYS = SS(A, B, C, D)

creeaza o matrice 2–5 a modelelor SS.

SYS(:,:,k,m) = SS(A(:,:,k,m), ..., D(:,:,k,m)), k=1:2, m=1:5.

Toate modelele SS rezultate din aceste matrici au acelasi numar de outputuri, inputuri, si stari.

SYS = SS(ZEROS([NY NU S1 ... Sk])) aloca spatiu pentru o matrice SS cu NY intrari, NU iesiri, si dimensiunile matricii de [S1 ... Sk].

Transformarea.

SYS = SS(SYS) transforma un model arbitrar LTI intr-unul dinamic;

SYS = SS(SYS, 'min') calculeaza o realizare minima de SYS.