

Laborator 7. Sisteme dinamice pe spatiul starilor

Sinteza elementara

Introducere

Scopul acestui laborator este elaborarea și implementarea câtorva metode de modificare a dinamicii unui sistem astfel încât acesta să satisfacă anumite cerințe. Cerințele elementare sunt legate de stabilitate și comporatarea sistemului la diverse semnale de referință/perturbații. Cerințele de proiectare se pot realiza prin cuplarea unui nou sistem dinamic (de preferință din aceeași clasă de modele) astfel încât sistemul rezultat să se comporte în modul dorit.

0.1 Lege de comandă, stabilizabilitate, alocabilitate

A) Descriere teoretică

Considerăm un sistem liniar

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), & x(0) = x_0 \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (1)$$

cu $x(t) \in \mathcal{R}^n$, $u(t) \in \mathcal{R}^m$, $y(t) \in \mathcal{R}^p$. Această secțiune tratează construcția unei legi de reacție după stare $F \in \mathcal{R}^{m \times n}$ care alocă sau stabilizează (face ca matricea $A + BF$ să aibă spectrul dorit sau să fie asimptotic stabilă). Se va trata mai întâi cazul $m = 1$ și apoi cazul general.

Definiția 0.1. Pentru un sistem (A, B, C, D) dependența

$$u = Fx + Gv \quad (2)$$

se numește lege de comandă prin reacție după stare, unde $F \in \mathcal{R}^{m \times n}$, $G \in \mathcal{R}^{m \times m}$, F se numește matricea de reacție și v este noua mărime de intrare.

Legea de comandă după stare implică accesul (din punct de vedere tehnic) la stare, i.e., cunoașterea mărimii de stare !

După implementarea comenzii (2) sistemul în buclă închisă devine

$$\begin{cases} \dot{x}(t) = (A + BF)x(t) + BGv(t), & x(0) = x_0 \\ y(t) = (C + DF)x(t) + DGv(t) \end{cases} \quad (3)$$

având noua intrare v și ieșirea y .

Perechea (A, B) , $A \in \mathcal{R}^{n \times n}$, $B \in \mathcal{R}^{n \times m}$ se numește stabilizabilă dacă există o matrice de reacție după stare $F \in \mathcal{R}^{m \times n}$ astfel încât $\Lambda(A + BF) \subset \mathcal{C}^-$.

Perechea (C, A) , $C \in \mathcal{R}^{p \times n}$, $A \in \mathcal{R}^{n \times n}$ se numește detectabilă dacă există o matrice de reacție după stare $K \in \mathcal{R}^{n \times p}$ astfel încât $\Lambda(A + KC) \subset \mathcal{C}^-$.

Perechea (A, B) , $A \in \mathcal{R}^{n \times n}$, $B \in \mathcal{R}^{n \times m}$ se numește alocabilă dacă oricare ar fi mulțimea simetrică Λ_0 de n numere complexe (orice $s \in \Lambda_0 \Rightarrow \bar{s} \in \Lambda_0$), există o matrice de reacție după stare $F \in \mathcal{R}^{m \times n}$ astfel încât $\Lambda(A + BF) = \Lambda_0$.

Procedura de alocare: Cazul $m=1$

Pasul 0: Dându-se perechea controlabilă (A, b) , $A \in \mathcal{R}^{n \times n}$, $b \in \mathcal{R}^{n \times 1}$ și un set simetric de n valori proprii s_1, s_2, \dots, s_n se construiește polinomul

$$\chi(s) = \prod_{i=1}^n (s - s_i) = \alpha_0 + \alpha_1 s + \alpha_2 s^2 + \dots + \alpha_{n-1} s^{n-1} + s^n$$

cu $\alpha_0, \alpha_1, \dots, \alpha_n \in \mathcal{R}$.

Pasul 1: Se construiește matricea de controlabilitate

$$R = \begin{bmatrix} b & AB & \dots & A^{n-1}b \end{bmatrix}$$

care este automat pătrată și nesingulară.

Pasul 2: Se rezolvă ecuația

$$R^T q = e_n$$

în necunoscuta $q \in \mathcal{R}^n$.

Pasul 3: Se calculează

$$f^T = -q^T \chi(A).$$

Procedura de alocare: Cazul general

Pasul 0: Dându-se perechea controlabilă (A, B) , $A \in \mathcal{R}^{n \times n}$, $B \in \mathcal{R}^{n \times m}$ și un set simetric de n valori proprii s_1, s_2, \dots, s_n se construiește polinomul

$$\chi(s) = \prod_{i=1}^n (s - s_i) = \alpha_0 + \alpha_1 s + \alpha_2 s^2 + \dots + \alpha_{n-1} s^{n-1} + s^n$$

cu $\alpha_0, \alpha_1, \dots, \alpha_n \in \mathcal{R}$.

Pasul 1: Se aleg $\tilde{F} \in \mathcal{R}^{m \times n}$ și $g \in \mathcal{R}^m$ aleator și se construiesc matricele

$$A_{\tilde{F}} := A + B\tilde{F}, \quad b = Bg.$$

Pasul 2: Se aplică procedura de alocare (cu $m = 1$) perechii $(A_{\tilde{F}}, b)$ și polinomului $\chi(s)$ obținându-se reacția $f \in \mathcal{R}^m$.

Pasul 3: Calculăm reacția finală sub forma

$$F = \tilde{F} + gf^T.$$

Observații:

- a) Orice pereche (A, B) are o subpereche care poate fi alocată (parte din spectru poate fi alocat) - aceasta coincide cu partea controlabilă - și o parte nealocabilă, i.e. anumiți poli fiși. Acești poli sunt invarianți în raport cu orice reacție după stare.
- b) Alocarea se poate extinde și asupra vectorilor proprii (simultan cu valorile proprii) cu anumite precauții privitoare la alegerea acestora.
- c) Metodele de mai sus au numeroase dezavantaje din punct de vedere numeric și de aceea s-au dezvoltat alte proceduri numeric stabile de construcție a reacției F (algoritm de tip Schur, alocare cu normă minimă, etc) câteva din ele fiind prezentate în secțiunea **B**.
- d) Alocabilitatea nu implică în general faptul că matricea $A + BF$ are orice elemente prescrise ci doar orice spectru prescris

B) Testare numerică

În această secțiune vom prezenta două metode de alocare a polilor în cazul multivariabil: prima este o metoda bazată pe forma Schur reală a matricei A a sistemului (A, B, C, D) , iar a doua este o procedură robustă bazată pe minimizarea unei nome.

Proceduri de alocare suboptimală

Un algoritm numeric stabil de alocare a polilor este bazat pe forma Schur a matricei de stare și pe utilizarea transformărilor ortogonale. Sunt modificate numai valorile proprii "bad" ale sistemului.

Fie o partiție a planului complex \mathcal{C}

$$\mathcal{C} = \mathcal{C}_g \oplus \mathcal{C}_b, \quad \mathcal{C}_g \cap \mathcal{C}_b = \emptyset$$

unde \mathcal{C}_g și \mathcal{C}_b specifică regiunile "good", respectiv "bad" ale planului \mathcal{C} . Vom considera numai partiții simetrice relativ la axa reală. Alegem mulțimea Γ de poli pe care dorim să îi alocăm astfel încât $\Gamma \subset \mathcal{C}_g$.

Algoritmul are următorii pași:

1. Se reduce A la forma Schur reală $S = U' * A * U$, unde printr-o reordonare în blocul diagonal dreapta jos se găsesc valorile proprii ce se doresc mutate. Numărul de valori proprii din \mathcal{C}_g este q . Se aplică transformarea asupra lui B : $B = U' * B$.

