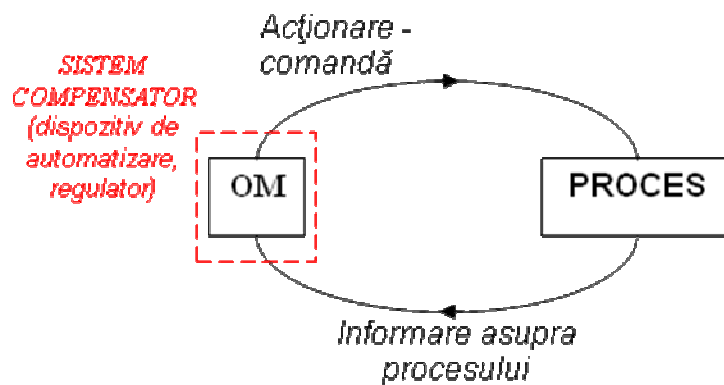


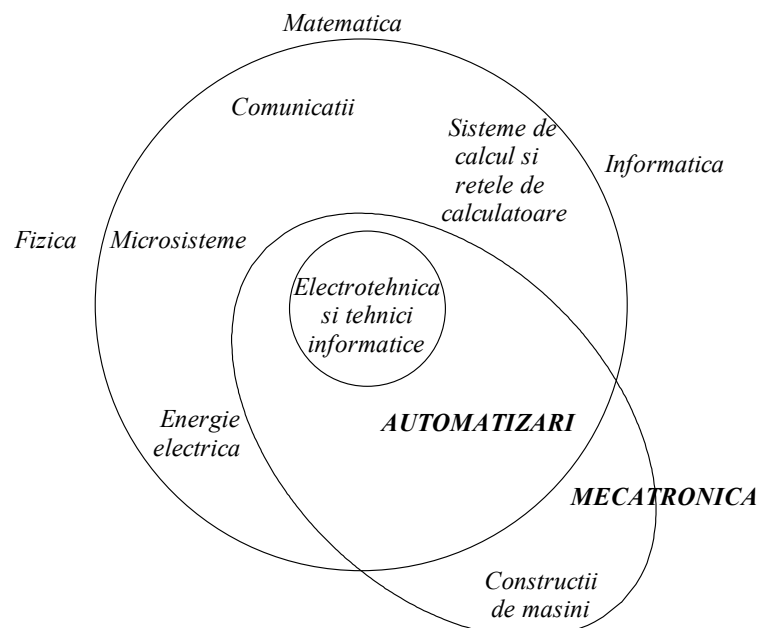
PRELIMINARII PRIVIND EVALUAREA PERFORMATELOR SISTEMELOR - PROBLEMATICA CONDUCERII

a) Teoria sistemelor

- *Sistem* – Proces (tehnic/individual). Subsistemele conditionate ale unui proces si marimile reprezentative. Clasificarea sistemelor/proceselor (tehnice). Interconexiunea om-proces. Comanda automata si neautomata (normala). Sisteme de reglare automata (SRA): definitie, tipuri. Reglare si conducere. Problematika abordarii SRA: analiza si sinteza. Alegere si acordarea reglatoarelor.

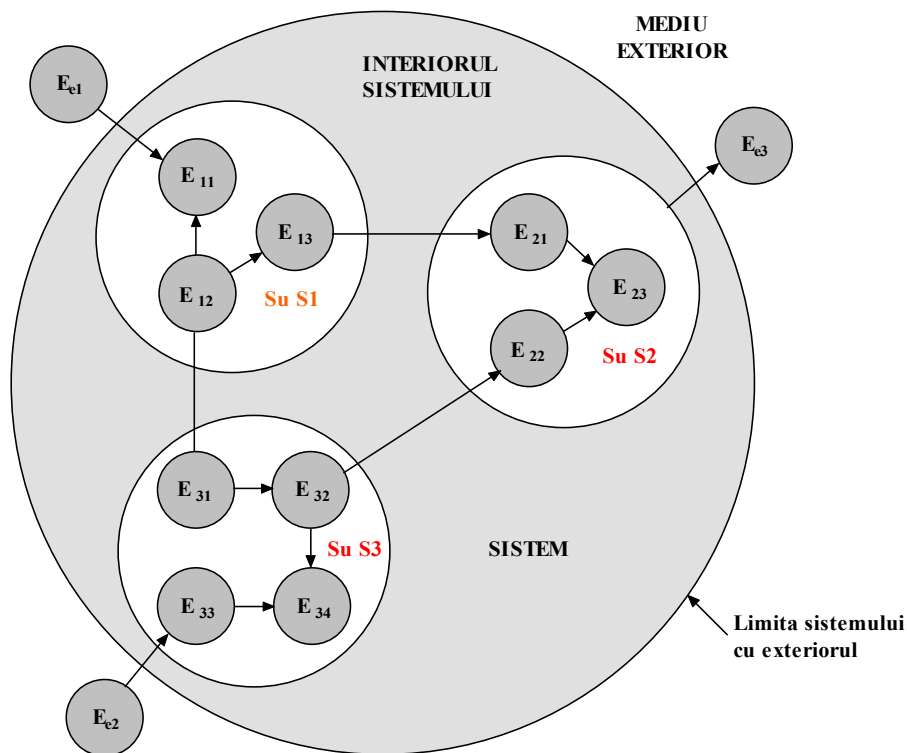


Relația operator uman - proces



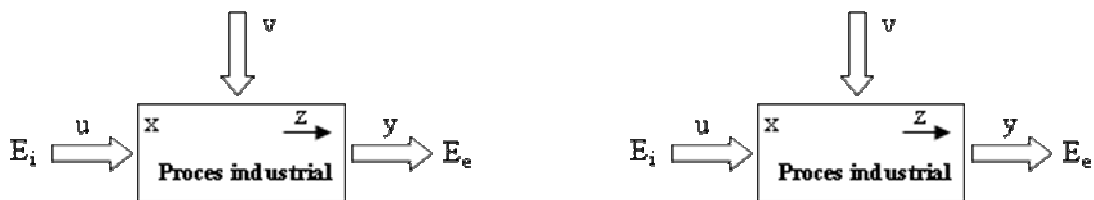
SISTEME SI PROCESE

Într-un limbaj tehnic aplicativ prin noțiunea de **sistem (tehnic)** se înțelege un ansamblu de elemente componente fizico-tehnice, care acționează unele asupra altora într-un mod bine determinat.



Sistem (tehnic): E_{ij} – element constituant al sistemului; $Su S_k$ - subsistemul k

În același sens tehnic, **procesul industrial**, ca ansamblu de fenomene de natură complexă, concepute, de regulă, de către om cu o destinație funcțională precisă, explicitează transformările masice și / sau de energie și de informații.



a)

b)

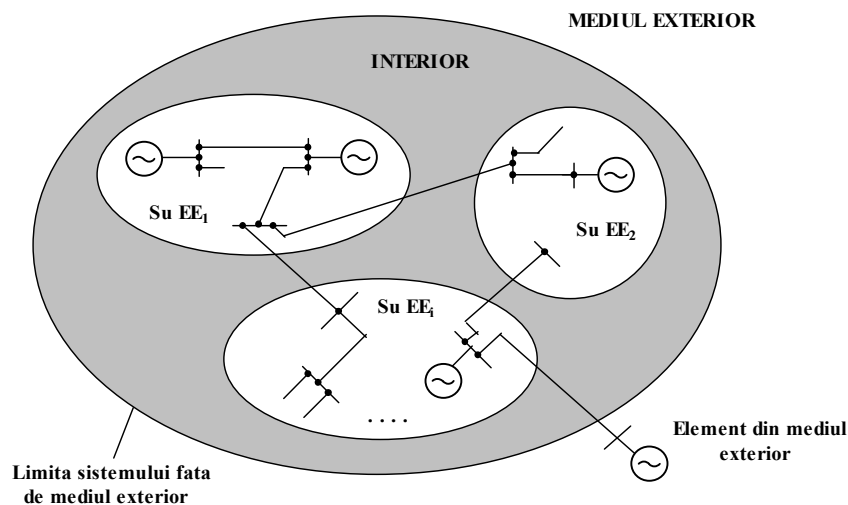
*Reprezentarea unui proces industrial sub formă de schemă bloc
Sistem cu mai multe intrări și mai multe ieșiri (MIMO);
Sistem cu o intrare și o ieșire (SISO)*

■ *Sisteme mari - exemplu sistem electroenergetic*

Un sistem mare poate fi recunoscut dupa un set de caracteristici (Filip 1986):

- structura interconectata;
- existenta mai multor obiective, uneori vagi si/sau conflictuale (Tomovic, 1972);
- restrictii in structura informationala;
- dimensionalitate mare;
- prezenta incertitudinii (Siljak 1983).

Sistemul electroenergetic (SEE) este constituit din elemente generatoare de energie electrică, transformatoare, linii electrice, transport și echipamente de distribuție a energiei electrice. Aceste elemente sunt grupate zonal constituind subsistemele unui SEE (SuEE).



Sistem electroenergetic (SEE): Su EE_i subsistemul electroenergetic "i"

Caracteristici ale unui SEE:

- producerea si consumul de energie electrica se face simultan;
- procesele dintr-un SEE prezinta, in cea mai mare parte, proprietati de autoreglare, dar gradul de statism natural este mare;
- raspandire pe o arie geografica mare;
- ansamblu de procese rapide si lente;
- energia electrica generata trebuie sa indeplineasca o serie de criterii stricte de calitate;
- alimentarea fara intrerupere cu energie electrica, in special, pentru consumatorii industriali.



Sistemul electroenergetic ≡ Sistem mare ≡ Sistem complex

b) Analiza sistemelor informatice (ASI)

ASI ca sistem: definitie, problematici. Terminologie de baza: produs informatic (sistem/aplcatie informatica, produs program), ciclul de realizare si viata a unui produs informatic, calculator de proces, timp real.

ANALIZA DE SISTEM

- Cresterea numarului de aplicatii implicand sisteme numerice de conducere
- Largirea sistemului aplicatiilor informatice

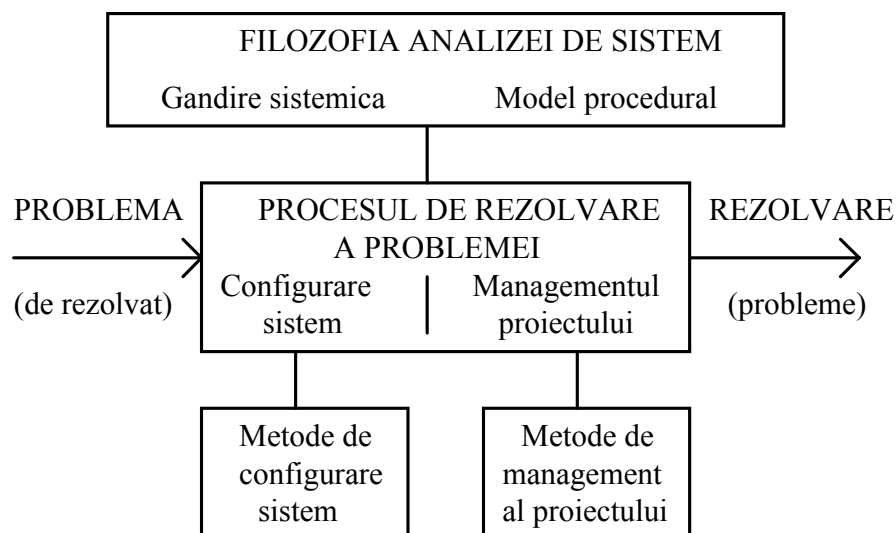


Necesitatea realizarii unui cadru metodologic cat mai bine conturat pentru analiza, doar in special pentru proiectarea sistemelor ce implica utilizarea tehnologiei numerice.



ANALIZA DE SISTEM

- **Ce** trebuie facut ?
- **De ce** trebuie sa se faca ?
- **Ce ar trebui** sa se faca ?
- **Cand** trebuie facut ?
- **Cine** trebuie sa faca ?
- **Cine** raspunde ?
- **Unde** se face ?



Componentele analizei de sistem [Daenzer]

Interactiunea MIS (Management Information System) cu conducerea proceselor (MES - Manufacturing Execution System)

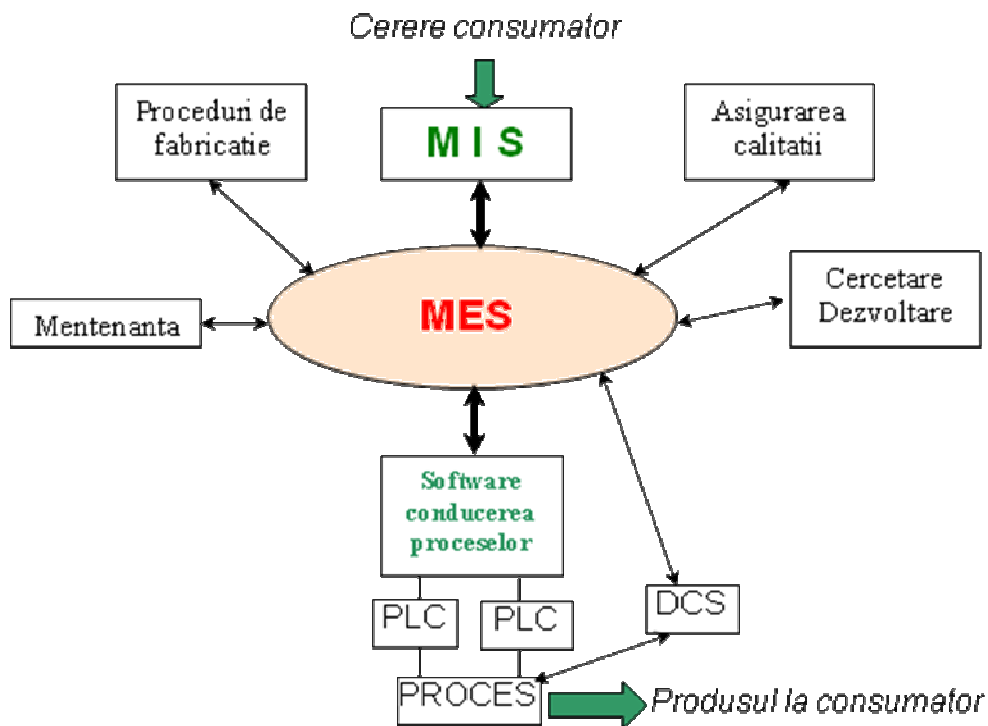
Obiective MES

Legatura intre cele doua sisteme de gestiune si de proces realizand o structurare progresiva a informatiei.

Rolul principal → relationarea intre cele doua sisteme existente si coordonarea unei multitudini de sarcini noi necesare punerii in functiune a unei structuri integrate inteligente.

Funcțiile MES

- 1) colectarea si distribuirea datelor din proces;
- 2) supervizarea productiei(executia procedurilor de fabricatie);
- 3) monitorizarea procesului.



c) Teoria multiagent

- **Necesitate si avantaje**

In cadrul lucrarii vom intelege prin **agent tehnic** un sistem cu proprietati bine determinate.

In acest context, un sistem complex care este constituit din mai multe subsisteme cu functionalitati bine determinate, poate fi considerat un ansamblu format din mai multi agenti tehnici ce interactioneaza activ intre ei, numind acest ansamblu, **sistem de mai multi agenti tehnici (SMAT)**.

Agens_(lat.), termenul corect are urmatoarele semnificatii: *mod sau principiu de a proceda si a actiona* ⇒ agent (eng.).

Dezvoltarile tehnice si tehnologice din domeniul conducerii proceselor au ridicat o serie de probleme:

- Cum acoperim din punct de vedere tehnico-ingineresc necesitatea de realizare de sisteme automate autoorganizate?
- Cum integram actualele sisteme de conducere pentru a se realiza sisteme/echipamente/masini inteligente ale viitorului?
- Care este rolul omului in cadrul unor astfel de sisteme integrate? Care om sa fi descongestionat de sarcinile de conducere de mare amploare? Si sa se asigure o singuranta sporita!

Raspunsul este dat de viata reala care rezolva aceste tipuri de intrebari de asociere prin realizarea (evolutia) a unor comunitati autoorganizate.

Cu alte cuvinte, se doreste sa se ajunga la sisteme tehnice autonome care sa fie organizate in asociatii/comunitati complet independente. Desi acest tel este departe de a fi realizat, exista deja necesitati de sisteme autonome care sa fie cuplate sau decuplate in mod modular, flexibil si dinamic.

Dorim sa punem in evidenta, pentru un sistem electroenergetic, abordarea activitatilor si comenzii inteligente sistemelor autonome modulare ce pot fi preluate in sistemele complexe globale, atat individual, cat si cu sarcini comune.

Pentru activarea impreuna intr-un sistem global/total un sistem autonom trebuie sa fie echipat cu capacitati, care sa inlesneasca / sa faca posibil un compromis intre atingerea optima a scopurilor individuale, impiedicarea si stanjenirea minimala a altor sisteme.

Revenind asupra notiunii de agent tehnic vom preciza ca acest termen va fi utilizat pentru a desemna acea entitate din interiorul unui SMAT care are urmatoarele proprietati:

- *Optimizarea proceselor*: un agent tehnic este un sistem ce cauta optimizarea functionarii unuia sau mai multor procese;
- *Comportare autonoma*; unde prin autonomia unui agent tehnic intelegem ansamblul simultan a doua proprietati: agentul sau o parte a lui este cvasicontinuu activ pentru a genera stabilitatea optimizarii (*activitate autonoma*) si in afara de acesta, el dispune de o strategie alternativa pentru atigerea unui scop/tel optimal, el putand decide independent acest lucru (*competenta si raspunderea decizionala autonoma*)

- *Controlul de sine a interactiunilor*; un agent tehnic foloseste intr-un SMAT cauzele schimbarii posibile intre agenti pentru atingerea scopurilor/telurilor optimizarii sale. Dupa imprejurari aceste interactiuni fizice sau tehnico-informationale, dorite sau nedorite pot sa fie agent.