2. Se setează $H = 0$ și $i = q + 1$.

3. Dacă $i > n$, stop.
4. Se notează cu F ultimul bloc de pe diagonala lui A de dimensiune p ($p = 1$ sau 2) și cu G ultimele p linii din B .
5. Se calculează K care alocă p poli din Γ .
6. Se calculează $A = A + B * [0 \ K]$ și $H = H + [0 \ K] * U$.
7. Se mută ultimul bloc din A pe poziția (i, i) acumulând transformările în U și se calculează $B = U * B$.
8. $i = i + p$ și se reia de la pasul 3.

Proceduri de alocare robustă

O altă categorie de proceduri de alocare o constituie procedurile de alocare robustă. Acestea exploatează libertățile de alegere a matricei de reacție F , existente în cazul sistemelor cu mai multe intrări, în sensul insensibilizării maxime a valorilor proprii alocate la variații atât în datele inițiale (elementele matricelor A și B) cât și în rezultate (elementele matricei F), acestea din urmă fiind posibile în faza de implementare a legii de conducere.

În cazul $m > 1$ problema alocării nu are soluție unică. Gradele de libertate ale alegerii reacțiilor de alocare a valorilor proprii sunt legate de posibilitățile de alocare a direcțiilor (vectorilor) proprii. Alocarea adecvată a direcțiilor proprii joacă un rol important în reducerea sensibilității spectrului sistemului în circuit închis la perturbații numerice în elementele matricelor A , B și F .

Pentru cazul când Λ_c are elemente *distincte* condițiile în care alocarea vectorilor proprii este posibilă sunt date de următorul rezultat.

Teorema 0.1. Presupunem că mulțimea $\Lambda_c = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ are elementele distincte și fie $x_j \in \mathbb{C}^n$, $j = 1 : n$, n vectori nenuli. Atunci există o matrice $F \in \mathbb{R}^{m \times n}$ astfel încât

$$(A - BF)x_j = \lambda_j x_j, \quad j = 1 : n \quad (4)$$

dacă și numai dacă

- i) vectorii x_j $j = 1 : n$ sunt liniar independenți, i.e. matricea

$$X = [x_1 \ x_2 \ \dots \ x_n] \quad (5)$$

este nesingulară;

- ii) $\lambda_i = \bar{\lambda}_j$ dacă și numai dacă $x_i = \bar{x}_j$;

iii)

$$(\lambda_j I - A)x_j \in \text{Im} B, \quad j = 1 : n. \quad (6)$$

Dacă B este monică atunci matricea F este unic determinată.

Considerând B monică, pentru a obține matricea F considerăm mai întâi factorizarea QR a matricei B

$$B = QR = [Q_1 \ Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1 \quad (7)$$

cu $Q \in \mathcal{R}^{n \times n}$ ortogonală și $R_1 \in \mathcal{R}^{m \times m}$ superior triunghiulară, nesară. Atunci, definind

$$\Lambda \stackrel{\text{def}}{=} \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \quad (8)$$

cu notația (4), relația (3) se scrie

$$(A - BF)X = X\Lambda. \quad (9)$$

Ținând seama de (6), (??) se descompune în

$$Q_2^T (X\Lambda - AX) = 0 \quad (10)$$

și

$$F = R_1^{-1} Q_1^T (A - X\Lambda X^{-1}). \quad (11)$$

Relația (??) transferă problema fixării unei soluții a problemei de alocare a valorilor proprii într-o problemă de selecție a matricei X a vectorilor proprii care este supusă restricțiilor 5. Aceste restricții se pot scrie și în forma

$$x_j \in \mathcal{S}_j = \text{Ker} [Q_2^T (\lambda_j I - A)], \quad j = 1 : n. \quad (12)$$

În cazul în care perechea (A, B) este controlabilă subspațiul liniar \mathcal{S}_j din (??) are dimensiunea $\dim \mathcal{S}_j = m$.

Cu scopul de a aprecia sensibilitatea valorilor proprii ale unei matrice $A \in \mathcal{R}^{n \times n}$ considerăm matricea $\bar{A} = A + E$ unde $E \in \mathcal{R}^{n \times n}$ este o matrice de perturbație cu $\varepsilon = \|E\|$ și $E = \varepsilon G$ cu $\|G\| = 1$. Dacă $\lambda_i \in \lambda(A) \cap \mathcal{R}$ și $x_i, y_i \in \mathcal{R}^n$ sunt vectorii proprii ai matricelor A și A^T asociați lui λ_i notăm cu $\lambda_i(\varepsilon) \in \lambda(A + \varepsilon G)$ valoarea proprie obținută prin modificarea lui λ_i și $x_i(\varepsilon), y_i(\varepsilon)$ vectorii proprii asociați.

Definim *sensibilitatea* sau *numărul de condiție* al valorii proprii λ_i în raport cu perturbații mici în elementele matricei A prin

$$c_i = \left| \frac{d\lambda_i(\varepsilon)}{d\varepsilon} \right|_{\varepsilon=0} \stackrel{\text{def}}{=} |\lambda_i'(0)|. \quad (13)$$

O margine superioară a lui c_i este dată de

$$(c_i)_{\max} = \frac{\|y_i\| \|x_i\|}{|y_i^T x_i|} \quad (14)$$

a cărei valoare depinde exclusiv de direcțiile proprii definite de vectorii proprii x_i și y_i și care își atinge valoarea minimă în $(c_i)_{\max} = 1$ în cazul în care acești vectori sunt coliniari. Evident, o valoare proprie este cu atât mai puțin sensibilă la perturbații, i.e. este mai *robustă*, cu cât numărul său de condiție este mai mic și este esențial faptul că valoarea acestuia este determinată de poziția relativă a vectorilor proprii asociați.

Robustetea *întregului* spectru de valori proprii $\lambda(A)$ al unei matrice $A \in \mathcal{R}^{n \times n}$ poate fi apreciată, e.g. printr-o măsură a vectorului $c = [c_1 \ c_2 \ \dots \ c_n]^T$ al numerelor de condiție ale valorilor proprii individuale, cum ar fi $\nu_1 = \|c\|$.

Putem adopta ca măsură a robusteții spectrului de valori proprii al unei matrice simple (i.e. având un set complet de vectori proprii liniar independenți)

numărul de condiție la inversare al matricei vectorilor proprii $\kappa_2(X)$ și formula următoarea problemă de alocare robustă:

Date perechea (A, B) cu $m > 1$ și mulțimea simetrică Λ să se determine matricea de reacție F astfel încât să aibă loc alocarea $\lambda(A - BF) = \Lambda_c$ și, simultan, robustețea spectrului Λ al matricei $A - BF$ să fie maximă, i.e. $\kappa_2(X_{A-BF})$ să fie minimă.

Fără a reduce generalitatea problemei putem presupune că matricea B este monică. În această situație, cu factorizarea (6) a lui B vectorii proprii ai matricei $A - BF$ aparțin subspațiilor liniare \mathcal{S}_j , $j = 1 : n$ definite în (??).

În cazul special $m = n$ avem $\mathcal{S}_j = \mathcal{C}^n$ și, deci, vectorii proprii pot fi alocați arbitrar. Rezultă că putem obține robustețea maximă posibilă a spectrului de valori proprii alegând matricea X unitară, în particular, din motive de simplitate și eficiență,

$$x_k = e_k \quad (15)$$

pentru $\lambda_k \in \Lambda_c \cap \mathcal{R}$, respectiv

$$x_{k,k+1} = \frac{1}{\sqrt{2}}(e_k \pm ie_{k+1}) \quad (16)$$

pentru $\lambda_{k,k+1} = \alpha_k \pm i\beta_k \in \Lambda \cap (\mathcal{C} \setminus \mathcal{R})$. Este ușor de văzut că alegerea de mai sus conduce la

$$L \stackrel{\text{def}}{=} X\Lambda X^H = \text{diag}(\Lambda_1, \Lambda_2, \dots, \Lambda_q) \quad (17)$$

cu $\Lambda_k = \lambda_k$ pentru valorile proprii reale și $\Lambda_k = \begin{bmatrix} \alpha_k & \beta_k \\ -\beta_k & \alpha_k \end{bmatrix}$ pentru perechile de valori proprii complex conjugate. Matricea de reacție care realizează o alocare robustă optimă, prin alegerea (10), (11) a vectorilor proprii, conform (??), este

$$F = B^{-1}(A - L). \quad (18)$$

În cazul practic $1 < m < n$ subspațiile \mathcal{S}_j , $j = 1 : n$ sunt strict proprii în \mathcal{C}^n și, în consecință, alocarea robustă se reduce, în principal, la următoarea problemă.

Să se determine un set simetric de n vectori liniar independenți (de normă euclidiană unitară) x_k , $k = 1 : n$ din \mathcal{S}_j , $j = 1 : n$ astfel încât numărul de condiție

$$\nu_2 \stackrel{\text{def}}{=} \kappa_2(X) = \|X\|_2 \|X^{-1}\|_2 \quad (19)$$

al matricei $X = [x_1 \ x_2 \ \dots \ x_n]$ să fie minim.

Pentru simplitatea expunerii, vom considera că mulțimea Λ este reală.

Fie $\mathcal{S}_j \in \mathcal{R}^{n \times m}$ o bază ortonormală a subspațiului \mathcal{S}_j , $j = 1 : n$. Atunci orice vector $x_j \in \mathcal{S}_j$ se scrie sub forma $x_j = \mathcal{S}_j d_j$ cu $d_j \in \mathcal{R}^m$ iar matricea X din (13) capătă expresia

$$X = SD, \quad (20)$$

unde

$$S = [S_1 \ S_2 \ \dots \ S_n] \in \mathcal{R}^{n \times mn}, \quad (21)$$

$$D = \text{diag}(d_1, d_2, \dots, d_n) \in \mathcal{R}^{mn \times n}. \quad (22)$$

Putem reformula problema de mai sus în următorii termeni.

Fiind dată matricea S cu structura (??), să se găsească X din (??) care minimizează ν_2 din (13) în raport cu toți D de forma (??).

Cea mai simplă metodă de a măsura abaterea de la ortogonalitate, din punct de vedere al implementării, urmărește, la pasul curent al unei iterații fixate să determine acel $x_j \in \mathcal{S}_j$ care este "cel mai ortogonal" pe subspațiul liniar

$$\chi_j = \text{Im} X_j = \text{Im} \begin{bmatrix} x_1 & x_2 & \dots & x_{j-1} & x_{j+1} & \dots & x_n \end{bmatrix} \quad (23)$$

generat de ceilalți vectori proprii alocați, temporar, în faza de calcul respectivă. Această extremizare poate fi făcută prin tehnici standard cum sunt factorizarea **QR** sau descompunerea valorilor singulare. Fie, astfel,

$$X_j = Q_j R_j = \begin{bmatrix} \bar{Q}_j & \bar{y}_j \end{bmatrix} \begin{bmatrix} \bar{R}_j \\ 0 \end{bmatrix} \quad (24)$$

factorizarea **QR** a matricei $X_j \in \mathcal{R}^{n \times (n-1)}$ din (14), unde X_j are coloanele liniar independente, deci coloanele lui $\bar{Q}_j \in \mathcal{R}^n$ formează o bază ortogonală a subspațiului χ_j iar $\bar{y}_j \in \mathcal{R}^n$ generează $\mathcal{Y}_j = \chi_j^\perp$. Vectorul x_j , conținut în \mathcal{S}_j care maximizează unghiul față de χ_j , respectiv minimizează unghiul față de $\mathcal{Y}_j = \text{Im} \bar{y}_j$. În forma normalizată, acest vector este dat de

$$x_j^{nou} = \frac{S_j d_j^*}{\|S_j d_j^*\|} \quad (25)$$

unde d_j^* este soluția, în sensul cmm, a sistemului supradeterminat $S_j d_j = \bar{y}_j$, cu care devine

$$x_j^{nou} = \frac{S_j S_j^T \bar{y}_j}{\|S_j S_j^T \bar{y}_j\|}. \quad (26)$$

Trebuie reținut faptul că optimizarea condiției numerice c_j a lui λ_j alterează numerele de condiție ale celorlalte valori proprii și procesul iterativ nu este, în mod necesar, convergent către un optim global. Totuși, experiențele numerice au arătat că această procedură dă bune rezultate practice pentru un număr relativ redus de iterații.

Matricea vectorilor proprii $X \stackrel{\text{def}}{=} X_0$, de inițializare a procesului iterativ, poate fi orice set de n vectori liniar independenți, câte unul din fiecare subspațiu \mathcal{S}_j , $j = 1 : n$.

Algoritmul 0.1 (Date matricea $S \in \mathcal{R}^{n \times nm}$ din (??) a bazelor ortonormale $S_j = S(:, m(j-1) + 1 : mj)$ pentru subspațiile liniare \mathcal{S}_j , $j = 1 : n$, numărul maxim de iterații n_{iter} și toleranța tol pentru aprecierea convergenței procesului iterativ, procedura calculează un set X de n vectori liniar independenți din \mathcal{S}_j având gradul de ortogonalitate maxim, respectiv valoarea numărului de condiție $\kappa_2(X)$ minimă.

1. $X = S(:, 1 : m : nm)$
2. $\kappa_v = \kappa_2(X)$
3. Pentru $j = 1 : n$
 1. $S_j = S(:, m(j-1) + 1 : mj)$
4. Pentru $k = 1 : n_{iter}$
 1. Pentru $j = 1 : n$

1. $X_j = \begin{bmatrix} X(:, 1:j-1) & X(:, j+1:n) \end{bmatrix}$
2. Se calculează ultima coloană $\bar{y}_j = Q_j(:, n)$ a matricei ortogonale Q_j din factorizarea **QR** (15) a matricei X_j .
3. Se calculează x_j^{nou} din (??) conform secvenței
 1. $w = S_j^T \bar{y}_j$
 2. $w = S_j w$
 3. $x_j^{nou} = w / \|w\|$
4. $X = \begin{bmatrix} X(:, 1:j-1) & x_j^{nou} & X(:, j+1:n) \end{bmatrix}$
2. $\kappa = \kappa_2(X)$
3. Dacă $\kappa \geq \kappa_v$ sau $|\kappa - \kappa_v| < tol$ atunci return X
altfel $\kappa_v = \kappa$.

Procedura constituie punctul cheie al variantei de alocare robustă a polilor în cazul $1 < m < n$.

Algoritmul 0.2 (Se consideră date: perechea controlabilă (A, B) cu $A \in \mathcal{R}^{n \times n}$ și $B \in \mathcal{R}^{n \times m}$, $1 < m \leq n$, monică; mulțimea Λ a valorilor proprii impuse; numărul maxim de iterații n_iter și toleranța tol).

Algoritmul calculează matricea de reacție $F \in \mathcal{R}^{m \times n}$ și matricea vectorilor proprii $X \in \mathcal{R}^{n \times n}$ a sistemului în circuit închis astfel încât $\lambda(A - BF) = \Lambda$ și X să aibă numărul de condiție $\kappa_2(X)$ minim.)

1. $\Lambda = diag(\lambda_1, \lambda_2, \dots, \lambda_n)$
2. Dacă $m = n$ atunci
 1. $X = I_n$
 2. Se rezolvă sistemul liniar matriceal $BF = A - \Lambda$ în raport cu F (vezi ??)
 altfel
 1. Se calculează factorizarea **QR** (6) a matricei B
 2. $Q_1 = Q(:, 1:m)$, $Q_2 = Q(:, m+1:n)$
 3. $R_1 = R(1:m, :)$
 4. Se construiesc bazele ortogonale S_j ale subspațiilor liniare S_j , $j = 1:n$ definite de (??)
 5. $S = \begin{bmatrix} S_1 & S_2 & \dots & S_n \end{bmatrix}$
 6. Se calculează setul optim de vectori proprii pentru sistemul în circuit închis $X = aloc_vec(S, n_iter, tol)$
 7. Se calculează matricea A_0 a sistemului în circuit închis prin rezolvarea sistemului liniar matriceal $A_0 X = X \Lambda$
 8. Se calculează matricea de reacție F , conform (??), prin rezolvarea sistemului matriceal superior triunghiular $R_1 F = Q_1^T (A - A_0)$.