Prin aceasta definitie putem desemna componentele/oarecari ai unui sistem complex ca agenti, daca se dau si se cunosc proprietatile in acel sistem.

Putem astfel desemna oameni, masini, procese soft (task-uri) si procesoare.

Agent este sinonim cu cea de unitati integrate autonome in sisteme complexe

Filozofia aplicarii SMAT consta in realizarea de arhitecturi si comenzi inteligente din sisteme autonome modulare care sa poata prelua sarcini individuale sau in comun pentru sisteme complexe

- **Sisteme inteligente**

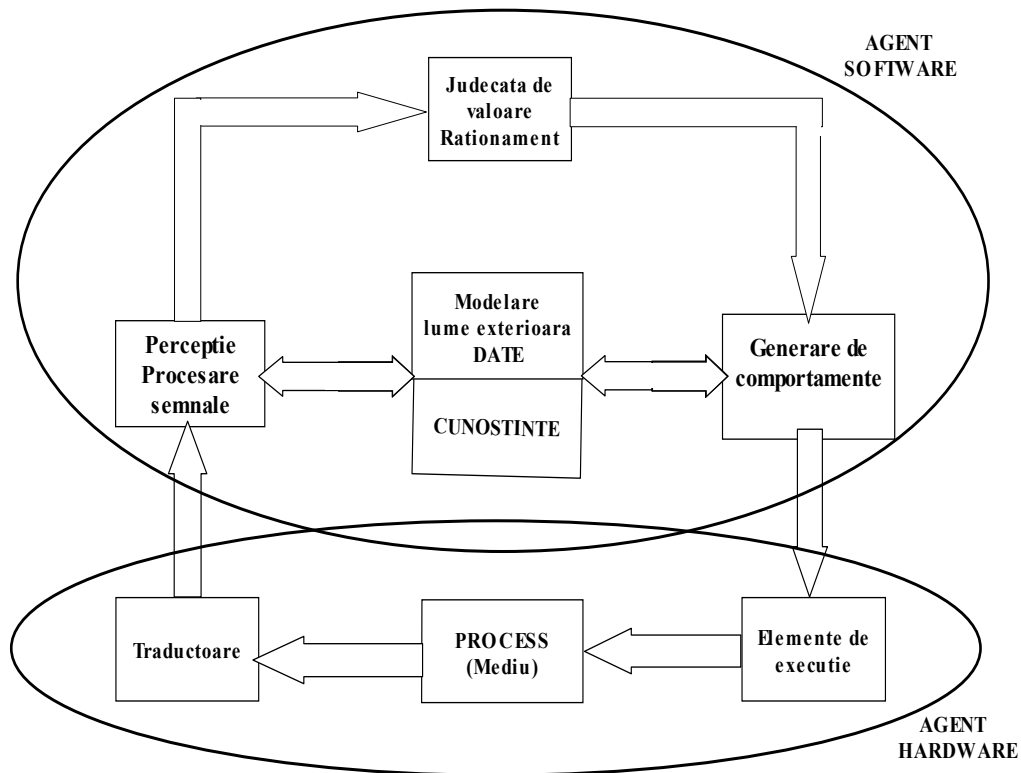
Imprecizia in modelarea matematica riguroasa a elementelor si subsistemelor constituate unui proces (tehnic) complex au impus in ultimii 20 de ani o serie de concepte neconventionale in conducerea proceselor. S-a apelat astfel la concepte ale inteligentei artificiale, realizandu-se sisteme bazate pe cunostinte, sisteme expert de conducere in timp real etc.

Un sistem inteligent de conducere este dezvoltat si implementat cu o metodologie inteligenta, cu o cumulare a unor tehnici ce reproduc functii ale unor sisteme biologice. Temenul de *inteligent* este usor abuziv daca ne gandim ca in spatele acestei sintagme sta un automat care isi propune modelarea gandirii si a modului de actiune umana. In acest scop, acest automat are la baza un model si unul sau mai multi algoritmi care reactioneaza in urma unor strategii prestabilite. In masura in care acest automat este dotat cu tehnici si tehnologii de invatare care si-l face creativ si cat mai autonom, se poate spune ca exista o anumita acoperire a termenului de inteligenta.

Pentru o lunga perioada s-a considerat si s-a statuat ca regula atigerea unor forme de operare, prin care subsistemele din interirul sistemelor complexe sa interactioneze cu success. In domeniul ingineriei acest lucru este foarte important deoarece este necesar sa existe un control stabil pentru indeplinirea unor decizii (comenzi) luate.

Dezvoltarile tehnologice au impus odata cu dezvoltarea teoriei sistemelor multi-agent si aparitia necesitatii **functionarii autonome** ceea ce presupune ca orice element constituent al unui sistem poate sa ia decizii independente dintr-un mediu necunoscut si dinamic.

Un astfel de sistem, pentru a putea functiona trebuie structurat ca un **sistem cu inteligenta (artificiala)** reprezentat in schema bloc de mai jos.



Schema unui sistem cu inteligenta artificiala

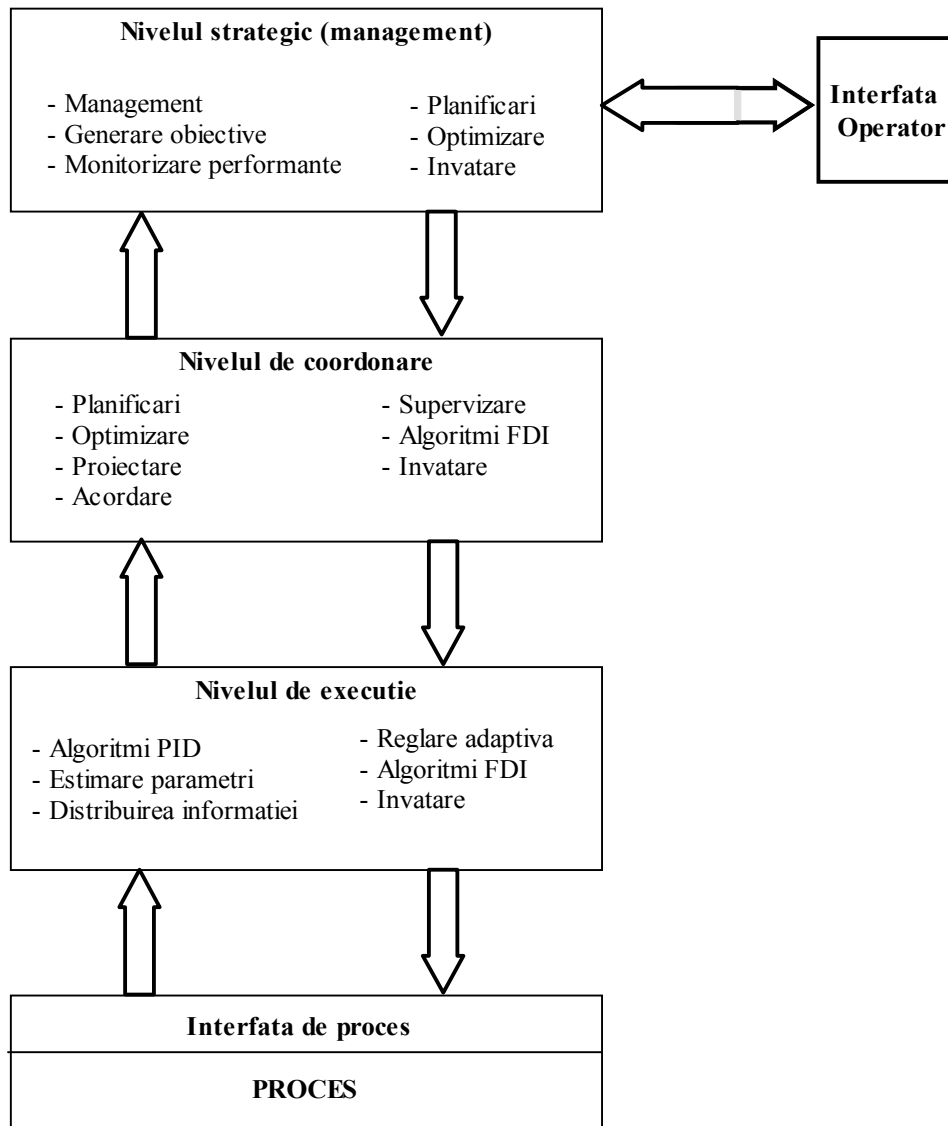
Acest model pune in evidenta cele patru functii esentiale ale unui astfel de sistem: perceptie, invatare, rationament si generarea de comportament.

- **Sisteme autonome**

Sistemele autonome necesita un inalt nivel de inteligenta pentru a se asigura o independenta, in sensul autonomiei de inalt nivel.

Nota: *Proiectarea unui sistem de conducere autonom constituie un obiectiv, iar mijlocul prin care se realizeaza acest lucru este inteligenta (artificiala)*

Arhitectura unui sistem inteligent autonom este prezentata mai jos:



Sistem inteligent autonom

Se observa organizarea unui astfel de sistem pe trei niveluri/straturi ierarhice:

- **Nivelul (stratul) executive** comunica cu procesul prin interfata de proces, el avnd rolul de a realiza atat legi de comanda standard (clasice) de tip PID dar si algoritmi complexi (adaptive, estimare, aptimala, etc.);

- **Nivelul (stratul) de coordonare** este un nivel de actiune tactic ce realizeaza o integrare a tuturor functiilor si algoritmilor de la nivelul executiv cu cel de planificare, invatare, supervizare si coordonare, respectiv de identificare a defectelor si de reproiectare on-line a strategiei de conducere.

- **Nivelul (stratul) de management** este un nivel strategic si are rolul de a superviza nivelele inferioare (coordonare si executiv) si de a monitoriza performantele intregului sistem.

De mentionat este faptul ca pe fiecare nivel (strat) sunt introduse functii ce confera acestora intr-o distributie ierarhica un anumit grad de inteligenta. In dialectica ascenderata gradul de inteligenta creste concomitent cu scaderea preciziei asupra fenomenelor si performantelor procesului condus.

Funcțiile specifice sistemelor inteligente (SI) destinate conducerii proceselor (tehnice) sunt realizate prin tehnologii și sisteme avansate: fuzzy, rețele normale, genetice, hibride (neuro-fuzzy, geno-fuzzy, geno-neuro-fuzzy). La care adăugăm, evident și tehnici standard (clasice, convenționale).

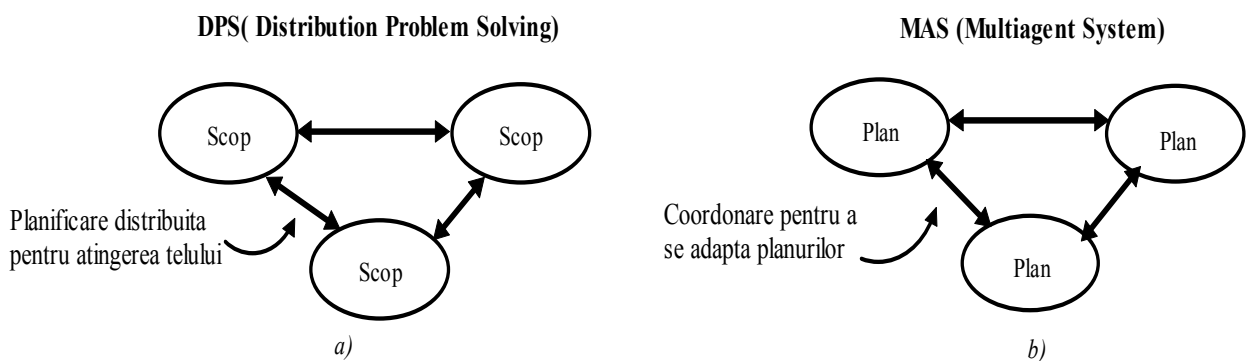
- **Inteligența artificială distribuită**

Un mod de utilizare a noțiunii de agent este și cea legată de **inteligența artificială distribuită**. Aceasta noțiune presupune o analiză de entități care să se ocupe de activități ce presupun cooperare în sisteme de planificare distribuită sau destinate rezolvării de probleme.

Există două clase de utilizări în cadrul acestui concept:

- Mai multe sisteme lucrează împreună pentru a rezolva o problemă globală, fiecare sistem neputând rezolva singur acea problemă- **rezolvarea distribuită a problemei (DPS)**;

- Mai multe sisteme rezolvă problema locală propusă, unde rezolvarea problemei unui sistem poate fi influențată negativ sau pozitiv de rezolvarea problemei altui sistem – **sistem multiagent (MAS)**.



Inteligența artificială distribuită

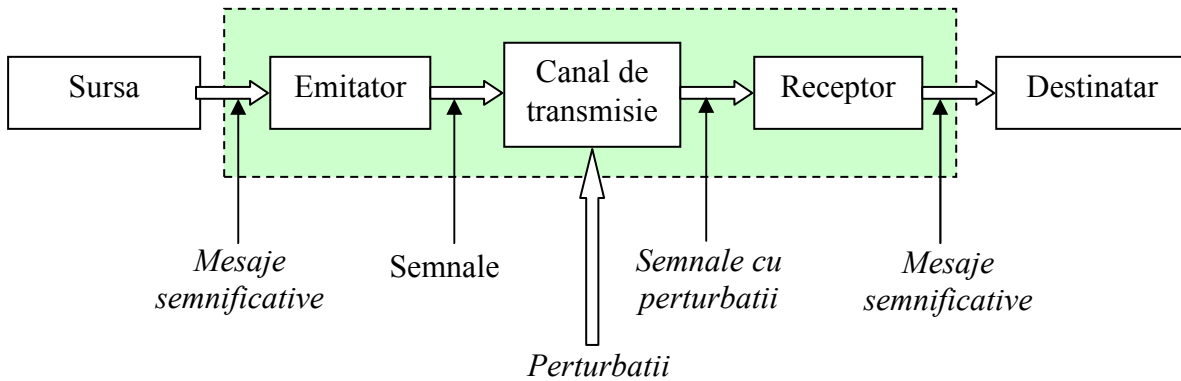
Este evident că în practică nu putem clasifica sistemele de planificare distribuite în mod exclusiv în una din aceste două clase. În mod natural SMAT în planificările destinate rezolvării problemelor locale funcționează și lucrează împreună pentru a deține un scop global.

Metodele inteligenței artificiale distribuite se bazează pe o modelare a lumii deterministe, în care se pot executa de către agenți planuri elaborate în tehnici simbolice.

□ Elemente de matematica aplicata in conducerea proceselor

□ Bazele transmisiei informatiei - exemplu de interpretare

TRANSMISIA INFORMATIEI. CODIFICAREA.



Schema bloc generala a unui sistem de transmisie a informatiei

Alfabet A – cuvânt (text) – limba în A

$$f : E \rightarrow L(A)$$

$$\rightarrow \{E, L(A), f\} \stackrel{\Delta}{=} \text{codificare}$$

multimea
care se
modifica

cod

regula de
codificare

rang (pozitie)

$$\underbrace{(a_0, a_1, a_2, \dots, a_{n-1})}_{n \text{ lungimea(cuvantului)cod}}$$

Cuvant cod

Coduri definite pe CG (p)

- binare $p = 2$, CG (2) sau $\{0, 1\}$
- ternar $p = 3$, CG (3) sau $\{-1, 0, +1\}$
- p – ar $CG(p)$ sau $\{0, 1, \dots, p-1\}$

Moduri de reprezentare pentru cuvintele cod:

- **Polinomial.** Un cuvânt de cod $(a_0, a_1, a_2, \dots, a_{n-1})$ se exprima sub forma polinomială ca un polinom de gradul $n-1$:

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1},$$

unde $a_i \in CG(p)$ cu $0 \leq i \leq n-1$, sau pentru codul binar:

$$a_i \in CG(2) = \{0, 1\}$$

Exemplu, polinomul de grad 4:

$f(x) = 1 + 0 \cdot x + 1 \cdot x^2 + 1 \cdot x^3 + 1 \cdot x^4 = 1 + x^2 + x^3 + x^4$ reprezintă combinația de cod (10111) de lungime $n = 5$.

În această reprezentare, cuvintele cod de lungime n pot fi privite ca elemente ale algebrei A_n de polinoame mod $(x^n - 1)$, operațiile cu polinoame făcându-se după regulile câmpului din care sunt luați coeficienții.

- **Matriceal.** Multimea combinațiilor nenule diferite, de lungime n , ale unui cod uniform în număr $p^n - 1$ se pot scrie sub forma unei matrici cu $p^n - 1$ linii și n coloane.

Exemplu, pentru $p = 2$ (cod binar), $n = 3$ obținem $2^3 - 1 = 7$ combinații nenule de cuvinte de cod:

$$T = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

Prin transformări liniare succesive asupra liniilor se obține:

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Se observă că $\det I \neq 0$ și deci rangul matricii coincide cu ordinul ei. Toate cele 8 cuvinte cod se obțin prin combinații liniare ale celor 3 linii ale lui I .

- **Vectorial.** Cuvintele cod de lungime n formeaza un spatiu vectorial V_n .

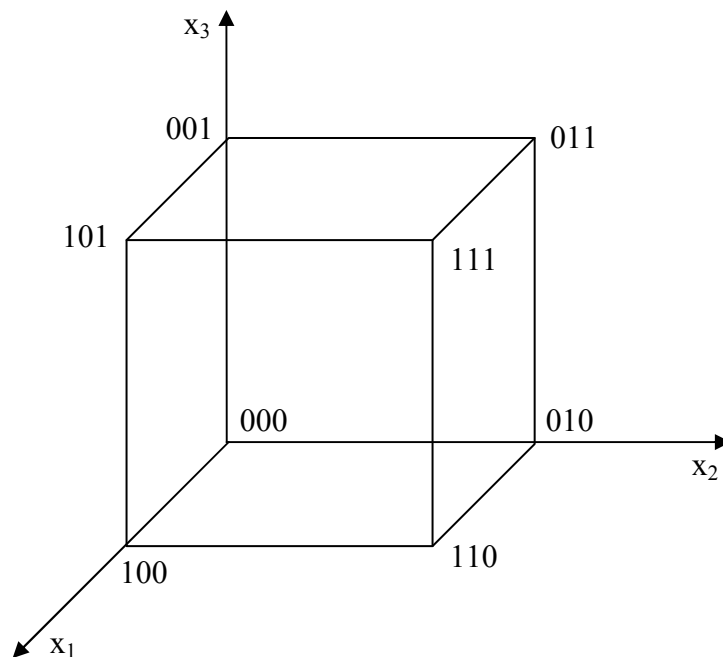
Ca urmare, un cuvânt este un vector cu n componente:

$$v_i = (a_0, a_1, \dots, a_{n-1}), \text{ cu } a_i \in CG(p).$$

Din toate cuvintele posibile se pot selecta acele combinatii care au o anumita proprietate comuna, ce formeaza astfel un *subspatiu vectorial*.

- **Geometric.** In aceasta reprezentare cuvintele codului de lungime n se identifica cu un punct al spatiului n dimensional, formand o submultime a multimii varfului cubului unitar din acest spatiu. Acest lucru permite o serie de posibilitati de realizare a codurilor, utilizand proprietatile figurilor geometrice.

Exemplu, pentru cuvinte cod cu $n = 3$ se obtine reprezentarea din figura de mai jos.



Reprezentarea geometrica a cuvintelor de cod cu $n = 3$

Observatie: Mentionam si alte doua posibilitati de reprezentare a cuvintelor cod si anume ca elemente ale multimilor finite si ca elemente ale geometriilor proiective.

Capitolul 1

Noțiuni introductive

1.1 Concepte de bază în diagnoza anomaliilor proceselor

În cadrul controlului automat al proceselor tehnice, funcțiile de supervizare servesc pentru a indica stările nedorite sau nepermise ale procesului, precum și pentru a lua măsurile corespunzătoare în scopul menținerii funcționării și evitării defectelor sau accidentelor. Pot fi evidențiate următoarele funcții de supervizare [Isermann, 1994; Isermann, 1997]:

- *monitorizarea*: variabilele măsurabile sunt verificate în raport cu toleranțe impuse și sunt generate semnale de alarmă către operator în cazul în care aceste valori sunt depășite;
- *protecția automată*: în cazul unei stări periculoase a procesului funcția de monitorizare inițiază, în mod automat, contra-acțiuni;
- *supervizare cu diagnoza anomaliilor*: se extrag trăsături ale comportării curente a procesului pe baza variabilelor măsurate, se generează simptome care să evidențieze schimbări ale trăsăturilor, se realizează diagnoza anomaliilor și se iau decizii asupra contra-acțiunilor necesare.

Metodele clasice de supervizare (monitorizarea și protecția automată) se recomandă pentru supravegherea de ansamblu a proceselor. În ceea ce privește stabilirea toleranțelor pentru variabilele măsurabile, trebuie realizat un compromis între mărimile deviațiilor anormale necesare detecției anomaliei și fluctuațiile normale ale variabilelor. Simpla verificare a unor valori de prag este recomandată în cazul proceselor ce lucrează în regim staționar. Situația se complică atunci când starea procesului se schimbă rapid în jurul punctului de operare.

În cazul sistemelor automate, cu circuit închis, schimbările în proces sunt estompate de către acțiunile de control și nu pot fi detectate pe baza semnalelor de ieșire, atâta vreme cât semnalele de intrare comandate se mențin în domeniul normal de variație. Astfel, sistemele automate nu permit o detecție anticipată a anomaliilor proceselor.

Avantajele esențiale ale metodelor de supervizare bazate pe verificarea valorii de prag sunt simplitatea și fiabilitatea. Dezavantajul principal al acestor metode este că nu sunt capabile să reacționeze decât după o schimbare esențială a mărimii (trăsăturii) implicate, adică o deviere bruscă, de valoare mare, sau o deviere care crește lent în timp și semnificativ. Un alt dezavantaj al metodelor clasice este acela că, de obicei, acestea nu permit o analiză de detaliu a comportării anormale detectate.

Ținând cont de acestea, apare necesitatea folosirii unor *metode avansate de supervizare și diagnoză a anomaliilor proceselor*. Aceste metode trebuie să satisfacă următoarele cerințe [Isermann, 1994; Isermann, 1997]:

- detecție anticipată a anomaliilor mici cu evoluție lentă sau bruscă în timp;
- diagnoza anomaliilor în elementele de execuție, componentele sau senzorii procesului;
- detecția anomaliilor în sistemele cu circuit închis;
- supervizarea proceselor aflate în regim tranzitoriu.

Termenul de anomalie [Frank, 1987; Marcu, 1995] este folosit ca sinonim pentru căderi, erori sau perturbații în proces care conduc la comportări nedorite și intolerabile ale acestuia. Scopul diagnozei anomaliilor este acela de a detecta apariția unei comportări anormale a sistemului suficient de devreme astfel încât căderea (defectarea) întregului sistem să fie evitată. Scopul esențial al detecției anticipate și al diagnosticării comportărilor nedorite ale sistemului este acela de a avea suficient timp pentru realizarea unor contra-acțiuni, cum ar fi: schimbarea modului de operare, reconfigurarea structurii de reglare, întreținere sau reparație.

Datorită complexității și gradului de risc crescute ale sistemelor de control moderne, pe de o parte, și a cerințelor ridicate pentru calitate, eficiență, disponibilitate, fiabilitate și siguranță, pe de altă parte, devine tot mai importantă cerința ca sistemele automate să fie *tolerante la anomalii* [Frank, 1994a]. Aceasta se poate obține prin strategii pasive și active. Abordarea pasivă face apel la strategii robuste de proiectare cu scopul ca procesul să devină insensibil în raport cu anomaliile. În contrast, abordarea activă promovează o acomodare a anomaliei în sensul reconfigurării sistemului (a controlului) atunci când o anomalie a apărut [Frank, 1994a].

Ca urmare, în ultimii ani s-a dezvoltat tot mai mult un domeniu distinct al automatizării ca subdomeniu al teoriei controlului și anume *diagnoza anomaliilor sistemelor dinamice* caracterizat prin tehnici avansate pentru detecția, localizarea și remedierea anomaliilor și defectelor. Acest nou domeniu este un domeniu interdisciplinar, aflat la intersecția științei sistemelor și științei calculatoarelor, bazat pe cunoștințe profunde de inginerie. Progresele realizate în teoria modernă a controlului și în tehnologia calculatoarelor au permis dezvoltarea unor sisteme inteligente capabile să realizeze sarcinile sofisticate ale supervizării proceselor [Marcu, 1995].

Astfel, dacă sarcinile de bază ale automatizării proceselor legate de reglare constituie un prim nivel de automatizare, diferitele sarcini ale funcțiilor de supraveghere (supervizare) formează un al doilea nivel de automatizare. Cum se poate lucra cu același model al procesului la ambele nivele, rezultă că automatizarea poate fi realizată de un *controler inteligent (autonom)* cu structura din figura I. 1 .

Activitatea controlerului autonom are loc astfel [Marcu, 1995]:

- monitorizează comportarea sistemului în vederea detectării anomaliilor atunci când ele apar și estimează configurația procesului corespunzătoare acestei stări;
- activează un controler cu structură fixată pentru a stabili procesul conform

noii sale configurații;

- activează un controler adaptiv capabil să îmbunătățească răspunsul sistemului.

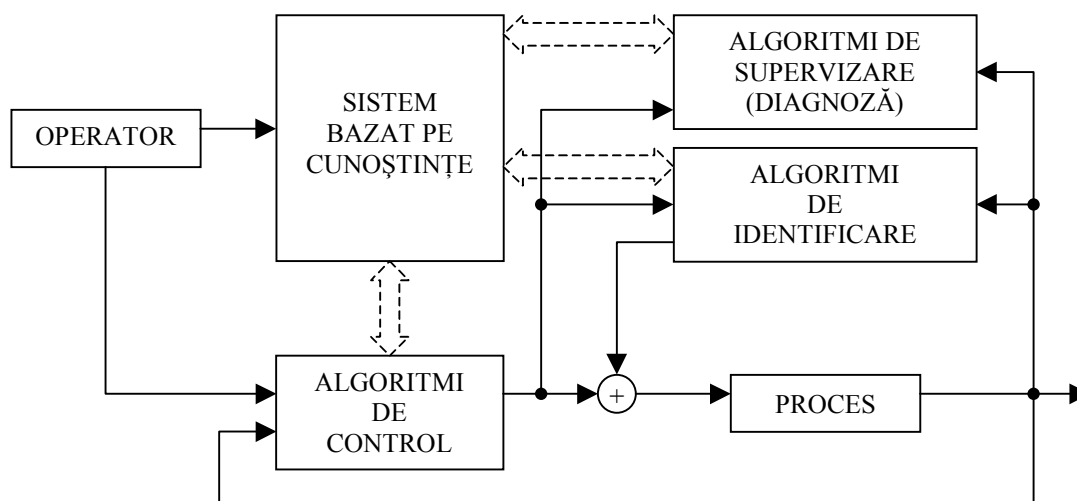


Fig. I. 1 Arhitectura unui sistem ce permite implementarea unui controler autonom

În continuare se consideră că o instalație automatizată este constituită din trei tipuri de subsisteme, așa cum este prezentat în figura I. 2 [Frank, 1996]:

- elemente de execuție;
- procesul automatizat (componentele procesului);
- senzori (instrumente de măsură).

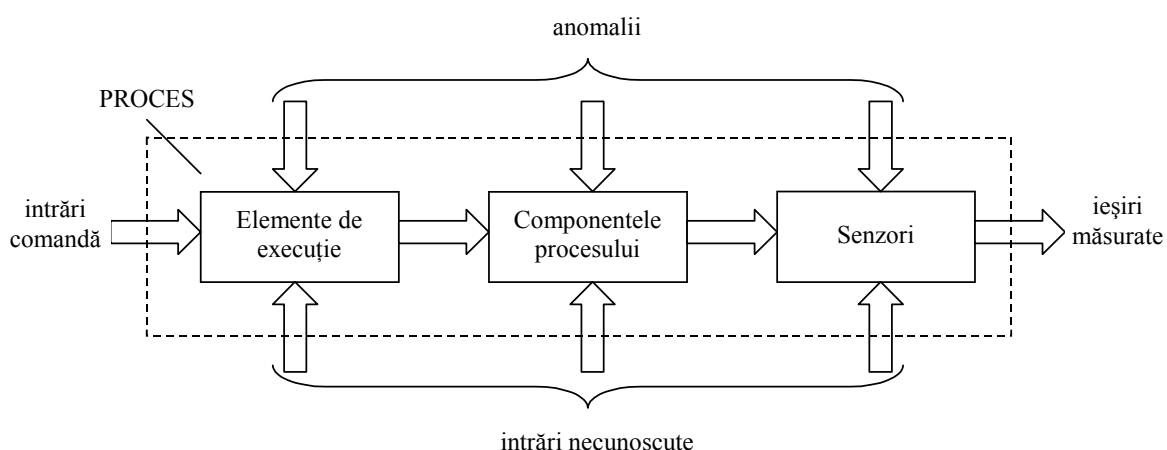


Fig. I. 2 Acțiunea anomaliilor asupra unui sistem automat

Sistemul automat este supus următoarelor acțiuni externe, suplimentar față de intrările de comandă din proces:

- anomalii în:
 - elemente de execuție;
 - subsistemele procesului (componente);
 - senzori;
- erori de modelare;
- zgomotul sistemului de măsurare;
- perturbațiile externe.

Erorile de modelare, zgomotul de măsurare și perturbațiile externe reprezintă *intrările necunoscute* ce acționează asupra sistemului.

În funcție de natura anomaliilor, acestea pot fi [Marcu, 1995]:

- *anomalii aditive de măsurare*: pun în evidență faptul că există o discrepanță între valorile măsurate și valorile adevărate ale mărimilor de ieșire sau intrare ale procesului; acestea reflectă anomaliile în senzori;
- *anomalii aditive de comandă*: pun în evidență proasta funcționare a elementelor de execuție;
- *anomalii aditive ale procesului*: reprezintă perturbații ce acționează asupra procesului și care în mod normal sunt nule; acestea produc deplasări ale valorilor ieșirilor independente de intrările măsurate;
- *anomalii multiplicative ale procesului*: evidențiază schimbări ale parametrilor instalației.

Trebuie făcută distincția între anomaliile aditive și zgomot. În general zgomotul este considerat aleator, de exemplu de tip gaussian și de medie nulă. Anomaliile aditive sunt considerate fie deterministe (deviație liniar variabilă în timp sau constantă în timp), fie semideterministe (sub forma unor salturi ce apar la intervale aleatoare de timp și de amplitudini aleatoare).

Pentru a obține un sistem de diagnoză performant, trebuie să se urmărească, în proiectarea acestuia, evitarea interpretării efectelor intrărilor necunoscute drept anomalii, lucru ce ar putea conduce la sesizări (alarme) false. De o deosebită importanță este diagnoza anomaliilor incipiente care ar putea permite evitarea

condițiilor de operare periculoase, întreținerea funcționării la parametri doriți, diagnoza pe termen lung, creșterea disponibilității, productivității și asigurarea automată a calității [Frank, 1996].

1.2 Clasificarea metodelor de diagnoză a anomaliilor

Conceptul de diagnoză a anomaliilor presupune următoarele trei obiective [Marcu, 1995; Frank, 1996]:

- *deteția anomaliei*: indicarea faptului că apar nereguli (anomalii) în funcționarea sistemului care să conducă la comportări nedorite și intolerabile ale acestuia, în prezența intrărilor necunoscute;
- *localizarea anomaliei*: determinarea subsistemului (sursei) care a generat anomalia;
- *analiza anomaliei*: determinarea tipului, mărimii și cauzei ce a generat o comportare anormală a procesului.

Metodele de diagnoză a anomaliilor proceselor pot fi împărțite în următoarele categorii [Marcu, 1995]:

1. metode ce nu folosesc un model al procesului

Sunt folosite următoarele abordări:

- *verificarea limitelor*: măsurătorile efectuate asupra instalației sunt comparate cu limite cunoscute apriori, depășirea acestora indicând o situație de anomalie;
- *instalarea unor senzori speciali*: pot fi senzori limitatori (ce verifică anumite limite) sau care pot măsura anumite variabile speciale (ex.: sunet, vibrație);
- *instalarea unor senzori multipli* (redundanță fizică): se compară măsurătorile furnizate de senzori diferiți asupra aceleiași variabile, o discrepanță serioasă indicând prezența unei anomalii la cel puțin un senzor;
- *analiza frecvențială a măsurătorilor asupra instalației*: orice deviație a

spectrului tipic de frecvențe în condiții normale de lucru al anumitor măsurători evidențiază prezența unei anomalii;

- *folosirea sistemelor expert tradiționale*: sistemul constă din combinarea unor reguli de tipul:

DACĂ *simptom 1* ȘI *simptom 2* ATUNCI *concluzie*

până se obține concluzia finală, o anomalie specifică.

2. metode ce folosesc un model al procesului

Aceste metode se bazează pe un model al procesului, exploatându-se conceptul de *redundanță analitică*. În acest caz, măsurătorile de la senzori sunt comparate cu valorile obținute analitic, pe baza modelului matematic și a măsurătorilor actuale, diferențele rezultate numindu-se *reziduuri*. În caz ideal, în absența intrărilor necunoscute și a anomaliilor, reziduurile sunt nule. În practică, însă, reziduurile sunt nenule, ele fiind rezultatul combinării intrărilor necunoscute și ale anomaliilor. Dacă intrările necunoscute sunt neglijabile, atunci reziduurile pot fi analizate direct, putându-se trage concluzii asupra prezenței anomaliilor. Dacă, însă, intrările necunoscute se situează la un anumit nivel, se apelează la o analiză statistică sau, mai general, la *recunoașterea de forme*. În concluzie reziduurile trebuie să fie sensitive la anomalii și insensitive la intrările necunoscute care pot conduce la alarme false.

Pentru realizarea diagnozei anomaliilor pe baza modelului asociat procesului, trebuie parcurse următoarele etape, așa cum este ilustrat în figura I. 3 [Frank, 1992; Frank, 1994c; Isermann and Ballé, 1997]:

- *generarea reziduurilor (simptomelor)*: se realizează generarea unor semnale sau simptome care reflectă anomaliile;
- *evaluarea reziduurilor (clasificarea anomaliei)*: se apelează la o logică de luare a deciziei în ceea ce privește momentul apariției anomaliei și a localizării acelei anomalii;
- *analiza anomaliei*: se determină tipul anomaliei, dimensiunea acesteia și cauza ce a provocat-o.