C) Sarcini de lucru

- (i) Dacă perechea (A_c, b_c) a unui sistem cu o singură intrare este în forma standard controlabilă

$$A_c = \begin{bmatrix} -\alpha_1 & \dots & -\alpha_{n-1} & -\alpha_n \\ 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix}, \quad b_c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

iar polinomul caracteristic dorit $p_0(s)$ pentru sistemul în circuit închis este

$$p_0(s) = \prod_{k=1}^n (s - \lambda_k) = a_n + a_{n-1}s + a_2s^2 + \dots + a_1s^{n-1} + s^n,$$

arătați că matricea de reacție f_c^T care asigură alocarea dorită are componentele

$$f_c(i) = a_i - \alpha_i, \quad i = 1 : n.$$

Aceasta sugerează următoarea procedură de alocare pentru o pereche controlabilă (A, b) dată.

1. Se aduce perechea (A, b) la forma standard controlabilă printr-o transformare nesingulară de asemănare $(A_c, b_c) = (TAT^{-1}, Tb)$ cu calculul matricei de transformare T .
2. Se calculează matricea de reacție f_c^T folosind relațiile de mai sus.
3. $f^T = f_c^T T$.

Indicație. Arătați că matricea de controlabilitate este $T = R_c R^{-1}$, unde R și R_c sunt matricele de controlabilitate ale perechilor (A, b) și, respectiv, (A_c, b_c) .

- (ii) Scrieți un program MATLAB eficient pentru implementarea algoritmului de alocare a polilor pentru o pereche controlabilă (A, b) dată.

Indicație. Matricea de controlabilitate a unei perechi (A, b) în *forma superior Hessenberg* este superior triunghiulară și nu trebuie calculată efectiv întrucât rezolvarea sistemului $R^T g = e_n$ se reduce la calculul unui singur element nenul.

Se aduce perechea (A, b) la forma *FSH*

Dacă $b_1 \neq 0$ și $a_{j+1,j} \neq 0, j = 1 : n - 1$ atunci

1. $f_n = \frac{1}{b_1 \prod_{j=1}^{n-1} a_{j+1,j}}$
2. $k = 1$
3. Cât timp $k \leq n$
 1. Dacă $\beta_k \neq 0$ atunci

1. $w = \alpha_k^2 + \beta_k^2$
 2. $g^T = f^T A; h^T = g^T A$
 3. $f^T = h^T - 2\alpha_k g^T + w f^T$
 4. $k = k + 2$
- altfel**
1. $f^T \leftarrow f^T A - \lambda_k f^T$
 2. $k = k + 1$
 4. $f^T \leftarrow f^T U$

(iii) Implementați în MATLAB algoritmul de alocare bazat pe forma Schur.

(iv) Implementați în MATLAB algoritmul de alocare cu minimizare de normă.

0.2 Estimatori de stare

A) Descriere teoretică

Așa cum am văzut, legea de comandă cu reacție după stare poate asigura în mod satisfăcător cerințele fundamentale ale sintezei sistemelor de stabilizare și alocare cu condiția ca starea să fie disponibilă pentru măsură (să fie accesibilă din punct de vedere tehnic). Din păcate acest lucru nu este în general posibil și atunci ne punem problema estimării cât mai exacte a stării unui sistem dinamic prin construcția unui nou sistem care să citească mărimile accesibile sau măsurabile (intrarea u și ieșirea y) și care să genereze la ieșire o estimare a stării \hat{x} . Un astfel de sistem se numește estimator de stare (FIG).

Dându-se sistemul (A, B, C, D) , dorim să construim un nou sistem

$$\begin{cases} \dot{\hat{w}}(t) = J\hat{w}(t) + H y(t) + M u(t) \\ \hat{x}(t) = K \hat{w}(t) + N y(t) + P u(t) \end{cases} \quad (27)$$

care să îndeplinescă simultan următoarele 2 condiții:

1. Să fie intern asimptotic stabil, i.e. $\Lambda(J) \subset \mathcal{C}_-$.
2. $\lim_{t \rightarrow \infty} (\hat{x}(t) - x(t)) = 0$, i.e. ieșirea $\hat{x}(t)$ (numită estimarea stării) să aproximeze asimptotic starea sistemului original $x(t)$.

Observații:

a) Ideal ar fi $x(t) = \hat{x}(t)$, $\forall t$. Acest lucru nu este însă în general posibil și atunci această cerință se relaxează la cea asimptotică.

b) Cerința de stabilitate internă este esențială pentru construcția estimatorului.

Punând în evidență "fidelitatea" cu care starea estimatorului urmărește starea sistemului original obținem

$$\begin{cases} (w - Vx)' &= J(w - Vx) + (JV + HC - VA)x + (M - VB + HD)u \\ \hat{x} - x &= K(w - Vx) + (KV + NC - I)x + (P + ND)u \end{cases}$$

Pentru ca starea estimatorului să urmărească asimptotic starea sistemului original (condiția 2) trebuie ca ieșirea sistemului dinamic de mai sus să tindă asimptotic la zero (indiferent de inițializări și de semnalele de intrare x și u). Acest lucru este posibil dacă satisfacem simultan următoarele condiții:

- a) J asimptotic stabilă;
- b) $JV + HC - VA = 0$;
- c) $M - VB + HD = 0$;
- d) $KV + NC - I = 0$;
- e) $P + ND = 0$.

Problema de construcție a estimatorului asimptotic stabil s-a redus la problema algebrică de satisfacere simultană a condițiilor a) - e): avem 5 condiții (4 ecuații algebrice și o locație de spectru) și 7 necunoscute.

Evident c) și e) se pot satisface automat alegând $M := VB - HD$ și $P := -ND$. Rămân în continuare mai multe grade de libertate în satisfacerea restului de condiții

$$\begin{array}{rcl} \Lambda(J) & \subset & \mathcal{C} \\ JV + HC - VA & = & 0 \\ KV + NC - I & = & 0 \end{array}$$

funcție de care se deosebesc mai multe tipuri de estimatori.

Estimatori de tip 1: $N=0$

Condiția este echivalentă cu a spune că estimatorul nu are transfer direct I/O, i.e., matricea sa "D" este zero. În acest caz obținem

$$\begin{array}{rcl} JV + HC - VA & = & 0 \\ KV & = & I \end{array}$$

și putem alege $K = I$, $V = I$, rămânând de satisfăcut doar condiția

$$J = A - HC, \quad \Lambda(J) \subset \mathcal{C}_-.$$

Această condiție se poate satisface automat dacă perechea (C, A) este detectabilă, caz în care alegem $-H$ egal cu reacția care stabilizează (problema de stabilizare pentru perechea (A^T, C^T)). Mai mult, dacă perechea (C, A) este observabilă, atunci spectrul matricei de stare J a estimatorului poate fi alocat arbitrar.

Estimatorul rezulta de forma

$$\left\{ \begin{array}{l} \dot{\hat{w}}(t) = (A - HC)w(t) + Bu(t) - HDu(t) + Hy(t) \\ \hat{x}(t) = w(t) \end{array} \right\} \quad (28)$$

sau forma echivalentă

$$\dot{\hat{x}} = A\hat{x} + Bu + L(C\hat{x} + Du - y) \quad (29)$$

în care este cunoscut sub numele de estimator Luenberger.