Primele două etape sunt realizate de către *subsistemul de detecție și localizare a anomaliei*. Acesta trebuie să fie sensibil în raport cu anomaliile, pentru a putea

detecta și localiza anomalii incipiente, dar, totodată, trebuie să fie robust în raport cu intrările necunoscute în sistem cu scopul de a evita (sau reduce) alarmele false.

Calitatea detecției anomaliei poate fi pusă în evidență de către raportul dintre sensibilitatea la anomalii și frecvența alarmelor false. Calitatea izolării a cât mai multor anomalii depinde de informațiile disponibile despre proces (numărul variabilelor măsurabile): cu cât numărul acestora este mai mare, cu atât se pot distinge mai multe tipuri de anomalii. Obținerea măsurătorilor fiind, în general, un proces care presupune cheltuieli ridicate, se pune problema proiectării unui sistem de diagnoză care să poată diagnostica, cu robustețe maximă în raport cu intrările necunoscute, cât mai multe și mai incipiente anomalii cu putință, folosind cât mai puține mărimi măsurabile.

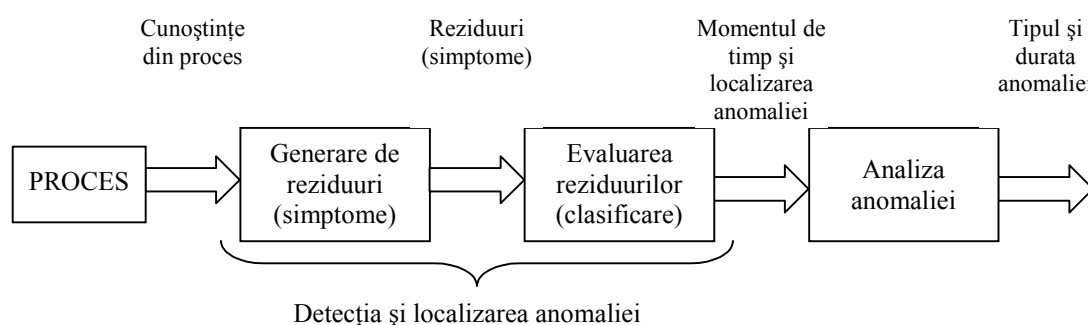


Fig. 1. 3 Schema principală a diagnozei anomalilor bazată pe model

În raport cu *modul de generare al simptomelor*, metodele de diagnoză a anomaliiilor pot fi clasificate în trei categorii [Frank, 1994b]:

- metode bazate pe analiza de semnal;
- metode bazate pe modele analitice;
- metode bazate pe cunoștințe și rețele neuronale artificiale.

Des folosite în practică sunt metodele bazate pe *analiza de semnal*. În acest caz, se extrag din proces semnalele sau simptomele care conțin cât mai multe informații referitoare la anomalie. Pe baza lor, direct sau după prelucrări adecvate, se realizează detecția și clasificarea anomaliei. Simptome tipice sunt: amplitudinea semnalelor, valoarea medie, valorile limită, tendința, momentele

statistice ale distribuției amplitudinii sau înfășurătoarea, densitățile spectrale de putere sau liniile spectrale de frecvență, coeficienții de corelație, covarianță etc. Eficiența folosirii analizei de semnal în diagnoză este limitată, în particular, pentru detecția anticipată a anomaliilor sau pentru detecția anomaliilor care apar în dinamica procesului considerat.

Rezultate superioare celor oferite de metodele bazate pe analiza de semnal, se pot obține folosind metode bazate pe modelul procesului. În acest caz, comportarea actuală a procesului este comparată cu cea a modelului corespunzător comportării normale a acestuia. Modelul este condus de aceleași intrări ca și procesul în funcționare actuală, așa cum este prezentat în figura I. 4 .

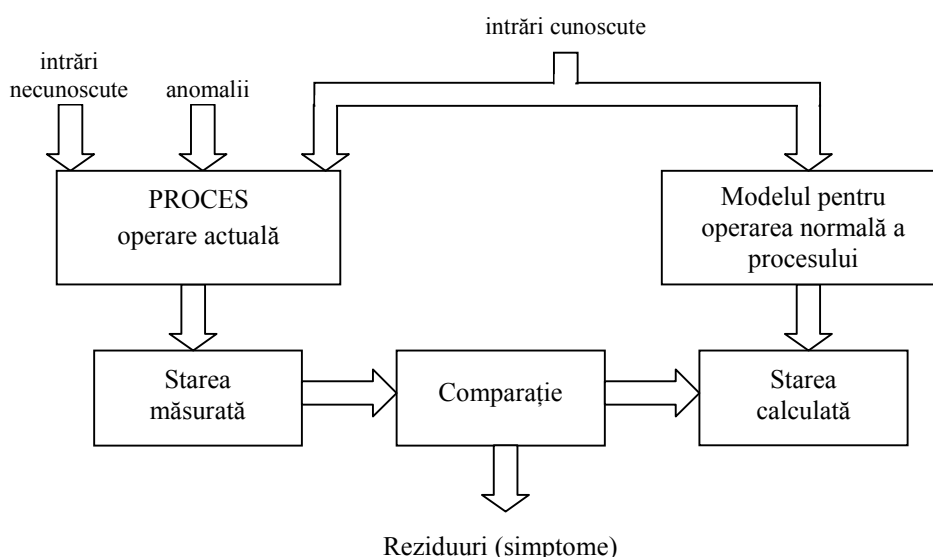


Fig. I. 4 Principiul de funcționare al generatorului de reziduuri

Comportarea dinamică a procesului poate fi descrisă prin:

- modele cantitative (analitice);
- modele bazate pe cunoștințe și rețele neuronale.

Modelul analitic, dacă este disponibil, reprezintă cea mai profundă și concisă modalitate de cunoaștere a procesului. În practică, însă, este dificil să se obțină un model analitic, pentru sisteme complexe acest lucru fiind aproape imposibil. În astfel de situații, modelele bazate pe cunoștințe și rețele neuronale rămân

singura alternativă reală care să permită folosirea cunoștințelor euristice sau a descrierilor calitative în termenii unor reguli și fapte bazate pe observații empirice ale expertului uman, despre comportarea procesului. Din acest motiv, în ultimii ani, s-a acordat o atenție deosebită abordărilor bazate pe modele calitative ale procesului. Aceste metode nu cer un model analitic complet al procesului și permit folosirea cunoștințelor empirice despre proces [Frank, 1994b; Marcu, 1995].

A. Metode analitice de diagnoză a anomaliilor proceselor tehnice

Metodele analitice de diagnoză a anomaliilor folosesc, în prima etapă de generare a reziduului (semnal ce indică apariția unei anomalii în proces), modelul analitic al procesului. Pentru că, în practică, o modelare matematică exactă a procesului este imposibil de făcut, trebuie să se țină seama de erorile de modelare în raport cu care reziduul generat trebuie să fie robust.

Metodele analitice de diagnoză dezvoltate în ultimii 25 de ani se pot grupa în trei categorii, [Isermann, 1994]:

- abordarea bazată pe ecuații de paritate;
- abordarea bazată pe observări;
- abordarea bazată pe estimarea parametrilor fizici ai procesului.

A.1 Abordarea bazată pe ecuații de paritate

Principiul metodei [Isermann, 1994] se descrie în cele ce urmează. Se consideră un sistem dinamic liniar monovariabil, descris de funcția de transfer:

$$G_P(s) = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)}, \quad (\text{I. 1})$$

unde $Y(s)$ și $U(s)$ sunt transformatele Laplace ale mărimilor de ieșire $y(t)$ și, respectiv, de intrare $u(t)$ ale procesului considerat. Dacă atât structura cât și parametrii modelului asociat sistemului considerat sunt cunoscuți, atunci se poate scrie:

$$G_M(s) = \frac{Y_M(s)}{U(s)} = \frac{B_M(s)}{A_M(s)}, \quad (I. 2)$$

unde $G_M(s)$ este funcția de transfer a modelului procesului și $Y_M(s)$ este transformata Laplace a mărimii de ieșire a modelului.

Fie $f_u(t), f_y(t)$ anomalii aditive la intrarea și la ieșirea procesului considerat, iar $F_u(s), F_y(s)$ transformatele Laplace corespunzătoare. În acest caz, ieșirea procesului este:

$$Y(s) = G_P(s) \cdot U(s) + G_P(s) \cdot F_u(s) + F_y(s). \quad (I. 3)$$

Dacă $G_P(s) = G_M(s)$, atunci transformata Laplace a erorii la ieșire este:

$$E_1(s) = Y(s) - Y_M(s) = G_P(s) \cdot F_u(s) + F_y(s) \quad (I. 4)$$

Se observă că anomaliile ce apar la intrarea sau ieșirea procesului influențează eroarea la ieșire, $E_1(s)$.

Se poate folosi și exprimarea polinomială:

$$E_2(s) = A_M(s) \cdot Y(s) - B_M(s) \cdot U(s). \quad (I. 5)$$

Ținând cont de anomalii și presupunând că modelul este exact ($G_P(s) = G_M(s)$), atunci:

$$E_2(s) = B_P(s) \cdot F_u(s) + A_P(s) \cdot F_y(s) \quad (I. 6)$$

Ecuțiile (I. 4) și (I. 6) se numesc *ecuații de paritate*.

Reziduul poate fi obținut filtrând erorile furnizate de către expresiile (I.4) și (I.6):

$$R(s) = G_F(s) \cdot E(s), \quad (I. 7)$$

unde $E(s)$ poate fi $E_1(s)$ sau $E_2(s)$, iar $G_F(s)$ este funcția de transfer a filtrului.

Diagnoza bazată pe ecuațiile de paritate se bazează pe testarea consistenței acestora folosind semnale măsurate din comportarea actuală a procesului. Folosirea filtrului $G_F(s)$ are drept scop decuplarea reziduurilor de variabilele de stare ale procesului și decuplarea diferitelor anomalii pentru a le putea

diagnostica corect. Folosind ecuațiile de paritate se poate descrie ușor efectul anomaliilor aditive asupra reziduurilor, în schimb influența variațiilor în parametri (anomalii multiplicative) este dificil de descris. Acestea din urmă se referă la anomalii reflectate de către modificări ale parametrilor polinoamelor ce constituie funcția de transfer.

A.2 Abordarea bazată pe observeri de stare

Idea de bază a acestei abordări [Isermann, 1994; Frank, 1994b; Frank, 1996; Isermann, 1997] este de a genera reziduul pe baza estimării ieșirilor procesului, folosind observeri Luenberger sau filtre Kalman. Astfel, eroarea de estimare sau, respectiv, inovația, pot fi considerate reziduuri, acestea fiind influențate de anomaliile ce pot apare în funcționarea procesului. Figura I. 5 ilustrează principiul acestei abordări.

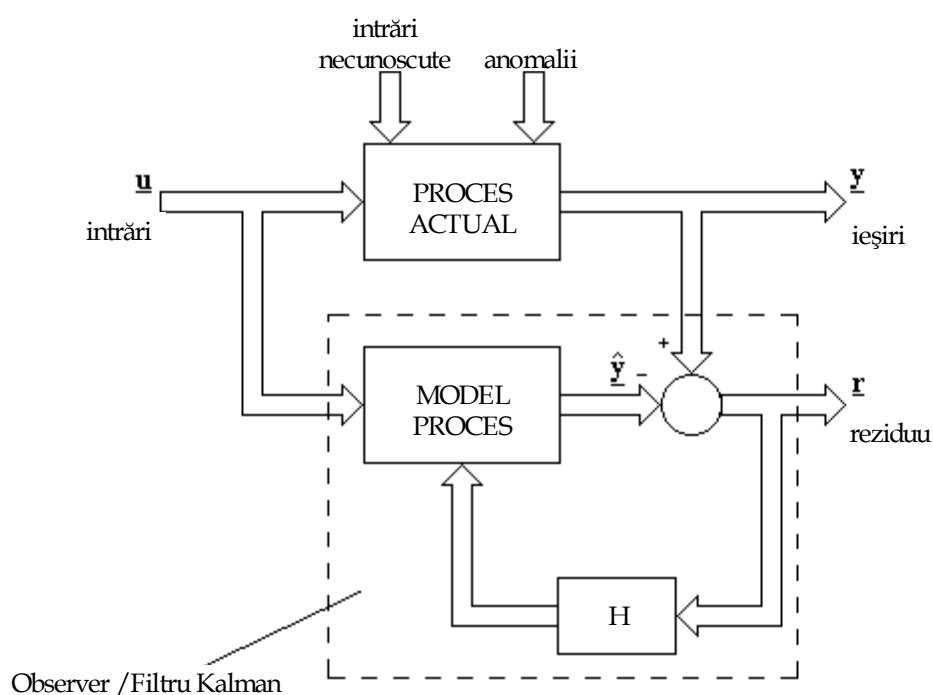


Fig. I. 5 Observer de stare pentru diagnoza anomaliilor

Observerul constă dintr-un model al procesului cu reacție inversă după eroarea de estimare $e = y - \hat{y}$, H . Reacția inversă este necesară din următoarele motive:

- realizează compensarea diferențelor dintre proces și model datorate

condițiilor inițiale;

- stabilizează modelul, în cazul proceselor instabile;
- asigură libertate în proiectarea observerului pentru decuplarea efectelor anomaliilor de efectele intrărilor necunoscute asupra reziduurilor generate;
- permite izolarea anomaliilor (decuplarea anomaliilor una față de cealaltă).

Pentru prezentarea principiului acestei abordări, se consideră un model de stare liniarizat după un punct de funcționare pentru un proces multivariabil, constant, cu parametri concentrați. Se mai consideră că procesul este influențat de perturbații, $\underline{v}(t)$ la intrare și $\underline{n}(t)$ la ieșire și anomaliile aditive care pot acționa la intrare, $\underline{f}_u(t)$, sau la ieșire, $\underline{f}_y(t)$. În acest caz, se pot scrie următoarele ecuații de stare asociate procesului:

$$\begin{aligned}\dot{\underline{x}}(t) &= \underline{A} \cdot \underline{x}(t) + \underline{B} \cdot \underline{u}(t) + \underline{F} \cdot \underline{v}(t) + \underline{L} \cdot \underline{f}_u(t) \\ \underline{y}(t) &= \underline{C} \cdot \underline{x}(t) + \underline{N} \cdot \underline{n}(t) + \underline{M} \cdot \underline{f}_y(t)\end{aligned}\tag{I. 8}$$

Ecuațiile observerului de stare sunt [Voicu, 1986]:

$$\begin{aligned}\frac{d\hat{\underline{x}}(t)}{dt} &= \underline{A} \cdot \hat{\underline{x}}(t) + \underline{B} \cdot \underline{u}(t) + \underline{H} \cdot \underline{e}(t) \\ \underline{e}(t) &= \underline{y}(t) - \underline{C} \cdot \hat{\underline{x}}(t)\end{aligned}\tag{I. 9}$$

Eroarea de estimare a stării este:

$$\underline{\varepsilon}(t) = \underline{x}(t) - \hat{\underline{x}}(t)\tag{I. 10}$$

și evoluția sa în timp, în ipoteza absenței perturbațiilor și zgomotului ($\underline{v}(t) = \underline{n}(t) = 0$), este dată de ecuația:

$$\begin{aligned}\dot{\underline{\varepsilon}}(t) &= [\underline{A} - \underline{H} \cdot \underline{C}] \cdot \underline{\varepsilon}(t) + \underline{L} \cdot \underline{f}_u(t) - \underline{H} \cdot \underline{M} \cdot \underline{f}_y(t) \\ \lim_{t \rightarrow \infty} \underline{\varepsilon}(t) &= 0\end{aligned}\tag{I. 11}$$

Eroarea de ieșire devine:

$$\underline{e}(t) = \underline{C} \cdot \underline{\varepsilon}(t) + \underline{M} \cdot \underline{f}_y(t).\tag{I. 12}$$

Eroarea de estimare a stării, cât și eroarea de ieșire vor devia de la zero atunci când apar anomalii aditive, după cum se poate observa și din expresiile acestor două marimi. Cele două tipuri de erori pot fi considerate reziduuri.

Dacă anomaliile apar ca variații ale parametrilor procesului (ΔA , ΔB , ΔC), comportarea procesului va fi descrisă de ecuațiile:

$$\begin{aligned}\dot{\underline{x}}(t) &= (A + \Delta A) \cdot \underline{x}(t) + (B + \Delta B) \cdot \underline{u}(t) \\ \underline{y}(t) &= (C + \Delta C) \cdot \underline{x}(t)\end{aligned}\tag{I. 13}$$

În această situație, ecuațiile ce descriu comportarea dinamică a erorii de estimare a stării și a erorii de ieșire vor fi:

$$\begin{aligned}\dot{\underline{e}}(t) &= [A - H \cdot C] \cdot \underline{e}(t) + [\Delta A - H \cdot \Delta C] \cdot \underline{x}(t) + \Delta B \cdot \underline{u}(t) \\ \underline{e}(t) &= C \cdot \underline{e}(t) + \Delta C \cdot \underline{x}(t)\end{aligned}\tag{I. 14}$$

Variațiile în parametri influențează cele două tipuri de erori prin multiplicare cu variabilele de stare și, respectiv, cu cele de intrare. Din acest motiv, anomaliile ce se manifestă prin variații ale parametrilor procesului se numesc *anomalii multiplicative*. În acest caz, reziduurile vor varia odată cu parametrii procesului, precum și odată cu variabilele de stare și de intrare. Pentru diagnoza acestui tip de anomalii este absolut necesar ca mărimile de ieșire să fie măsurabile și se face uz de conceptul de *redundanța analitică*.

A.3 Abordarea bazată pe estimarea parametrilor fizici ai procesului

Această abordare se bazează pe faptul că anomaliile sunt reflectate în parametrii fizici ai procesului, [Isermann, 1994; Marcu, 1995]. Ideea de bază a acestei metode de detecție este aceea că parametrii fizici ai procesului, corespunzător comportării curente, sunt estimați în mod repetat și comparați cu valorile corespunzătoare obținute inițial în condiții de funcționare normală (absență a anomaliilor). Orice discrepanță detectabilă între aceste valori indică o schimbare în proces și poate fi interpretată drept anomalie.

Etapele ce trebuie parcurse în acest caz sunt următoarele [Marcu, 1995]:

1. găsirea unui model parametric al procesului de forma:

$$\underline{y} = \underline{f}(\underline{u}, \underline{n}, \underline{\theta}, t),$$

unde \underline{u} reprezintă vectorul intrărilor procesului, \underline{y} reprezintă vectorul ieșirilor procesului, \underline{n} reprezintă vectorul intrărilor necunoscute (zgomote, perturbații aleatoare etc.), $\underline{\theta}$ reprezintă vectorul parametrilor modelului

procesului și t reprezintă variabila timp;

2. determinarea relațiilor dintre parametrii modelului $\underline{\theta}$ și parametrii fizici ai procesului \underline{p} :

$$\underline{\theta} = \underline{h}(\underline{p}); \quad \underline{p} = \underline{h}^{-1}(\underline{\theta}),$$

unde funcția \underline{h} este, în general, neliniară;

3. identificarea parametrilor modelului folosind mărimile corespunzătoare funcționării curente a procesului, $\hat{\underline{\theta}}$;
4. determinarea parametrilor procesului, $\hat{\underline{p}} = \underline{h}^{-1}(\hat{\underline{\theta}})$;
5. calculul deviațiilor parametrilor fizici ai procesului estimați față de valorile corespunzătoare funcționării normale \underline{p}_{H_0} :

$$\Delta \underline{p}_{H_0} = \hat{\underline{p}} - \underline{p}_{H_0};$$

dacă există modele ale procesului corespunzătoare comportărilor anormale, atunci se calculează și deviațiile:

$$\Delta \underline{p}_{H_k} = \hat{\underline{p}} - \underline{p}_{H_k}, \quad k = 1, 2, 3, \dots,$$

unde \underline{p}_{H_k} reprezintă vectorul parametrilor procesului corespunzător funcționării anormale k .

6. decizia asupra existenței anomaliei, utilizând relațiile dintre anomalii și schimbările în parametrii procesului;
7. localizarea anomaliei pe baza relațiilor dintre anomalii și schimbările în parametrii procesului.

Figura I. 6 prezintă schema principială corespunzătoare etapelor 3-7 [Marcu, 1995].

Avantajele acestei abordări sunt:

- în general, sunt necesare mai puține informații apriorice asupra modelului matematic;
 - anomaliile pot fi detectate anticipat și localizate mai precis decât verificând schimbările survenite în ieșirile procesului;
 - se pot extrage informații semnificative despre anomalie, reprezentate prin
-

schimbări ale parametrilor procesului, ceea ce duce la îmbunătățirea sensibilității schemei de diagnoză;

- se pot detecta anomalii incipiente ale sistemelor lucrând în circuit închis;
- necesită un număr limitat de senzori a căror funcționare să fie robustă la anomalii și perturbații.

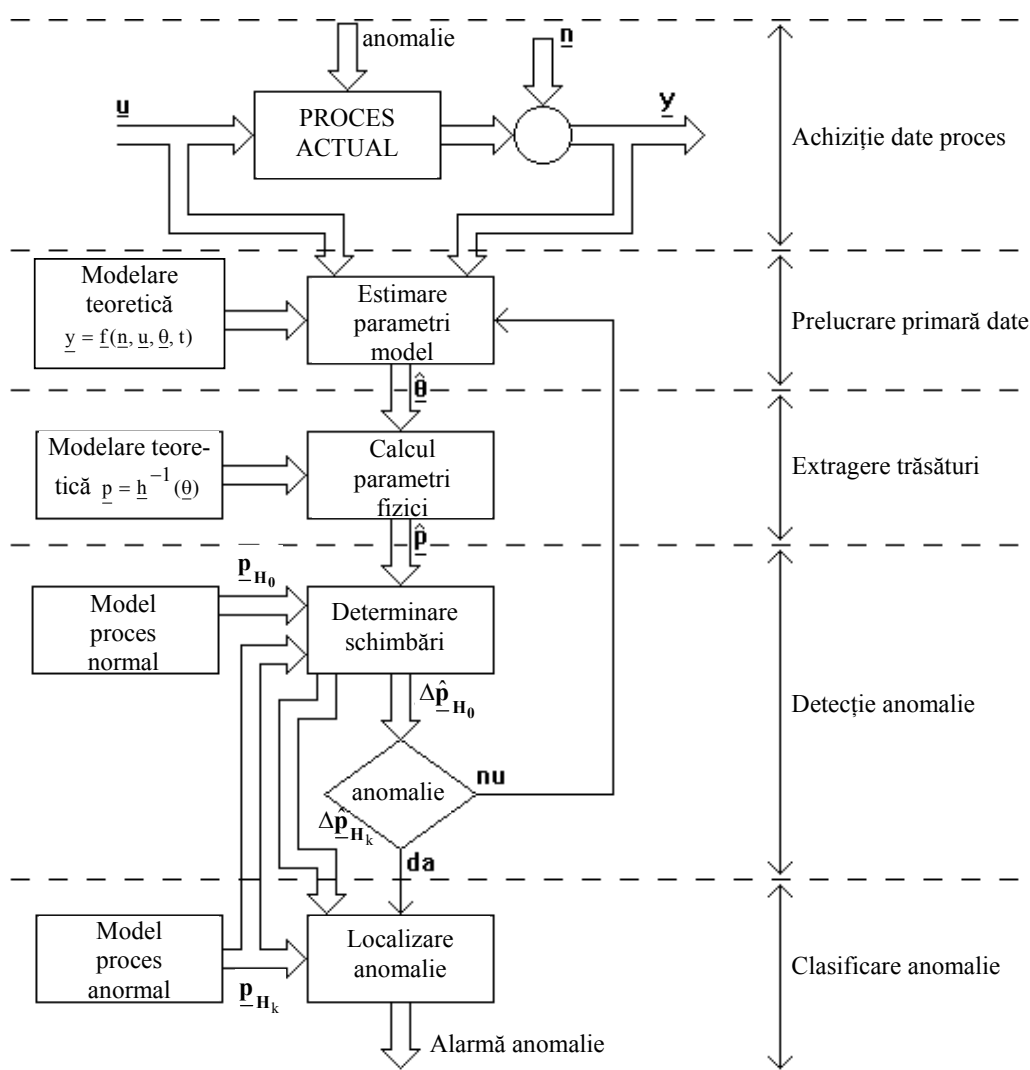


Fig. I. 6 Reprezentarea schematică a metodei de diagnoză a anomaliilor bazată pe estimarea parametrilor procesului

Dezavantajele abordării sunt:

- necesitatea unor semnale suplimentare de excitație în faza de identificare a parametrilor modelului;

- în general nu există o relație unică între parametrii modelului și coeficienții fizici;
- metodele bazate pe estimarea de parametri au aplicabilitate limitată în cazul sistemelor de ordin ridicat sau care conțin neliniarități esențiale;
- necesită un volum de calcul mare datorat repetării procedurii de identificare;
- estimarea tuturor parametrilor procesului în regim staționar nu este posibilă întotdeauna;
- robustețe scăzută a schemei de diagnoză în raport cu incertitudini asupra modelului, perturbații și zgomot.

B. Metode bazate pe cunoștințe și rețele neuronale pentru diagnoza anomaliilor proceselor tehnice

Abordarea analitică a diagnozei anomaliilor proceselor are dezavantajul că, în condiții reale, nu se poate obține un model matematic exact pentru sistemul supus observației. Tehnicile analitice avansate de proiectare a unor subsisteme de diagnoză robuste în raport cu intrările necunoscute pot anula acest neajuns numai până la un anumit nivel și cu un efort deosebit de proiectare [Wünnenberg, 1990; Patton, 1994; Marcu, 1995].

În cazul unor incertitudini de modelare pronunțate, o strategie adecvată este cea a folosirii *tehnichilor bazate pe cunoștințe*. În această situație, robustețea schemei de diagnoză poate fi obținută utilizând simptome care nu sunt deloc sau nu sunt puternic dependente de incertitudinile sistemului. Pe de altă parte, evaluarea reziduurilor este un proces logic complex care oricum cere folosirea unor tehnici inteligente de luare a deciziei cum ar fi: *căutarea în arborii de defectare* asociați sistemului, *recunoașterea de forme* folosind tehnicile fuzzy și rețele neuronale artificiale. Metode bazate pe cunoștințe pot fi aplicate în toate cele trei faze ale diagnozei anomaliilor proceselor și anume: generarea de reziduuri, evaluarea reziduurilor și analiza anomaliilor [Frank, 1994b].

Metodele de diagnoză bazate pe cunoștințe pot fi împărțite în două categorii [Frank, 1994b; Marcu, 1995]:

a) *metode bazate pe simptome*

Aceste metode folosesc simptome euristice, cunoștințe despre istoria procesului analizat și cunoștințe de natură statistică. Evaluarea acestor cunoștințe este făcută în cadrul unor sisteme expert pentru diagnoză. Problema principală, în acest caz, este achiziția de cunoștințe.

b) *metode bazate pe modele calitative*

Aceste metode folosesc cunoștințe exprimate sub formă de fapte și reguli derivate din descrierea structurii și a comportării sistemului, aceste informații putând fi incomplete și cu un anumit grad de incertitudine comparativ cu modelul analitic.

B.1 Concepte de bază și structura sistemelor expert [Winston, 1981; Georgescu, 1985; Malița, 1987; Pănescu, 2000]

Sistemul expert este un program ce înmagazinează cunoștințe ale unor experți umani și poate rezolva probleme complexe din domeniul de expertiză respectiv. Acest program utilizează cunoștințe și procedee de raționament pentru a rezolva probleme ce sunt destul de dificile, necesitând, în mod obișnuit, pentru obținerea soluției, un expert uman în domeniu.

Expertul este acea persoană ce posedă cunoștințe deosebite, fiind capabil să efectueze o expertiză într-un anumit domeniu. Utilizatorul furnizează sistemului expert sau sistemului bazat pe cunoștințe fapte care constituie o problemă și sistemul îi furnizează la ieșire soluția, anume expertiza la problema respectivă.

Așa cum se poate observa din figura I. 7 , sistemul expert conține o *bază de cunoștințe* în care sunt stocate elementele specifice domeniului de expertiză, actualizate prin faptele introduse de utilizator și *motorul de inferențe* care posedă elementele de control ale rulării programului.

Față de programarea convențională, într-un sistem expert apare o separare netă între cunoștințele în domeniu și elementele de control ale rulării programelor.

Aceasta determină următoarele deosebiri dintre sistemele expert și programarea convențională:

- stocarea cunoștințelor în cadrul sistemelor expert se face într-o formă ușor accesibilă modificărilor, dezvoltării, actualizărilor, lucru care în programarea convențională poate fi făcut doar de către programator;
- un sistem expert poate explica raționamentele efectuate pentru a ajunge la un rezultat dacă sunt evidențiați pașii parcurși pentru obținerea soluției, în timp ce un program convențional se limitează doar la a furniza soluția;
- este posibilă construirea bazei de cunoștințe într-o formă declarativă atât ca rezultat al separării cunoștințelor cât și prin folosirea unor tehnici adecvate de reprezentare a cunoștințelor, trecerea la forma procedurală făcându-se automat, utilizatorul realizând baza de cunoștințe într-o formă apropiată de limbajul natural.

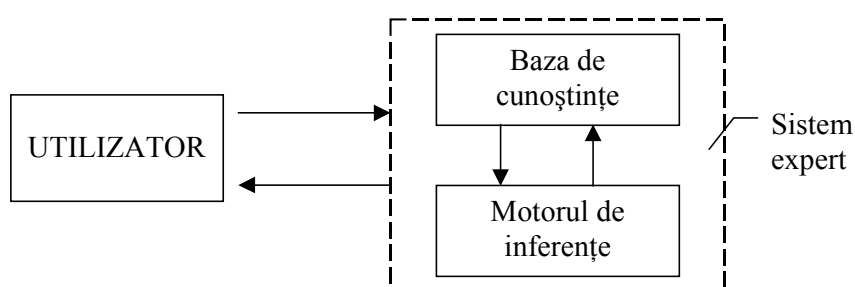


Fig. I. 7 Sistem expert – funcționare principală

Domeniile în care au fost realizate și aplicate sistemele expert sunt:

- *configurare*: stabilirea componentelor unui sistem și a modalității de conectare a acestora pentru a satisface anumite performanțe impuse;
- *diagnoză*: pe baza unor efecte observate se determină cauzele ce le-au generat;
- *instruire*: perfecționarea (educarea) într-un anumit domeniu;
- *monitorizare*: compararea unor date experimentale cu cele impuse și luarea măsurilor necesare păstrării performanțelor dorite;
- *planificare*: stabilirea secvenței de acțiuni pentru îndeplinirea unui scop dat;
- *conducerea proceselor*: îmbinarea elementelor de monitorizare, planificare și

diagnoză.

În proiectarea unui sistem expert trebuie rezolvate următoarele probleme:

- realizarea bazei de cunoștințe;
- realizarea motorului de inferențe.

Cunoștințele specifice domeniului de expertiză apar numai în baza de cunoștințe sub formă de constante, variabile, relații existente între variabile, reguli derivate din teoria domeniului respectiv sau euristice (derivate din experiența în domeniul respectiv).

Elementele de control ce constituie motorul de inferențe determină momentul când elemente din baza de cunoștințe devin relevante pentru problema ce se rezolvă și trebuie activate sau testate. *Motorul de inferențe* conține un algoritm generic care este același indiferent de conținutul bazei de cunoștințe.

Structura unui sistem expert (sau bazat pe cunoștințe) este descrisă în figura I. 8 .

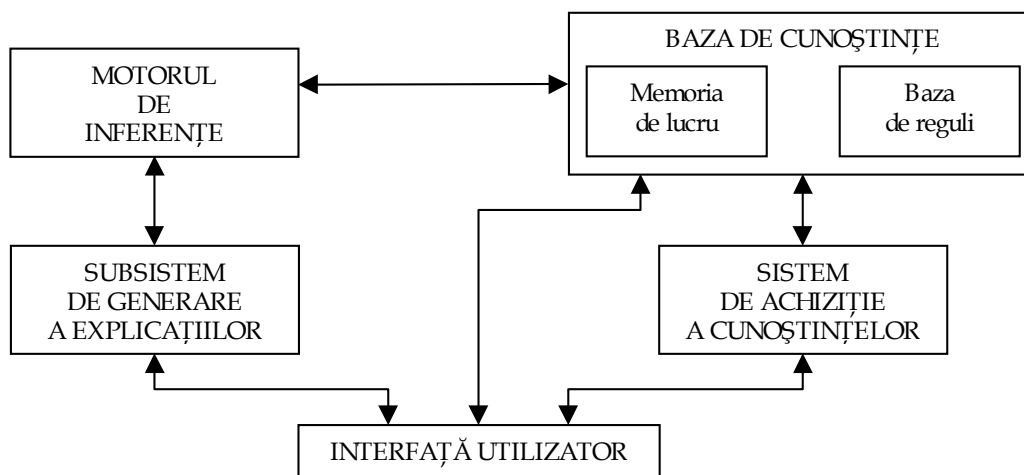


Fig. I. 8 Structura unui sistem expert

Baza de cunoștințe este formată din două părți:

- *baza de reguli*, ce conține cunoștințe generale din domeniul de expertiză;
- *baza de fapte* (memoria de lucru), ce conține cunoștințe specifice pentru problema care se rezolvă.

Regulile din baza de reguli, numite reguli de producție, sunt de forma:

DACĂ < parte condiție > ATUNCI < parte acțiune >

Cunoștințele din memoria de lucru care sunt fapte considerate ca fiind adevărate pentru problema ce se rezolvă, au aceeași formă ca și părțile de condiție ale regulilor.

Motorul de inferențe conține un mecanism de control ce permite efectuarea de raționamente care să genereze noi fapte adevărate prin aplicarea regulilor activate de faptele din memoria de lucru. Principial, motorul de inferențe trebuie să selecteze din baza de reguli acele reguli ce au partea de condiție satisfăcută conform faptelor din memoria de lucru la momentul respectiv și să determine punerea în aplicare a părții de acțiune a regulilor satisfăcute. Toate inferențele care se desfășoară într-un sistem expert se reduc la regula de deducție “modus ponens”:

fiind dat un fapt adevărat, A, ce activează o regulă, $A \Rightarrow B$, se deduce că faptul B este adevărat.

În funcție de modul de căutare a regulilor, există două tipuri de motoare de inferențe:

- a) *strategia de control progresivă* care pornește de la fapte pentru a ajunge la obiective;
- b) *strategia de control regresivă* care pornește de la scopuri (obiective), sistemul expert găsind faptele ce le determină.

Executarea unei reguli poate avea ca efect modificarea memoriei de lucru, transmiterea de mesaje către operator sau comenzi către proces. Când căutarea realizată de motorul de inferențe în baza de reguli eșuează, acesta poate adresa o întrebare (cerere) utilizatorului prin intermediul inferenței cu operatorul.