Observatii:

a) Dimensiunea unui estimator de tip Luenberger (de tip 1) este egală cu n (dimensiunea spațiului stărilor sistemului original (A, B, C, D)). În anumite situații se pot construi estimatoare de dimensiune mai mică și se poate formula problema de construcție a unui estimator de dimensiune minimală (legată de teoria estimatoarelor de tipul 2).

b) Am văzut că un sistem poate fi stabilizat/alocat cu o reacție constantă după stare. Deoarece starea nu este în general cunoscută am construit mai sus un sistem (numit estimator) care estimează asimptotic starea sistemului pe baza informațiilor furnizate de semnalele de intrare u și de ieșire y . Întrebarea naturală este dacă putem stabiliza/aloca sistemul original prin implementarea reacției constante după starea estimată \hat{x} (în locul stării reale x care nu este accesibilă)? Răspunsul este pozitiv obținându-se compensatorul Kalman.

Estimatori de ordin redus

Fie un sistem (A, B, C, D) cu m intrări, p ieșiri și ordinul n . Dacă dorim estimarea stării acestui sistem nu este nevoie să construim un estimator care să estimeze direct toate cele n stări întrucât o parte dintre acestea rezultă automat pe baza ieșirilor. Întrădevar, să presupunem că C are rangul egal cu numărul de ieșiri p . Atunci, cunoscând ieșirea $y(t)$ și estimând numai $n - p$ stări rezultă că celelate p stări se pot calcula din ecuațiile corespunzătoare ale ieșirii. Un astfel de estimator de ordin redus se numește estimator de tipul 2.

Pentru construcția estimatorului de tipul 2 facem ipotezele:

i) Matricea C este epică, i.e., $\text{rank} C = p$;

ii) Matricea C are forma $C = \begin{bmatrix} O & I_p \end{bmatrix}$.

Observații:

a) Ipoteza i) nu constituie o restricție majoră întrucât o putem asigura în etapa de modelare printr-o alegere judicioasă a ieșirilor sistemului. În cazul în care prin modelare nu s-a asigurat îndeplinirea ei, se poate face o schimbare de variabile în spațiul mărimilor de ieșire punându-se în evidență anumite ieșiri identice zero sau redundante ce pot fi eliminate rămânând o matrice C ce satisface ipoteza.

b) Ipoteza ii) se poate asigura printr-o simplă transformare de coordonate în spațiul stărilor (presupunând ipoteza i) adevărată). c) Ipotezele i) și ii) implică că

$$y = Cx + Du = \begin{bmatrix} O & I_p \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du = x_2 + Du$$
 deci stările x_2 în număr de p sunt automat cunoscute (prin cunoașterea intrărilor și ieșirilor) și rămân de estimat stările x_1 în număr de $n - p$.

Folosind forma specială a lui C în ecuațiile unui estimator general, obținem pentru un estimator de tipul 2 următoarea construcție:

Pasul 0: Dându-se sistemul (A, B, C, D) cu perechea (C, A) detectabilă (observabilă) și matricea C epică, se găsește o transformare de similaritate T a.i. în noul sistem de coordonate să avem (refolosind notațiile)

$$A = \begin{bmatrix} A_1 & A_3 \\ A_2 & A_4 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad C = [O \quad I_p]$$

în care perechea (A_2, A_1) este detectabilă (observabilă).

Pasul 1: Se folosește procedura de alocare pentru perechea (A_2, A_1) și se determină o matrice V_2 a.i. $A_1 + V_2 A_2$ să aibă spectrul dorit.

Pasul 2: Se calculează estimatorul cu parametrii

$$J = A_1 + V_2 A_2, \quad H = A_3 + V_2 A_4 - A_1 V_2 - V_2 A_2 V_2, \quad M = B_1 + V_2 B_2 - H D,$$

$$K = \begin{bmatrix} I_{n-p} \\ O \end{bmatrix}, \quad N = \begin{bmatrix} -V_2 \\ I_p \end{bmatrix}, \quad V = [I_{n-p} \quad V_2], \quad P = \begin{bmatrix} V_2 D \\ -D \end{bmatrix}$$

rezultând

$$\begin{cases} \dot{w}(t) &= (A_1 + V_2 A_2) w(t) + H y(t) + (B_1 + V_2 B_2 - H D) u(t) \\ \hat{x}(t) &= \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{bmatrix} = \begin{bmatrix} I_{n-p} \\ O \end{bmatrix} w(t) + \begin{bmatrix} -V_2 \\ I_p \end{bmatrix} y(t) + \begin{bmatrix} V_2 D \\ -D \end{bmatrix} u(t) \end{cases} \quad (30)$$

Pasul 3: Se "reversează" corespunzător transformarea de similaritate asupra estimatorului, obținându-se în final un estimator de tip 2 (de ordin $n - p$) pentru sistemul original.

Problema naturală care se pune este dacă pe baza acestui estimator putem construi în continuare un compensator care stabilizează intern - exact cum am făcut în cazul estimatorului de tip 1 obținând compensatorul Kalman.

B) Sarcini de lucru

(i) Considerăm un motor de curent continuu

pentru care avem următorii parametri:

- momentul de inerție al rotorului $J = 5E - 5kg * m^2/s^2$
- factorul de amortizare al sistemului mecanic $b = 1E - 4Nm/s$
- constanta motorului $K = Ke = Kt = 0.1Nm/Amp$
- rezistența din circuitul de comanda pe indus $R = 1ohm$
- rezistența din circuitul de comanda pe indus $L = 1E - 3H$
- intrare V : tensiunea de alimentare a înfășurării rotorului
- iesire θ : poziția unghiulară a arborelui motorului
- rotorul și arborele sunt presupuse rigide.

Cuplul motor, T , depinde de curentul din armături, i_a , printr-un factor constant K_t . Tensiunea contra-electromotoare indusă în înfășurarea rotorului e , este proporțională cu viteza de rotație

$$T = K_t i_a, \quad e = K_e \dot{\theta}$$

Modelul matematic al motorului este dat de ecuațiile

$$J\ddot{\theta} + b\dot{\theta} = Ki_a$$

$$L\frac{di_a}{dt} + Ri = V - K\dot{\theta}$$

Aceste ecuații dau o reprezentare pe stare. Dacă alegem curentul dintre armături, poziția motorului și viteza motorului ca variabile de stare, putem scrie

$$\begin{bmatrix} \dot{i}_a \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & 0 & -\frac{K}{L} \\ 0 & 0 & 1 \\ \frac{K}{J} & 0 & -\frac{b}{J} \end{bmatrix} \begin{bmatrix} i_a \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} V$$

$$\theta = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ \theta \\ \dot{\theta} \end{bmatrix}$$

% parametrii motorului

L = 1e-3; R = 1; J = 5e-5; B = 1e-4; K = 0.1;

% reprezentarea pe stare a ecuatiilor motorului

A = [-R/L, 0, -K/L; 0, 0, 1; K/J, 0, -B/J];

B = [1/L; 0; 0];

C = [0, 1, 0];

D = [0];

- (i) Proiectați un estimator Luenberger plasând polii lui $A+LC$ în $-500+j250, -500-j250, -200$.

% verificam daca sistemul este observabil

O = obsv(A,C);rank(O)

% proiectam un estimator plasand polii lui A-LC in -500+j250, -500-j250, -200

Lt = acker(A.',C.',[-500+250j, -500-250j, -200]);

L = Lt'

% verificam plasarea polilor

est_poles = eig(A - L*C)

% redefinim C si D pentru a avea la iesire toate starile motorului -

% vom folosi doar y = x_2 pentru estimator

C = [1, 0, 0; 0, 1, 0; 0, 0, 1];

D = [0; 0; 0];

% definim conditiile initiale pentru a fi utilizare in Simulink

x0 = [0, 0, 0];

% definim modelul pe stare al estimatorului

Ahat = A;

Bhat = [B, L];

Chat = [1, 0, 0; 0, 1, 0; 0, 0, 1];

Dhat = [0, 0; 0, 0; 0, 0];

xhat0 = [5, 5, 5];

- (ii) Creați schema Simulink a sistemului și a estimatorului.

simulink_DCmotor-eps-converted-to.pdf

- (iii) Reprezentați grafic stările și estimatele lor.

```
figure(1);
subplot(3,1,1); plot(tout, states(:,1), tout, states(:,4), '--');
xlabel('timp (sec)'); legend('x 1 = i a', 'xhat 1');
title('Starile x si estimarile xhat');
subplot(3,1,2); plot(tout, states(:,2), tout, states(:,5), '--');
xlabel('timp (sec)'); legend('x 2 = theta', 'xhat 2');
subplot(3,1,3); plot(tout, states(:,3), tout, states(:,6), '--');
xlabel('timp (sec)'); legend('x 3 = theta dot', 'xhat 3');
```

estimare_stari-eps-converted-to.pdf

- (ii) Construiți un estimator de ordin redus pentru sistemul (A, B, C, D) dat de

$$A = \begin{bmatrix} -4 & 1 & 0 & 0 & 0 \\ -5 & 0 & 1 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -4 & 1 \\ -2 & 0 & 0 & -4 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Pasul 0: Verificăm observabilitatea perechii (C, A) :

```
>> Q=obsv(A,C);
>> r=rank(Q)
```

r =

5

Rangul este 5, deci perechea este observabilă.

Căutăm matricea de transformare inversabilă T astfel încât matricea C să aibă forma $C = [O \ I_p]$. T se alege de forma $T = \begin{bmatrix} \bar{C} \\ C \end{bmatrix}$, unde \bar{C} este de dimensiune $(n - p \times n)$.

Alegem $\overline{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$ și prin urmare vom obține $T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$.

```
Cbar = [0 0 0 0 1;0 0 1 0 0;0 1 0 0 0];
T=[Cbar; C];
```

Folosind transformarea T și re folosind notațiile vom obține

```
>> A=T*A*inv(T)
```

A =

```

      0      0      0     -2     -4
      0      0      0     -2      0
      0      1      0     -5      0
      0      0      1     -4      0
      1      0      0     -1     -4
```

```
B>> B=T*B
```

B =

```

      0      0
      0      0
      0      0
      0      1
     -1     -1
```

```
>> C=C*inv(T)
```

C =

```

      0      0      0      1      0
      0      0      0      0      1
```

Perechea (A_2, A_1) este

```
>> A1 = A(1:3,1:3)
```

A1 =

```

      0      0      0
      0      0      0
      0      1      0
```

```
>> A2 = A(4:5,1:3)
```

A2 =

```

      0      0      1
      1      0      0
```

Dacă perechea inițială (C, A) este observabilă, atunci automat (A_2, A_1) este observabilă.

Pasul 1: Vrem să alocăm polii perechii (A_2, A_1) în $-1, -2, -3$. Obținem astfel matricea V_2 astfel încât $A_1 + V_2 A_2$ să aibă spectrul dorit.

```
>> V2 = place(A1', A2', [-1 -2 -3]);
>> V2 = -V2';
>> eig(A1+V2*A2)
```

ans =

```
-3.0000
-2.0000
-1.0000
```

Pasul 2: Calculăm parametrii estimatorului:

```
>> A3 = A(1:3, 4:5);
>> A4 = A(4:5, 4:5);
>> B1 = B(1:3, :);
>> B2 = B(4:5, :);
>> D = [0 0; 0 0];
>> n = 5;
>> p = 2;
>> J = A1+V2*A2;
>> H = A3+V2*A4-A1*V2-V2*A2*V2;
>> M=B1+V2*B2-H*D;
>> K = [eye(n-p); zeros(p, n-p)];
>> N = [-V2; eye(p)];
>> V = [eye(n-p) V2];
>> P = [V2*D; -D];
```

Pasul 3: "Reversarea" transformării de similaritate

```
>> K = T\K;
>> N = T\N;
>> P = T\P;
```

0.3 Compensatorul Kalman, reglarea sistemelor dinamice

A) Descriere teoretică

Compensatorul Kalman

Fie (A, B, C, D) un sistem dinamic și presupunem pentru problema de stabilizare cu reacție dinamică după ieșire că (A, B) este stabilizabilă și (C, A) este detectabilă, iar pentru problema cu alocare dinamică după ieșire că (A, B) este controlabilă și (C, A) este observabilă.

Considerăm un compensator cu reacție constantă F după starea estimată de către un estimator Luenberger descris de ecuațiile (17). Obținem atunci ecuațiile dinamice ale compensatorului - numit compensator Kalman -

$$\begin{cases} \dot{\hat{x}} &= (A + LC)\hat{x} + Bu + LDu - Ly \\ u &= F\hat{x} \end{cases} \quad (31)$$

sau inca

$$\begin{cases} \dot{\hat{x}} &= (A + LC + BF + LDF)\hat{x} - Ly \\ u &= F\hat{x} \end{cases} \quad (32)$$

avand matricea de transfer

$$K(s) := \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} = \begin{bmatrix} A + LC + BF + LDF & -L \\ F & O \end{bmatrix}, \quad u = Ky.$$

Matricele F și L se aleg a.i. $A + BF$ și $A + LC$ să fie asimptotic stabile (și eventual cu spectru impus).

Conectand compensatorul Kalman în reacție inversă cu sistemul original și aplicand transformarea de similaritate $T := \begin{bmatrix} I & O \\ -I & I \end{bmatrix}$ obținem pentru matricea de stare a sistemului rezultat în buclă închisă

$$TA_R T^{-1} = \begin{bmatrix} A + BF & BF \\ O & A + LC \end{bmatrix}.$$

Observație:

a) Polii sistemului în buclă închisă sunt dați de reuniunea polilor alocați ai lui $A + BF$ prin reacția după stare F cu polii alocați de estimator $\Lambda(A + LC)$ prin reacția L . Cele două alocări se pot face independent, punându-se astfel în evidență celebrul principiu al separației.

Reglarea sistemelor dinamice

Fie sistemul $(A, B, C, D = 0)$ cu matricea de transfer $T(s)$. Facem ipotezele că (A, B) este stabilizabilă și (C, A) detectabilă, care sunt condiții necesare și suficiente pentru existența unui compensator stabilizator.

Considerăm o schemă de reglare cu compensatorul pe calea directă, care are matricea de transfer $K(s)$.

Conform principiului modelului intern, pentru a regla la referințe de tip treaptă trebuie ca modelul treptei $M(s) = \frac{1}{s}I_p$ să fie inclus în matricea de transfer în buclă deschisă, în cazul nostru $L(s) := T(s)K(s)$. Cum $T(s)$ este dat, impunem ca modelul referinței să fie inclus în $K(s) = \tilde{K}(s)M(s)$.

Pentru a asigura stabilizarea sistemului, K trebuie să stabilizeze intern T , problemă echivalentă cu a stabili sistemul $\tilde{T}(s) = MT$ cu \tilde{K} . În consecință proiectăm un regulator \tilde{K} pentru \tilde{T} după care $K(s) = \tilde{K}(s)M(s)$.

Considerăm o realizare de stare pentru treaptă data prin $(A_M, B_M, C_M, D_M) = (0, I_p, I_p, 0)$. Atunci, o realizare de stare pentru sistemul \tilde{T} se obține automat înseriind $T(s)$ cu $M(s)$:

$$\tilde{T} = \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array} \right] = \left[\begin{array}{cc|c} A & O & B \\ -C & O & O \\ \hline O & I_p & O \end{array} \right]. \quad (33)$$

Pentru scrierea unui compensator stabilizator pentru \tilde{T} putem aplica schema standard de scriere a unui compensator Kalman (sau orice schemă de stabilizare cu estimator de stare și reacție după starea estimată). Întrucât matricea \tilde{C} este epică preferăm să scriem un estimator de ordin redus și să luăm o reacție după starea estimată obținând în final un sistem compensator de ordin mai mic decât $\tilde{T}(s)$ (și de același ordin cu $T(s)$).