Interfața utilizator este conectată cu două componente, care sunt opționale, ale sistemului expert:

- *subsistemul de generare de explicații*: permite, ghidat de motorul de

inferențe, evidențierea pașilor parcurși până la obținerea soluției prin memorarea ordinii în care au fost executate regulile;

- *subsistemul de achiziție de cunoștințe*: permite modificarea bazei de reguli prin adăugarea de noi elemente sau utilizându-le pe cele existente.

Există și sisteme expert care efectuează raționamente aproximative. Regulile din baza de reguli a unui sistem expert se pot împărți în:

- *reguli definiționare*: stabilesc o legătură sigură între termenii cuprinși în partea de condiție și cei din partea de acțiune;
- *reguli probabilistice*: exprimă un raționament care nu este sigur, ci are un anumit grad de incertitudine.

Sistemele expert ce lucrează cu reguli probabilistice și cu fapte care nu mai sunt cu certitudine sigure fac apel la așa-numiții *factori de încredere* care sunt coeficienți numerici atașați faptelor, respectiv regulilor, exprimând încrederea acordată acestora. Factorii de încredere nu se pot încadra strict în niște modele probabilistice, ci caută să capteze elemente euristice pe care le folosesc experții în timpul raționamentelor.

Există două tipuri de factori de încredere:

- *încrederea expertului*, pe care acesta o acordă în utilizarea unei anumite reguli;
- *încrederea utilizatorului*, pe care acesta o acordă unui fapt când este introdus în memoria de lucru (context, baza de fapte).

Nici expertul și nici utilizatorul nu pot furniza cifre exacte pentru factorii de încredere. De aceea se folosește o scală cu valori în intervalul $[-1,1]$ cu diferite porțiuni notate corespunzător, astfel ca indicațiile calitative ale expertului sau utilizatorului să poată fi transformate în coeficienți numerici, cum este arătat în figura I. 9.

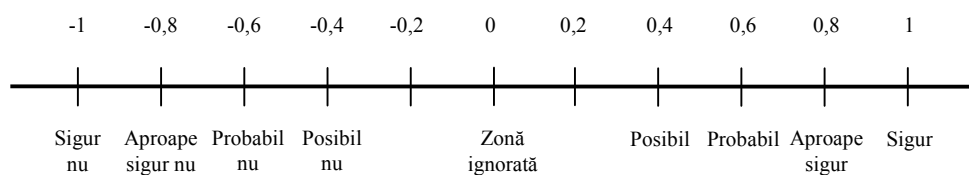


Fig. I. 9 Exemplu de atribuire a factorilor de încredere

O alternativă de modelare a incertitudinii într-un sistem expert, față de folosirea factorilor de încredere, o reprezintă sistemele expert de tip fuzzy.

B.2 Sistem expert pentru diagnoza anomaliilor bazat pe simptome

Sistemul de diagnoză a anomaliilor bazat pe simptome are structura prezentată în figura I. 10 [Isermann and Freyermuth, 1991]. Acesta conține o soluție analitică a problemei, o bază de cunoștințe asupra procesului, o componentă de achiziție a cunoștințelor din proces, precum și un mecanism de deducții.

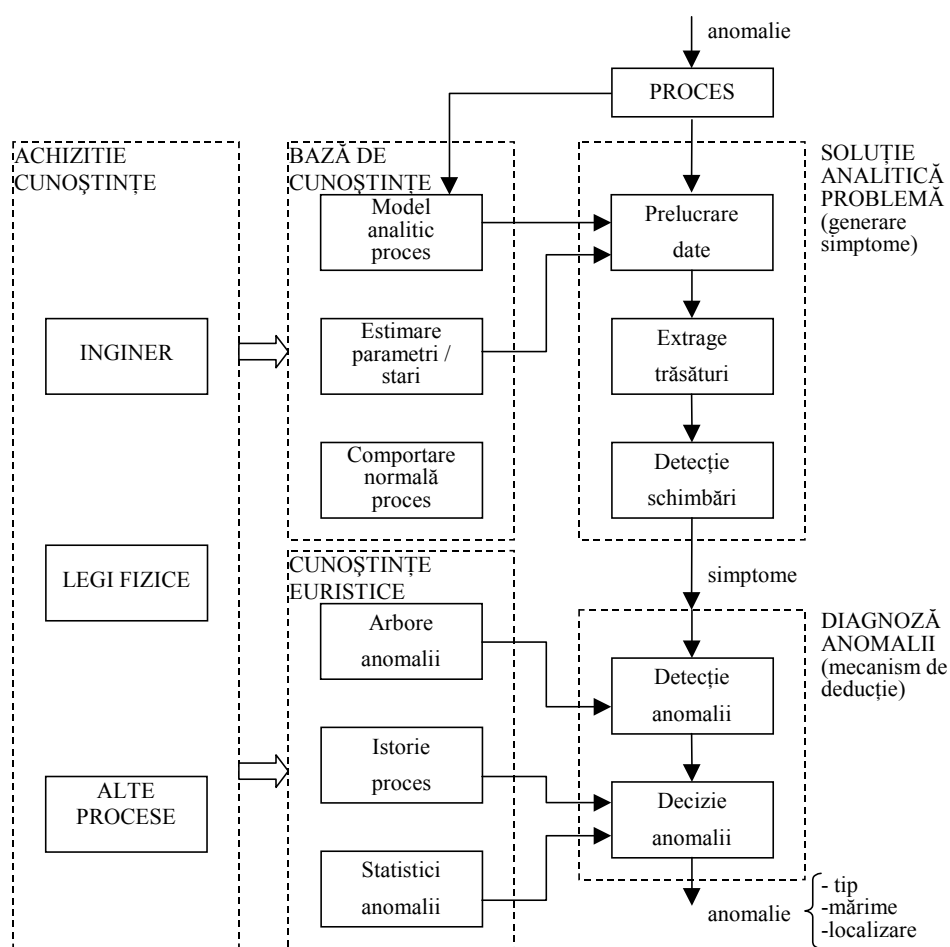


Fig. I. 10 Structura sistemului expert bazat pe simptome pentru diagnoza anomaliilor

Soluția analitică a problemei folosește estimațiile parametrilor sau stărilor procesului pentru a detecta schimbările apărute față de valorile corespunzătoare din comportarea normală, generându-se simptome ale anomaliiei. Acest tip de soluție cuprinde o parte de prelucrare a datelor, o parte de extragere a trăsăturilor și o parte de detecție a schimbărilor.

Baza de cunoștințe cuprinde cunoștințe analitice reprezentate prin modele analitice ale procesului și cunoștințe euristice cum ar fi arbori de defectare, statistici ale anomaliilor etc.

Componenta de *achiziție a cunoștințelor din proces* se bazează pe observațiile expertului uman, legile fizice ce guvernează procesul în atenție, cunoștințe despre comportarea normală a procesului și arborii de defecte.

Mecanismul de deducție realizează, de fapt, diagnoza anomaliilor pe baza simptomelor generate, a arborilor de defecte, probabilităților de apariție a anomaliilor și istoricului procesului analizat. Rezultatul final este constituit din tipul anomaliei, mărimea și cauza acesteia.

În mod uzual datele din proces sunt, în primă fază, analizate prin metode cum ar fi: filtre speciale, analiză spectrală sau estimare de parametri sau stări, dacă procesul este bine cunoscut analitic. Dacă problema diagnozei nu este rezolvabilă în întregime prin metode matematice, informațiile analitice din proces împreună cu cele euristice sunt prelucrate de către un mecanism de inferențe care, în final, propune soluții prin mesaje lansate către operator. Astfel se obține *sistemul expert pentru diagnoză* care, dacă este cuplat cu ieșirile din proces, devine un sistem expert on-line. Operatorul poate interveni schimbând intrările în proces sau inserând noi cunoștințe în baza de cunoștințe într-o manieră interactivă.

Sistemele expert cu cunoștințe analitice și euristice parțiale reprezintă o combinație între analiza inginerescă, tehnicile de prelucrare matematică și tehnicile tradiționale ale sistemelor expert.

B.3 Sistem expert pentru diagnoza anomaliilor bazat pe modele calitative

Metodele bazate pe raționamentul calitativ, dezvoltate în ultima perioadă, acoperă un gol în ceea ce privește modelarea, controlul și diagnoza proceselor, atunci când cunoștințele incomplete despre sistem fac ca metodele analitice să

fie dificil de aplicat. Un model calitativ poate suplini cunoștințele incomplete despre proces printr-o mulțime redusă de valori de referință pentru variabilele din proces și relații funcționale între acestea, fără a incorpora presupuneri referitoare la liniaritate și valori specifice pentru anumiți parametri incomplet cunoscute. Chiar dacă cunoștințele sunt incomplete, există suficientă informație într-o descriere calitativă pentru a realiza o simulare calitativă a procesului. În acest fel, pot fi predictate posibilele comportări ale sistemului incomplet descris.

Modelul asociat unui sistem, capabil să furnizeze predicții asupra unor posibile comportări ale acestuia, trebuie să satisfacă următoarele *cerințe* [Kuipers, 1989]:

- modelul trebuie să exprime ceea ce se cunoaște despre proces;
- modelul nu trebuie să se bazeze pe presupuneri, în afara a ceea ce se cunoaște;
- trebuie să existe posibilitatea (matematică și de calcul) de a realiza predicții;
- trebuie să existe posibilitatea de a compara predicțiile cu observațiile.

Metodele bazate pe raționamentul calitativ furnizează mai multă putere de expresie pentru cunoașterea incompletă a proceselor, prin intermediul reprezentării valorilor variabilelor în cadrul unui *spațiu cantitativ cu rezoluție slabă* și prin intermediul unor *clase de funcții monotone* utilizate pentru modelarea relațiilor funcționale dintre variabilele procesului.

Conceptele de bază ale simulării calitative [Kuipers, 1989] sunt următoarele:

1. Valoarea unei variabile din proces este descrisă calitativ, în termenii unui spațiu al cantităților definit printr-o mulțime ordonată de valori importante, exprimate lingvistic (de exemplu: zero, mediu, mare etc.); astfel, *spațiul cantităților* definește un limbaj descriptiv, reprezentând mulțimea de direcții calitative ce pot fi exprimate de model.
2. Cunoștințele incomplete asupra legilor ce guvernează comportarea procesului sunt suplinite prin clase de funcții monotone ce descriu dependențele funcționale dintre variabile și care sunt denumite *restricții*

calitative. În ipoteza că anumite dependențe funcționale descriu viteza de schimbare a unor variabile ca funcții de alte variabile, se obțin *ecuații diferențiale calitative*.

Spațiile cantităților asociate variabilelor din proces și restricțiile calitative asociate dependențelor funcționale dintre variabilele din proces constituie *cunoștințe structurale* materializate într-un *model calitativ*.

3. Modelul calitativ obținut pe baza cunoștințelor structurale este incomplet dacă nu se includ și informații asupra stărilor procesului. *Starea calitativă* a sistemului este descrisă de *valori calitative ale variabilelor* precum și de *direcțiile de schimbare* ale fiecărei variabile. Cum stările sistemului se schimbă în timp, este necesar să se descrie secvențe de stări calitative prin care sistemul poate trece.
4. Odată sintetizat modelul calitativ al procesului, prin *simulare calitativă*, plecând de la o stare inițială, sunt furnizate comportările posibile ale procesului. Simularea calitativă are loc sub următoarele *presupuneri*:
 - nu numai variabilele se schimbă, ci și derivatele lor;
 - secvența de stări calitative distincte ale sistemului este discretă; noțiunea de stare următoare a sistemului înseamnă, deci, următoarea descriere calitativă, distinctă de cea precedentă, din această secvență.Astfel, timpul este modelat ca o secvență alternantă de momente și de intervale deschise. Plecând de la un moment de timp căruia îi corespunde o anumită stare calitativă a sistemului, urmează un interval de timp în care starea nu se modifică, până se atinge un alt moment de timp corespunzător unei modificări de stare.

Limitările principale ale simulării calitative sunt [Shen and Leitch, 1993]:

- *ambiguitate calitativă*: apare datorită ambiguității inerente a calculului calitativ, manifestată prin generarea a numeroase comportări care tind să ascundă comportarea reală.
- *lipsa informațiilor temporale*: apare datorită faptului că tehnicile uzuale nu explicitează clar informațiile legate de durată.

O soluție pentru a depăși limitările simulării calitative este de a combina raționamentul calitativ cu logica bazată pe teoria mulțimilor vagi, astfel dezvoltându-se *tehnici de simulare calitativă fuzzy* care permit o extindere semi-cantitativă a simulării calitative. În acest fel, conceptele fundamentale ale simulării calitative sunt păstrate, teoria mulțimilor vagi furnizând doar instrumente practice pentru punerea în aplicare a acestor principii [Shen and Leitch, 1993]:

- reprezentarea informațiilor sub forma funcțiilor de apartenență și a relațiilor fuzzy;
- prelucrarea informațiilor prin intermediul logicii fuzzy.

Odată cu dezvoltarea teoriei raționamentului calitativ în legătură cu sistemele fizice, inițial motivată de probleme de diagnoză, a devenit posibilă supervizarea și diagnoza anomaliilor sistemelor dinamice, făcând apel la simularea calitativă. Schema de diagnoză bazată pe un model calitativ are o configurație asemănătoare cu cea bazată pe modelul analitic (Figura I. 11), numindu-se *observer bazat pe cunoștințe* [Frank, 1994b; Marcu, 1995].

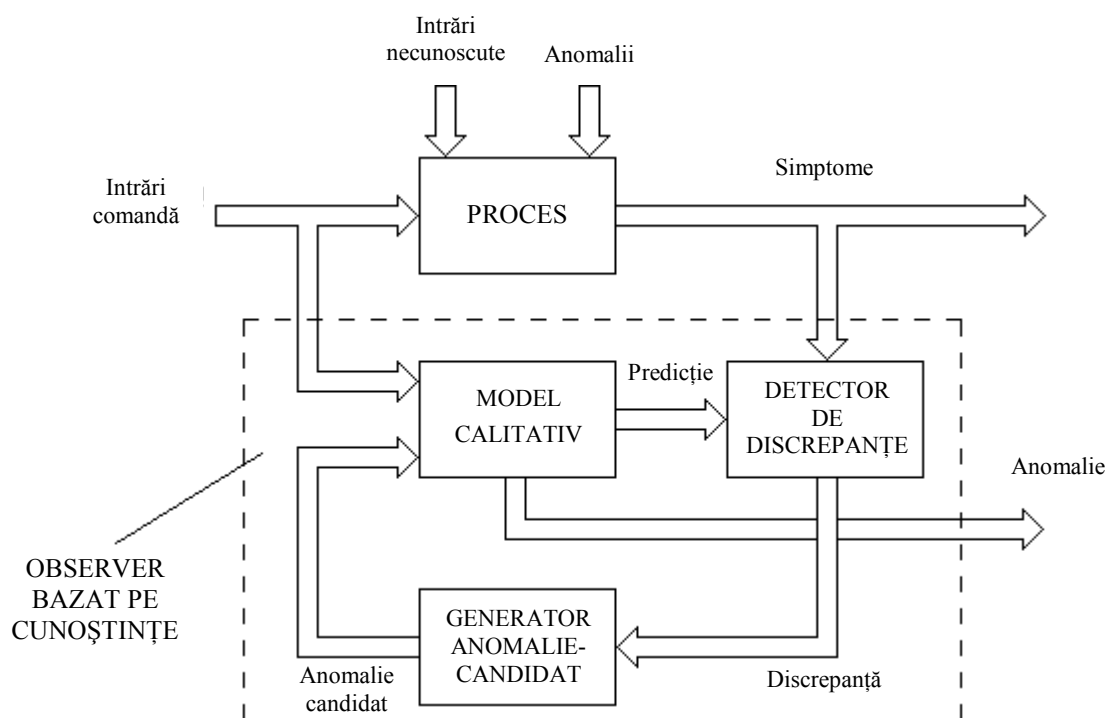


Fig. I. 11 Schema de diagnoză bazată pe modelul calitativ al procesului

Elementul cheie al acestui concept este *modelul calitativ*. Caracteristica modelării calitative este descrierea, printr-un număr mic de simboluri sau valori calitative (“on”, “off” etc.), a comportării dinamice a sistemului. Se pot folosi și valori intermediare imprecise, reprezentate prin intermediul mulțimilor fuzzy (mare, mic etc.), domeniul identificării cu ajutorul sistemelor relaționale fuzzy și al simulării calitative devenind de mare interes și de mare importanță pentru partea de modelare calitativă a sistemului de diagnoză bazat pe cunoștințe.

În cadrul modelării calitative fuzzy, variabilelor dinamice ale procesului li se asociază mulțimi fuzzy caracterizate de funcții de apartenență, iar legăturilor dintre variabile li se asociază reguli exprimate prin relații fuzzy, corespunzătoare legilor fizice ce guvernează comportarea sistemului.

Observerul bazat pe cunoștințe constă din următoarele blocuri funcționale [Frank, 1994b]:

- *modelul calitativ*: determină comportările posibile, obținute prin simulare calitativă;
- *detectorul de discrepanțe*: determină diferențele dintre simptomele măsurate și cele calculate de către modelul calitativ;
- *generatorul de anomalii – candidat*: propune diferite modele anormale, pe baza discrepanțelor detectate;
- *strategul de diagnoză*: coordonează procesul de căutare și asigură faptul că modelul cu care se lucrează urmărește evoluția simptomelor procesului actual.

Rezultă că evaluarea de reziduuri, realizată de detectorul de discrepanțe și generatorul de anomalii – candidat, este parte integrantă a observerului de diagnoză bazat pe cunoștințe.