Obținem pentru ecuațiile estimatorului de ordin redus

$$\left\{ \begin{array}{l} \dot{w} = (A + LC)w + (A + LC)L\tilde{y} + Bu \\ \hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} I_{n-p} \\ O \end{bmatrix} w + \begin{bmatrix} L \\ I_p \end{bmatrix} \tilde{y} \end{array} \right. \quad (34)$$

Un compensator stabilizator se obține atunci luând o reacție stabilizantă după starea estimată de forma

$$u = \begin{bmatrix} F & \tilde{F} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = F(w + L\tilde{y}) + \tilde{F}\tilde{y}. \quad (35)$$

O astfel de reacție stabilizantă există dacă și numai dacă perechea (\tilde{A}, \tilde{B}) este stabilizabilă. Acest lucru nu este însă automat îndeplinit decât dacă facem ipoteza că modelul referinței (polii $s = 0$) nu este simplificat de modelul sistemului $T(s)$.

O condiție suficientă ca să nu apară această simplificare este ca sistemul T să nu aibă zerouri în $s = 0$ ceea ce este asigurat de impunerea condiției

$$\text{rank} \begin{bmatrix} -A & -B \\ C & O \end{bmatrix} = n + p. \quad (36)$$

Deci în ipotezele (A, B) stabilizabilă, (C, A) detectabilă și (25) există un compensator stabilizator care se obține luând o reacție după starea estimată prin intermediul unui estimator de ordin redus pentru sistemul \tilde{T} . Ecuațiile compensatorului sunt

$$\begin{aligned} \dot{w} &= (A + LC)w + (A + LC)L\tilde{y} + Bu \\ \hat{x}_1 &= w + L\tilde{y} \\ \hat{x}_2 &= \tilde{y} \\ u &= F(w + L\tilde{y}) + \tilde{F}\tilde{y} \end{aligned} \quad (37)$$

sau eliminând w prin înlocuire din a doua ecuație și înlocuind u în prima ecuație obținem

$$\begin{aligned} \dot{\hat{x}}_1 &= (A + LC + BF)\hat{x}_1 + B\tilde{F}\hat{x}_2 + L\epsilon \\ \hat{x}_2 &= \epsilon \\ u &= F\hat{x}_1 + \tilde{F}\hat{x}_2 \end{aligned} \quad (38)$$

Deci un compensator stabilizator pentru $T(s)$ este dat de

$$K(s) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right] = \left[\begin{array}{cc|c} A + LC + BF & B\tilde{F} & L \\ O & O & I \\ \hline F & \tilde{F} & O \end{array} \right]. \quad (39)$$

Acest compensator (regulator) asigură de fapt reglarea.

Prin urmare pașii unei proceduri de reglare la referință treaptă cu compensare asimptotic intern stabilă sunt:

Pasul 0. Se verifică dacă

- (A, B) stabilizabilă și (C, A) detectabilă
- $\text{rank} \begin{bmatrix} -A & -B \\ C & O \end{bmatrix} = n + p$

Pasul 1. Se formează sistemul extins

$$\tilde{A} = \begin{bmatrix} A & O \\ -C & O \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B \\ O \end{bmatrix}, \quad \tilde{C} = [O \quad I_p], \quad \tilde{D} = O.$$

și se determină $F_{ext} = [F \quad \tilde{F}]$ astfel încât $\tilde{A} + \tilde{B}F_{ext}$ să fie stabilă.

Pasul 2. Se determină L astfel încât $A + LC$ să fie stabilă.

Pasul 3. Compensatorul $K(s)$ este dat de

$$K(s) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right] = \left[\begin{array}{cc|c} A + LC + BF & B\tilde{F} & L \\ O & O & I \\ \hline F & \tilde{F} & O \end{array} \right].$$

B) Exemple

- (i) Modelul liniarizat al mișcării longitudinale a unui elicopter poate fi descris de următorul sistem de ecuații:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -0.4 & 0 & -0.01 \\ 1 & 0 & 0 \\ -1.4 & 9.8 & -0.02 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ w \end{bmatrix} + \begin{bmatrix} 6.3 \\ 0 \\ 9.8 \end{bmatrix} u$$

heli-eps-converted-to.pdf

Au fost făcute următoarele notații : q este viteza unghiulară de tangaj, θ este unghiul de tangaj, w este viteza de deplasare pe orizontală, iar u (mărimea de comandă adică intrarea în sistem) este unghiul de înclinare al rotorului (*tangajul* este mișcarea în plan longitudinal a unei nave, mișcare efectuată în jurul unei axe transversale).

Cerințe:

- a) Să se determine polii sistemului folosind funcția `eig` și să se stabilească dacă sistemul este stabil sau nu.

- b) Considerând sistemul cu intrarea u și ieșirea w (viteza de zbor pe orizontală) să se stabilească dacă sistemul este *controlabil și observabil*.
Indicație. În acest caz o realizare pe stare a sistemului cu mărimea de ieșire w este dată de:

$$A = \begin{bmatrix} -0.4 & 0 & -0.01 \\ 1 & 0 & 0 \\ -1.4 & 9.8 & -0.02 \end{bmatrix}, \quad B = \begin{bmatrix} 6.3 \\ 0 \\ 9.8 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \quad D = 0$$

Pentru a crea în MATLAB o variabilă "sistem" din realizarea pe stare de mai sus, folosiți apelul `T=ss(A,B,C,D)`.

- c) **Problema stabilizării.** Să se scrie o funcție MATLAB care să aibă ca parametri de intrare matricele A, B, C, D ale unei realizări pe stare a sistemului și doi vectori P_{aloc} și P_{est} de lungime egală cu ordinul matricei A .

Funcția trebuie să întoarcă *compensatorul stabilizator* dat de următoarea realizare pe stare:

$$A_c = A - BF - KC, \quad B_c = K, \quad C_c = F, \quad D_c = 0$$

adică `T_c=ss(A_c,B_c,C_c,D_c)`.

În interiorul funcției se va folosi o procedură de alocare a polilor (de exemplu `place`) pentru a determina matricele F și K astfel încât $\Lambda(A - BF) = P_{aloc}$ și respectiv $\Lambda(A - KC) = P_{est}$.

Pentru exemplul în cauză folosiți următoarele date pentru P_{aloc} și P_{est} :

$$P_{aloc} = [-0.6565 \quad -1-j \quad -1+j] \\ P_{est} = [-8 \quad -4+\sqrt{3}*j \quad -4-\sqrt{3}*j]$$

Indicație. Atragem atenția asupra faptului că funcția `place` primește ca parametri de intrare o pereche (A,B) *controlabilă* și un vector de valori notat mai sus cu P_{aloc} și întoarce o matrice F astfel încât $\Lambda(A - BF) = P_{aloc}$. Pentru a rezolva problema "duală" de alocare pentru perechea *observabilă* (C,A) trebuie apelată funcția `place` cu parametrii A^T, C^T (care formează o pereche *controlabilă*). Apelul `place(A',C')` întoarce pe K^T astfel încât $\Lambda(A - KC) = P_{est}$.

Verificați stabilitatea sistemului H în buclă închisă, dat de `H=feedback(T,T_c)`.

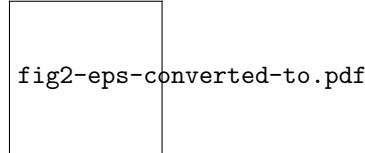
fig1-eps-converted-to.pdf

Comparați polii lui H cu elementele vectorilor P_{aloc} și P_{est} . Ce constatați?

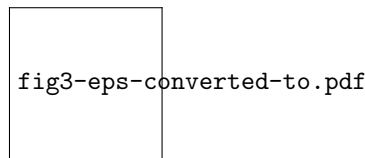
Trasați răspunsul sistemului în buclă închisă la intrare treaptă.

mtlb1-eps-converted-to.pdf

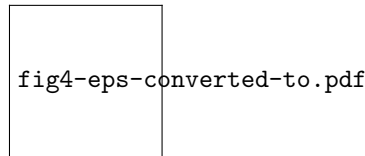
- d) **Problema reglării la referință treaptă unitară.** Pentru rezolvarea problemei reglării, ideea este de a introduce modelul intern al referinței de tip treaptă ($\frac{1}{s}$) în funcția de transfer a căii directe, deci în regulator (dacă ea nu este deja conținută în modelul sistemului).



În acest scop procedăm astfel:



unde am ales \tilde{T}_c astfel încât $T_c = \frac{1}{s} \tilde{T}_c$. Acum aplicăm procedura de stabilizare de la punctul anterior pentru sistemul $\frac{1}{s} T(s)$ (unde $T(s)$ este modelul elicopterului.)

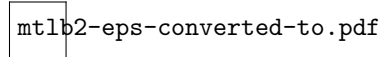


Așadar pentru `integ=ss(0,1,1,0)`, respectiv `T=ss(A,B,C,0)` apelăm funcția de la c) cu parametrii de intrare: `sysext=series(integ,T)` și vectorii \tilde{P}_{aloc} și \tilde{P}_{est} de lungime 4 (adică ordinul matricei A a lui `sysext`). Conform explicațiilor anterioare, acest apel va returna o realizare pe stare a lui \tilde{T}_c . Regulatorul este dat de $T_c = \frac{1}{s} \tilde{T}_c$.

Indicație. Pentru vectorii \tilde{P}_{aloc} și \tilde{P}_{est} folosiți datele numerice:

```
P_aloc = [-0.6565  -1-j   -1+j   -1]
P_est  = [-8      -4+sqrt(3)*j  -4-sqrt(3)*j  -1]
```

Să se vizualizeze răspunsul la treaptă al sistemului în buclă închisă: `feedback(series(T_c,T),1)`. De această dată ieșirea sistemului în buclă închisă urmărește în regim staționar referința treaptă unitară



- e) Să se construiască modelele Simulink ale sistemelor de la punctele c) și d).
- (ii) Modelul liniarizat al mișcării longitudinale a unui avion care zboară cu 0.9 Mach la 8 Km altitudine este dat de:

$$A = \begin{bmatrix} -0.02 & -36.62 & -18.90 & -32.09 & 3.25 & -0.76 \\ 0 & -1.90 & 0.98 & 0 & -0.17 & -0.01 \\ 0.01 & 11.72 & -2.63 & 0 & -31.60 & 22.39 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -30 & 0 \\ 0 & 0 & 0 & 0 & 0 & -30 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 30 & 0 \\ 0 & 30 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Cerințe:

- Să se determine polii sistemului și să se stabilească dacă sistemul este stabil sau nu.
- Construiți un regulator care să regleze la semnale de tip treaptă.

```
% dimensiunile matricelor reprezentarii pe stare
[n,n] = size(A);
[n,m] = size(B);
[p,n] = size(C);

% Pasul 0
% conditia i
R = ctrb(A,B);rank(R)
Q = obsv(A,C);rank(Q)

% conditia ii
rank([-A -B;C zeros(p,m)]);

% Pasul 1
Atilda = [A zeros(n,p);-C zeros(p,p)];
Btilda = [B; zeros(p,m)];
Ctilda = [zeros(p,n) eye(p)];

Fext = place(Atilda, Btilda, [-10 -10 -11+j -11-j -12 -12 -13 -13])
Fext = -Fext;

F = Fext(:,1:n);
Ftilda = Fext(:,n+1:n+p);

% Passul 2
L = place(A',C',[-1 -2 -3 -4 -5 -6]);
L=-L';

% Pasul 3
A1 = A+L*C+B*F;
A2 = B*Ftilda;

[l1,c1] = size(A1);
[l2,c2] = size(A2);
```

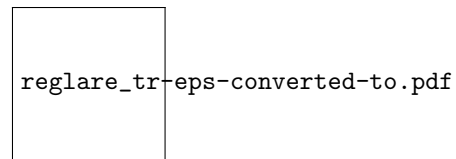
```

Ak = [A1 A2;zeros(c1+c2-11,c1) zeros(c1+c2-11,c2)];
Bk = [L; eye(size(L,2))];
Ck = Fext;
Dk = zeros(size(Ck,1),size(Bk,2));

% Formarea buclei de reactie
K = ss(Ak,Bk,Ck,Dk)
T = ss(A,B,C,D)
H0 = feedback(series(K,T),eye(2))
step(H0,10)

```

Răspunsul la treaptă va fi:



Bibliografie

- [1] **Oară C., Ștefan R.** *Cursul de Teoria sistemelor automate*, Facultatea de Automatica si Calculatoare, Universitatea Politehnica Bucuresti.
- [2] **Jora B., Popeea C., Barbulea S.** *Metode de calcul numeric în automatică*, Editura Enciclopedica, București 1996.
- [3] **Varga A.** *A Schur method for pole assignment*, IEEE Trans. Autom. Contr., 26:517-519, 1981.

0.4 ANEXA

```

function [H]=algoA(A,B,alpha)

n =size(A,1);
Q = eye(n);

[U,S] = schur(A);    % A = U*S*U'; S = U'*A*U
Y = U'*B;           % B = U*Y; Y = U'*B
Q = Q*U;

vp = ordeig(S);
[U,S] = ordschur([],S,real(vp)<-alpha);
Y = U'*Y;
Q = Q*U;

q=0;
for i=1:length(vp)
    if (real(vp(i))<-alpha) q=q+1;
    else continue
    end
end

H=0;
i=q+1;
while (i<=n)
    if (real(S(n,n))==real(S(n-1,n-1)) && imag(abs(S(n,n)))==imag(abs(S(n-1,n-1))))
        p=2;
        F=S(n-1:n,n-1:n)
        G=Y(n-p+1:n,:)
        K = procA(F,G,[-3*alpha -3*alpha])
        eig(F+G*K)
    else
        p=1
        F=S(n,n)
        G=Y(n-p+1:n,:)
        K = procA(F,G,[-3*alpha])
        eig(F+G*K)
    end

    S=S+Y*[zeros(size(K,1),size(S,2)-size(K,2)) K];

    H=H+[zeros(size(K,1),size(U,1)-size(K,2)) K]*Q';

    vp = ordeig(S);
    [U,S] = ordschur([],S,real(vp)<-alpha);
    Y=U'*Y;
    Q = Q*U;

    i = i+p;

```

```

end

function [K] = procA(F,G,gamma_p)

r = rank(G);
p = size(F,1);

[U,G_tilda_1,V] = svd(G);

G_tilda=G_tilda_1(:,1:rank(G_tilda_1));

F_tilda = U'*F*U;

asa= false;

if (r == p)
    if (p==1)
        J = gamma_p(1);
    else
        J(1,1) = gamma_p(1);
        J(2,2) = gamma_p(2);
        J(2,1) = 0;
        J(1,2) = rand(1,1);
    end
    K_tilda = inv(G_tilda)*(J-F_tilda);
    asa = true;
end

if (asa == false)
    tmp1 = (gamma_p(1)+gamma_p(2)-F_tilda(1,1)-F_tilda(2,2))/G_tilda(1,1);
    tmp2 = (F_tilda(2,2)/F_tilda(2,1))*tmp1+(F_tilda(1,1)*F_tilda(2,2)-F_tilda(1,2)*F_tilda(2,1))/F_tilda(2,1);
    K_tilda = [tmp1 tmp2];
end

K = V*[K_tilda;zeros(size(V,2)-size(K_tilda,1),size(K_tilda,2))]*U';

```