În afara avantajului legat de robustețea schemei de diagnoză, un alt avantaj îl reprezintă mărirea gradului de înțelegere a rezultatelor obținute, datorită faptului că stările procesului sunt caracterizate de variabile cu valori lingvistice și reguli exprimate într-o manieră mai apropiată de gândirea umană. Acest avantaj este

deosebit de important în ceea ce privește îmbunătățirea comunicării om-mașină din cadrul sistemelor expert folosite în procesul de luare a deciziilor [Frank, 1994b; Marcu, 1995].

B.4 Metode de diagnoză a anomaliilor bazate pe rețele neuronale

Rețelele neuronale artificiale au captat atenția unui număr mare de oameni de știință din diverse domenii cum ar fi: inginerie, matematică, fizică și biologie, datorită abilității lor de a învăța. Proprietatea ființelor umane de a memora și dobândi noi informații, pe parcursul procesului de învățare, a indus oamenilor de știință dorința de a dezvolta un sistem care să îndeplinească o anumită funcție, pe baza informațiilor învățate de acesta în prealabil. În ultimii 30 de ani, au fost dezvoltate un mare număr de tipuri de rețele neuronale, cu structuri și reguli de învățare diferite, deși doar o parte din ele sunt bine fundamentate teoretic. Totuși, numeroasele rezultate experimentale obținute susțin ideea că rețelele neuronale artificiale pot fi aplicate cu succes pentru rezolvarea unor probleme practice din domenii cum ar fi:

- analiza de semnal și recunoașterea de forme;
- comunicații;
- vedere artificială;
- controlul sistemelor: identificarea sistemelor neliniare, controlul neliniar adaptiv, diagnoza anomaliilor proceselor;
- robotică.

La începutul anilor '90, rețelele neuronale artificiale au fost aplicate în scopul diagnozei anomaliilor proceselor complexe, lent variante în timp, pentru care modelul analitic nu era disponibil sau era greu de obținut [Sorsa, *et al.*, 1991; Sorsa, *et al.*, 1993a; Sorsa, *et al.*, 1993b]. Primele aplicații s-au referit la evaluarea simptomelor obținute pe baza mărimilor de ieșire din proces, diferite deviații ale acestora (față de comportarea normală) indicând diverse stări de anomalie.

Principial, o *rețea neuronală artificială* este formată prin dispunerea în paralel a unor unități elementare de prelucrare a informației numite *neuroni*. Aceștia sunt activați de îndată ce semnalele aplicate la intrarea acestora depășesc anumite valori de prag. Neuronii sunt aranjați în straturi care sunt interconectate astfel încât mărimile aplicate la intrarea rețelei să fie propagate, prin intermediul acesteia, către ieșire. Activitatea neuronilor este descrisă prin intermediul unei funcții de transfer, în general, neliniară și a unui set de parametri. Alegerea tipului funcției de transfer a fiecărui neuron al rețelei, imprimă acesteia un anumit comportament neliniar [Frank and Köppen-Seliger, 1997].

Pe parcursul unei faze de antrenare, parametrii neuronilor rețelei sunt adaptați până când se obține un anumit comportament dorit al rețelei. Dacă rețeaua neuronală este folosită pentru diagnoza anomaliilor, atunci antrenarea se realizează folosind măsurătorile mărimilor de intrare - ieșire din procesul analizat, corespunzătoare funcționării normale a procesului și, dacă sunt disponibile măsurători, comportărilor anormale ale procesului.

Ca urmare a cercetării susținute, rețelele neuronale artificiale sunt folosite curent pentru diagnoza anomaliilor, atât în faza de generare a reziduurilor (simptomelor), cât și în faza evaluării acestora.

Capitolul 2

Tendințe actuale în diagnoza bazată pe utilizarea rețelelor neuronale

2.1 Noțiuni introductive privind rețelele neuronale artificiale

Se definesc următorii termeni, [Haykin, 1994; Ayoubi, 1996b]:

- *rețeaua neuronală artificială* este o structură paralelă, distribuită, de prelucrare a informațiilor, care apare sub forma unui graf orientat;
- un *graf orientat* este un obiect geometric compus dintr-un set de puncte numite *noduri* împreună cu un set de segmente orientate dintre noduri numite *legături*;
- în *nodurile* unei rețele neuronale se află elementele de prelucrare ale rețelei numite și *neuroni*;
- legăturile dintre neuronii unei rețele neuronale se numesc *conexiuni*, fiecare conexiune funcționând ca o cale de transmitere instantanee, unidirecțională a informației;
- punctele de conexiune dintre neuroni se numesc *sinapse*.

Fiecare element de prelucrare (neuron) poate primi orice număr de conexiuni de intrare, poate avea memorie locală și este descris de o funcție de transfer. Aranjarea în paralel a neuronilor formează un *strat* și corespunde nodurilor din graful orientat care aparțin aceluiași nivel. O rețea neuronală poate conține unul, două sau mai multe straturi. *Topologia* unei rețele neuronale este dată de numărul de straturi ale rețelei, numărul neuronilor dintr-un strat, tipul neuronilor (al funcțiilor lor de transfer) și tipul conexiunilor din rețea.

Proprietatea rețelelor neuronale de a învăța se reflectă în capacitatea acestora de

a-și ajusta parametrii pe baza datelor de intrare folosite în faza de antrenare (învățare). Această proprietate le face deosebit de atractive și ea se realizează prin intermediul *legii de antrenare (învățare)*. Aceasta trebuie să exprime totalitatea modificărilor efectuate la nivelul rețelei, care să inducă acesteia comportamentul dorit.

Antrenarea rețelei neuronale, pentru ca aceasta să realizeze o anumită funcțiune dorită, poate fi [Haykin, 1994]:

- *antrenare supervizată*: regulile de învățare din cadrul acestei categorii sunt de tipul procedurilor de estimare de parametri și își propun să găsească valorile parametrilor rețelei neuronale care să minimizeze o funcție criteriu; de regulă, funcția criteriu utilizată este eroarea medie pătratică dintre ieșirea rețelei și ieșirea dorită corespunzătoare unui vector de intrare cunoscut; vectorul de intrare cunoscut și vectorul de ieșire dorit, corespunzător intrării cunoscute, formează *perechea de antrenare*;
- *antrenarea nesupervizată*: reprezintă un model al procesului de învățare similar celui realizat de ființele umane; pentru acest tip de învățare, setul de antrenare nu conține decât vectorul de intrare, nefiind necesar un vector obiectiv de ieșire;

În domeniul diagnozei anomaliilor proceselor, rețelele neuronale artificiale (limba engleză: “Artificial Neural Networks” - ANN) au fost aplicate în etapa de generare de reziduuri (simptome) fiind folosit potențialul acestora de a aproxima o funcție, precum și în etapa evaluării reziduurilor (simptomelor) fază în care, în esență, trebuie rezolvată o problemă de clasificare, rețeaua fiind antrenată și folosită în acest scop.

Sistemele expert pentru diagnoza anomaliilor bazate pe teoria rețelelor neuronale artificiale, le folosesc pe acestea în scopul reprezentării cunoștințelor din proces, cât și în scopul realizării inferențelor. În cazul diagnozei anomaliilor proceselor, aceasta înseamnă, de fapt, procesul de luare a deciziei bazat pe recunoașterea de forme.

2.2 Rețele neuronale artificiale aplicate în etapa generării simptomelor

Rețelele neuronale artificiale sunt utile în mod deosebit pentru rezolvarea problemelor de identificare (modelare) a proceselor dinamice neliniare datorită capacității lor de aproximare universală. Mai mult decât atât, pentru o modelare neuronală cu performanțe bune, nu sunt necesare cunoștințe amănunțite despre structura sistemului dinamic supus analizei. Datorită faptului că nu există o interpretare fizică a parametrilor rețelei neuronale și, teoretic, este necesară o infinitate de parametri pentru a modela exact un proces, rețelele neuronale artificiale folosite la identificarea proceselor trebuie considerate drept *modele neparametrice* ale acestora [Isermann, *et al.*, 1997].

Rețelele neuronale artificiale sunt folosite pentru modelarea, în domeniul discret, a proceselor, ele lucrând cu secvențe de eșantioane ale mărimilor de intrare - ieșire din proces, cel mai adesea fiind folosită reprezentarea intrare-ieșire. Considerând un sistem cu mai multe intrări și o singură ieșire (limba engleză: “Multi-Input Single-Output” – MISO) de ordin (n_u, n_y) , ecuația cu diferențe ce caracterizează în domeniul discret comportarea intrare-ieșire a sistemului poate fi scrisă sub forma [Frank and Köppen-Seliger, 1997]:

$$y(k) = g(y(k-1), \dots, y(k-n_y), \underline{u}(k), \dots, \underline{u}(k-n_u)), \quad (\text{I. 15})$$

unde y este mărimea de ieșire a procesului, \underline{u} este vectorul mărimilor de intrare din proces, iar k este momentul normalizat de eșantionare. Pentru aproximarea ieșirii un astfel de sistem, descris în domeniul discret de o ecuație de forma (I. 15), este necesară o singură rețea neuronală artificială. În figura I. 12 este exemplificat cazul identificării unui sistem dinamic cu o singură intrare utilizând ANN.

Pentru modelarea neuronală a sistemelor cu mai multe intrări și mai multe ieșiri (limba engleză: “Multi-Input Multi-Output” – MIMO) uzual se procedează la o descompunere a acestora în subsisteme de tip MISO, pentru fiecare subsistem identificându-se un model neuronal. Aceasta se realizează deoarece modelarea

sistemelor MIMO cu ajutorul unor ANN cu mai multe ieșiri (fiecare ieșire a ANN modelând una din ieșirile procesului) conduce la obținerea unor modele neuronale cu structură complicată, iar calitatea aproximării este diminuată.

Pentru antrenarea rețelelor neuronale în scopul identificării sistemelor dinamice neliniare sunt necesare doar măsuratori precisi și relevante ale mărimilor de intrare - ieșire din proces și cunoașterea apriorică a ordinului sistemului.

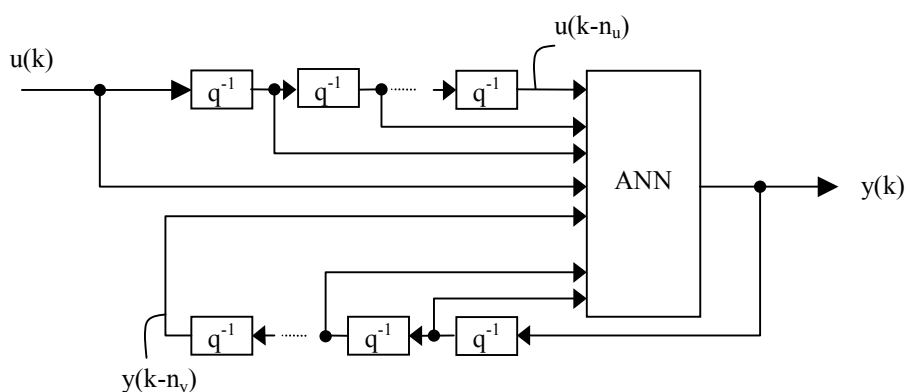


Fig. I. 12 Rețea neuronală artificială folosită în identificare

În scopul generării de reziduuri (simptome) care să poarte informații despre comportarea sistemului supus analizei și care să fie folosite pentru o detecție și localizare corectă a anomaliilor în cazul în care acestea apar, este necesară elaborarea unor modele neuronale atât pentru comportarea normală a procesului, cât și pentru comportări anormale cunoscute. Pentru aceasta sunt necesare măsuratori ale mărimilor de intrare - ieșire din procesul real, dacă sunt disponibile, sau date colectate prin simularea cât mai exactă a procesului supus atenției. Cum în general, nu se poate dispune de măsurători din comportări anormale ale procesului, pentru colectarea de date din astfel de comportări se apelează frecvent la simulare.

După ce rețelele neuronale artificiale au fost antrenate ca să aproximeze ieșirile procesului analizat, în comportarea normală și comportări anormale cunoscute, acestea pot fi incluse în scheme de generare on-line a reziduurilor, așa cum este prezentat în figura I. 13 [Frank and Köppen-Seliger, 1997]. În acest fel, se generează un set de reziduuri ce permite diagnoza unică a anomaliilor care apar în funcționarea curentă a procesului.

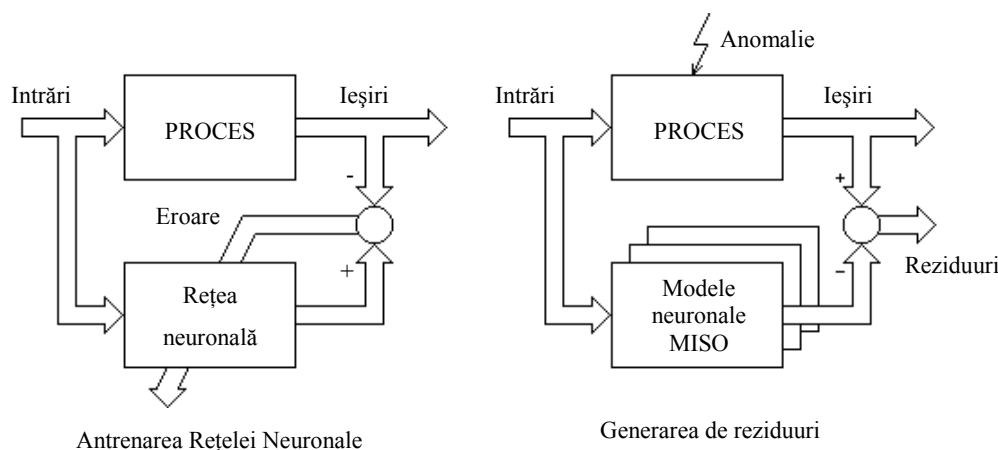


Fig. I. 13 Generarea on-line a reziduurilor

Rețelele neuronale descrise permit aproximarea unor funcții multidimensionale neliniare statice. Pentru ca aceste rețele neuronale artificiale să poată fi folosite în scopul identificării sistemelor dinamice neliniare, ele trebuie dotate în mod adecvat cu elemente dinamice, existând două modalități și anume:

- a) utilizarea unor elemente dinamice externe sub forma unor cascade de filtre conectate la intrările - ieșirile rețelei, rezultând așa-numitele *rețele neuronale artificiale cu structură dinamică externă*;
- b) includerea în structura neuronilor a unor elemente dinamice sub forma unor filtre și a unor conexiuni recurente, rezultând așa-numitele *rețele neuronale artificiale cu structură dinamică internă*.

a) Rețele neuronale artificiale cu structură dinamică externă

Abordările bazate pe folosirea elementelor dinamice externe, utilizează rețele neuronale artificiale statice în scopul identificării sistemelor dinamice neliniare [Narendra and Parthasarathy, 1990; Narendra, 1992].

În cel mai general caz, un model neliniar intrare - ieșire descrie ieșirea estimată, \hat{y} ca o funcție de intrările și ieșirile filtrate. Considerând q^{-1} drept operatorul de întârziere cu un tact, atunci se poate scrie:

$$\hat{y}(k) = f(F_0(q^{-1}) \cdot u(k-d), \dots, F_{n_u}(q^{-1}) \cdot u(k-d), G_1(q^{-1}) \cdot y(k), \dots, G_{n_y}(q^{-1}) \cdot y(k)) \quad (I. 15)$$

în care n_u și n_y sunt ordinele procesului dinamic, iar d reprezintă timpul mort. Acești parametri trebuie cunoscuți a-priori. Funcția necunoscută $f(\cdot)$ poate fi aproximată printr-o rețea neuronală statică, $F_i(q^{-1})$ și $G_i(q^{-1})$ fiind filtre liniare care furnizează rețelei neuronale statice informații despre valorile anterioare ale intrărilor și ieșirilor rețelei, așa cum este ilustrat în figura I. 14 .

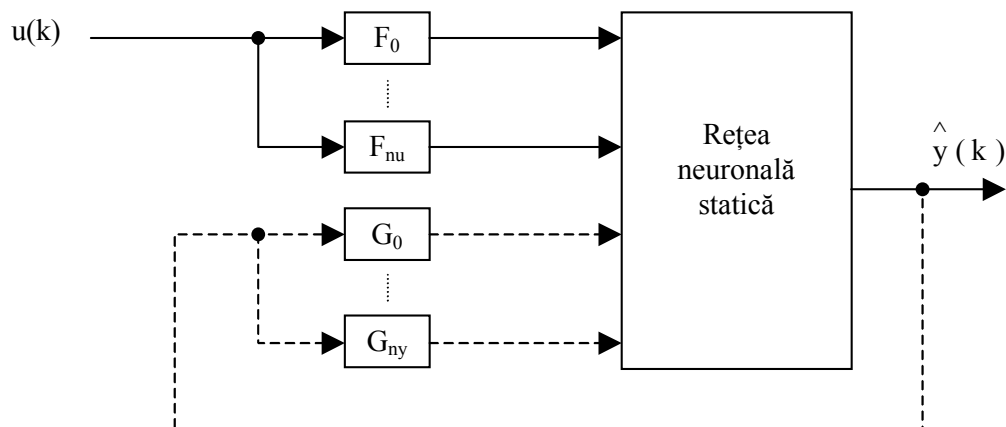


Fig. I. 14 Structura generală a unei RNA cu structură dinamică externă

În funcție de modul de alegere al filtrelor liniare, pot fi evidențiate următoarele abordări [Isermann, *et al.*, 1997]:

a.1) rețele neuronale artificiale cu reacție după ieșirea rețelei: apare reacția după ieșirea filtrată a rețelei; filtrele de regulă sunt elemente simple de întârziere:

$$F_i(q^{-1}) = q^{-i} \quad , \quad G_i(q^{-1}) = q^{-i} \quad (I. 16)$$

relația (I. 15) devenind:

$$\hat{y}(k) = f(u(k-d), \dots, u(k-d-n_u), \hat{y}(k-1), \dots, \hat{y}(k-n_y)),$$

Reacția după ieșire determină în general, pentru astfel de modele neuronale, ca stabilitatea ANN să nu poată fi dovedită cu ușurință. De regulă, în practică, se realizează o reacție după ieșirea măsurată și filtrată din procesul real în locul reacției după ieșirea rețelei.

a.2) rețele neuronale artificiale cu răspuns finit la impuls: această abordare nu implică reacție după ieșire, filtrele care apar în relația generală (I. 15) fiind de forma:

$$F_i(q^{-1}) = q^{-i} \quad , \quad G_i(q^{-1}) = 0. \quad (I. 17)$$

În felul acesta, ieșirea unui astfel de model neuronal este dată de relația:

$$\hat{y}(k) = f(u(k-d), \dots, u(k-d-n_u)). \quad (\text{I. 18})$$

Dezavantajul acestei abordări este acela că n_u trebuie să fie foarte mare, teoretic el trebuind să tindă la infinit, pentru o modelare exactă a procesului, lucru care determină o creștere a dimensiunii spațiului de intrare. Practic el este ales ca fiind $10 \div 40$ perioade de eșantionare. Deoarece acest tip de modele nu implică reacție după ieșire, stabilitatea lor este garantată.

a.3) rețele neuronale cu funcții de bază ortonormale: acest tip de rețele încorporează, prin intermediul filtrelor, informații apriorice despre sistemul supus atenției, informații legate de existența și tipul polilor dominanți; cu cât aceste informații sunt mai precise, cu atât ordinul n_u necesar unei modelări adecvate scade mai mult.

b) Rețele neuronale artificiale cu structură dinamică internă

Rețelele neuronale artificiale cu elemente dinamice interne se obțin dotând rețelele neuronale statice cu memorie, deci modificând structura internă a neuronilor, așa cum este prezentat în figura I. 15 [Isermann et al., 1997]. Spre deosebire de rețelele neuronale cu dinamică externă, la intrarea rețelelor cu dinamică internă nu mai este necesară furnizarea de valori anterioare ale intrărilor și ieșirilor din proces, în felul acesta reducându-se dimensiunea spațiului de intrare.

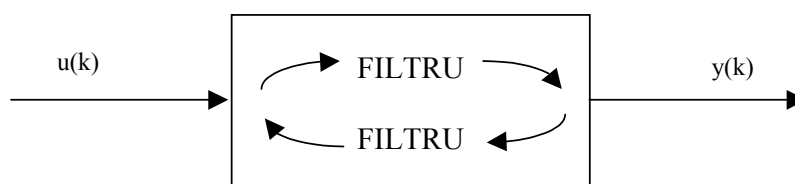


Fig. I. 15 Rețea neuronală cu structură dinamică internă

Rețelele neuronale artificiale cu structură dinamică internă au următoarea descriere generală în spațiul stărilor:

$$\begin{aligned} \underline{x}(k+1) &= f(\underline{x}(k), \underline{u}(k)), \\ y(k) &= g(\underline{x}(k), \underline{u}(k)). \end{aligned} \quad (\text{I. 19})$$

În această categorie sunt incluse trei tipuri de rețele, în funcție de modul în care

este realizată intern (la nivelul neuronilor) recurența:

b.1) rețele cu recurență totală: au în structura lor neuroni complet interconectați, fiecare neuron reprezentând o stare internă a rețelei, așa cum se poate observa și în figura I. 16 [Williams and Zipser, 1990]; dezavantajul acestei configurații este convergența lentă a procedurii de antrenare și problemele care apar legate de stabilitatea sa.

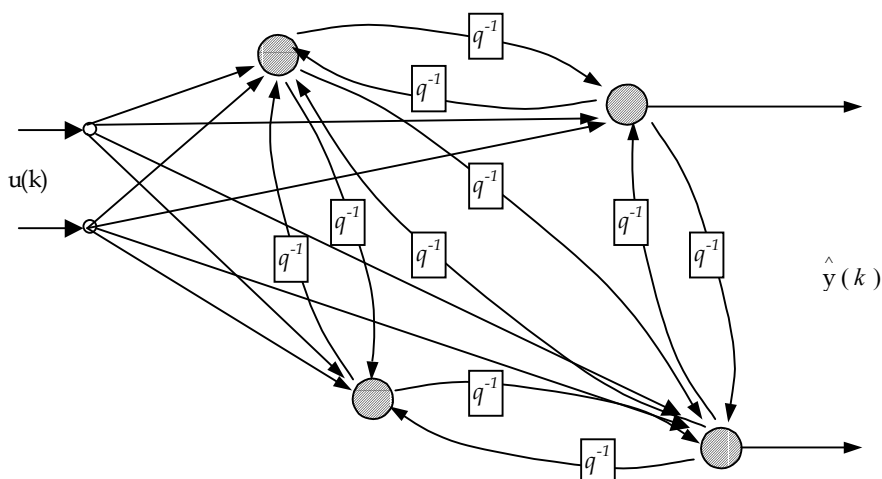


Fig. I. 16 Rețea neuronală cu recurență totală

b.2) rețele neuronale parțial recurente: au la bază structura generală a rețelelor neuronale artificiale, dar conțin suplimentar un strat de neuroni de context (figura I. 17); neuronii stratului de context au rolul de memorie internă a modelului neuronal.

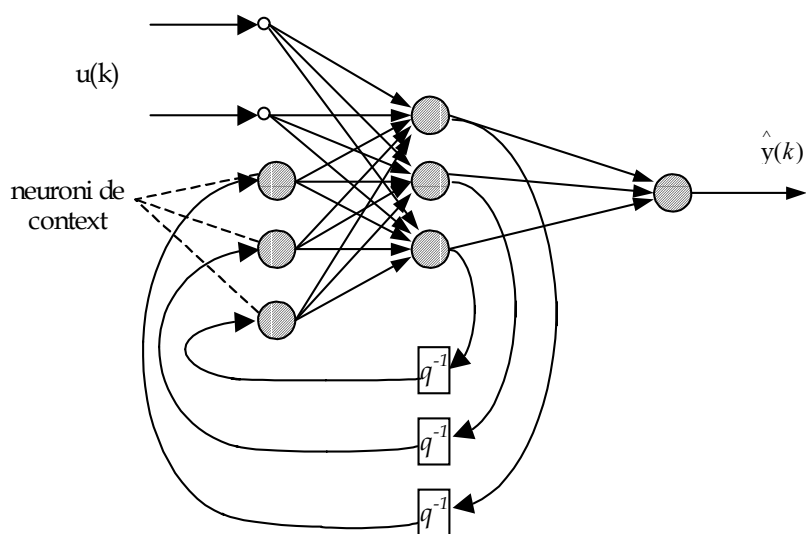


Fig. I. 17 Rețea neuronală parțial recurentă

b.3) rețele local recurente, cu transmitere globală înainte a intrării: au la bază structura rețelelor neuronale statice, în care apar elemente, sub forma unor filtre, ce asigură o recurență locală a structurii [Tsoi and Back, 1994]. Astfel de rețele nu conțin conexiuni de reacție între neuronii din straturi succesive sau conexiuni laterale între neuronii aceluiași strat, recurența fiind restricționată doar la legăturile (sinapsele) dintre neuroni sau la nivelul structurii interne a neuronului.

Adaptarea parametrilor unei rețele neuronale cu dinamică internă se face prin intermediul unor proceduri de optimizare neliniară, cel mai adesea fiind folosiți algoritmi de învățare de tip gradient care depind de stări anterioare ale rețelei neuronale, cum ar fi [Haykin, 1994; Pearlmutter, 1995]:

- algoritmul propagării temporale înapoi a erorii;
- algoritmul învățării recurente în timp real.

Acești algoritmi diferă între ei prin modul în care este prelucrată informația privind stările anterioare ale rețelei neuronale.

2.3 Rețele neuronale artificiale aplicate în faza de evaluare a simptomelor

Faza de evaluare a reziduurilor (simptomelor) presupune de fapt rezolvarea unei probleme de clasificare a formelor (reziduurilor) obținute prin folosirea unor modele asociate procesului analizat. Rețelele neuronale artificiale au fost aplicate cu succes în rezolvarea problemelor de clasificare ce le implică analiza reziduurilor, în scopul unei diagnoze corecte a anomaliilor, proprietățile acestora care le-au determinat utilitatea fiind [Pao, 1989; Marcu, 1995]:

- folosirea în cazul proceselor neliniare și cu incertitudini asupra modelului, ANN fiind antrenate numai pe baza datelor de intrare - ieșire, fără a folosi alte informații legate de procesul supus atenției;
- toleranța la zgomot;
- capacitatea de generalizare;
- capacitatea de a se adapta pe parcursul folosirii.

Rețeaua neuronală trebuie să joace rolul unui clasificator care să indice corect cărei clase aparțin formele prezentate la intrarea sa.

Înainte ca rețeaua să fie folosită pentru evaluarea on-line a simptomelor, ea trebuie antrenată pentru acest scop. Antrenarea rețelei neuronale este de regulă supervizată, acestea fiindu-i aplicate la intrare simptomele obținute pentru fiecare tip de comportare în parte, normală și anormale cunoscute, ieșirile acesteia trebuind să reproducă vectori care să pună în evidență tipul de comportare corespunzător. În general se folosesc rețele neuronale având atâtea intrări câte simptome sunt generate și atâtea ieșiri câte clase de comportare se dorește a fi recunoscute. Așadar, pentru faza de antrenare a clasificatorului neuronal, trebuie să se dispună de o bază de date conținând simptomele (semnături ale anomaliei) [Frank and Köppen-Seliger, 1997]. Figura I. 18 ilustrează modul în care este folosită o rețea neuronală în scopul evaluării reziduurilor.

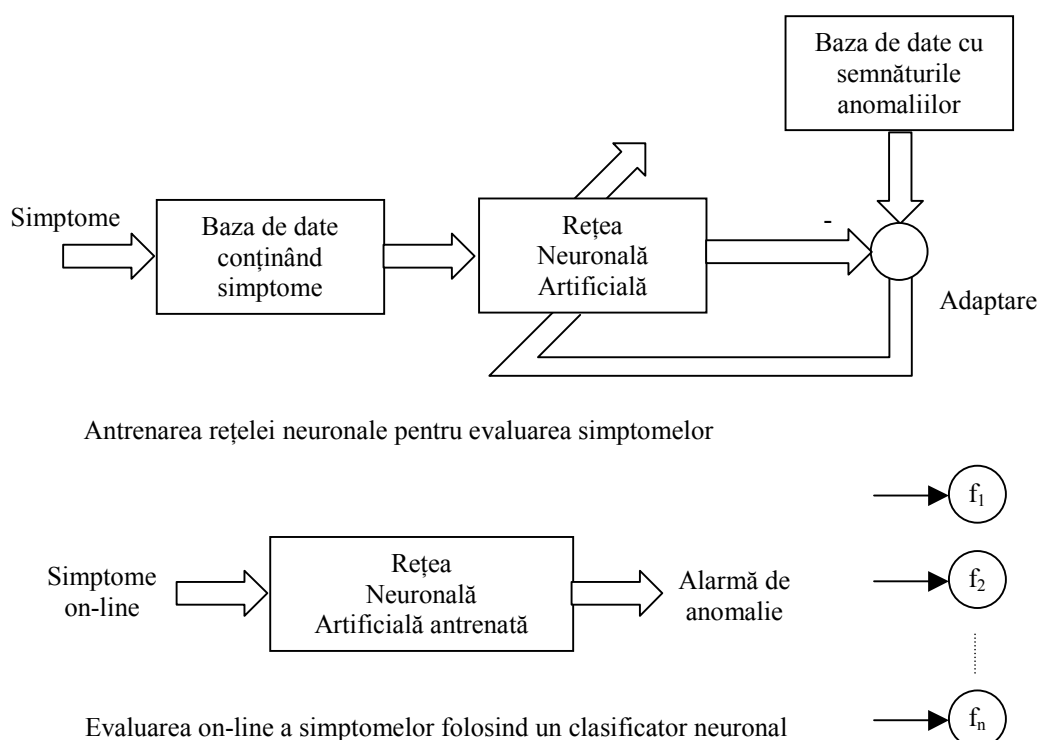


Fig. I. 18 Evaluarea (on-line a) simptomelor folosind un clasificator neuronal

Atât în procesul de antrenare și testare a clasificatorului neuronal (prin clasificarea setului de antrenare), precum și în procesul de generalizare a

cunoștințelor dobândite (clasificare a unor forme ce nu aparțin setului de antrenare), o importanță deosebită o are conceptul de *respingere* care se referă la faptul că sistemul de evaluare (clasificare) nu trebuie să clasifice o formă aflată într-o regiune de dubiu [Dubuisson, 1992]. Există două tipuri de respingere:

- *respingere bazată pe ambiguitate*: se aplică formelor ce pot aparține la mai mult de o clasă;
- *respingere bazată pe distanță*: se aplică formelor ce sunt departe de toate clasele, situație care, din perspectiva diagnozei anomaliilor proceselor, conduce la considerarea unei (unor) noi clase, necunoscută (necunoscute) la începutul antrenării clasificatorului.

Abilitatea de clasificare adecvată depinde de modul în care clasificatorul neuronal realizează discriminarea dintre regiunile spațiului de intrare (trăsături) corespunzătoare diferitelor clase de comportare ale procesului. În general, delimitarea regiunilor de decizie depinde de:

- alegerea datelor de intrare - ieșire care trebuie să fie relevante;
- arhitectura (topologia) rețelei neuronale folosite;
- extinderea perioadei de antrenare .

În ultimii ani, un mare interes l-a produs integrarea tehnicilor fuzzy și a rețelelor neuronale în scopul obținerii unor clasificatori capabili să prelucreze informații vagi. Antrenarea unor astfel de rețele neuro-fuzzy se face pornind de la algoritmi clasici de tip gradient, în care sunt incluse funcțiile de apartenență ale mărimilor fuzzy cu care se lucrează [Marcu, 1995].

Capitolul 3

Tendințe actuale în diagnoza bazată pe utilizarea tehnicilor fuzzy

3.1 Concepte de bază ale teoriei mulțimilor vagi

Noțiunea de *mulțime fuzzy* a fost introdusă ca o generalizare a conceptului de apartenență binară a unui element la o mulțime. Se apelează la descrieri de tip fuzzy când frontiera unei mulțimi nu este precis definită. Mulțimea fuzzy este o mulțime căreia i se asociază o funcție caracteristică ce ia valori în intervalul $[0,1]$, valorile acesteia descriind gradul de apartenență al unui element la acea mulțime [Jang, *et al.*, 1997].

Există două moduri de definire a unei mulțimi fuzzy și anume:

- prin valori ale funcției caracteristice atașate pentru orice element al mulțimii; se apelează la acest mod de descriere când mulțimea în cauză are un număr finit de elemente;
- prin expresia analitică a funcției caracteristice atașate; se apelează la acest mod de descriere când mulțimea în cauză are o infinitate de elemente.

Cel mai des folosite tipuri de funcții caracteristice sunt următoarele:

a) pentru o formulare de tipul:

“mărimea A are o valoare mare”

se poate asocia o mulțime fuzzy descrisă de o funcție caracteristică monoton crescătoare de forma:

$$\mu_{a,b}(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases} \quad (\text{I. 1})$$

parametrii a și b definind zonele în care mărimea A nu poate fi considerată ca având valoare mare, respectiv este sigur mare;

b) pentru formularea:

“mărimea A este aproximativ egală cu m ”

se poate folosi o mulțime fuzzy descrisă de o funcție caracteristică de formă trapezoidală sau triunghiulară dată de:

$$\mu_{m,d}(x) = \begin{cases} 1 - \frac{|x-m|}{d}, & m-d \leq x \leq m+d \\ 0, & x < m-d \text{ sau } x > m+d \end{cases}, \quad d > 0, m \in \mathbb{R}. \quad (\text{I. 2})$$

Se poate folosi și o mulțime fuzzy a cărei funcție caracteristică este de tip Gauss:

$$\mu_{a,m}(x) = e^{-(a) \cdot (x-m)^2}, \quad a > 0, m \in \mathbb{R} \quad (\text{I. 3})$$

c) pentru o formulare de tipul:

“mărimea A este cuprinsă cu aproximație între b și c ”

se poate folosi o mulțime fuzzy cu o funcție caracteristică trapezoidală de forma:

$$\mu_{a,b,c,d}(x) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b < x < c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x < a \text{ sau } x > d \end{cases}, \quad a < b < c < d. \quad (\text{I. 4})$$

Un alt concept de bază al teoriei mulțimilor vagi îl reprezintă *relația fuzzy*. Aceasta este o generalizare a noțiunii de relație definită pe mulțimi obișnuite. O relație între n mulțimi fuzzy definite pe universurile X_1, \dots, X_n este de fapt o mulțime fuzzy definită pe universul $X_1 \times \dots \times X_n$.

Relațiile fuzzy pot fi compuse în mod asemănător compunerii relațiilor din teoria clasică a mulțimilor. Astfel, dacă R este o relație între mulțimile fuzzy X_1 și X_2 , iar S este o relație între mulțimile fuzzy X_2 și X_3 , atunci compunând relațiile R și S se obține o relație între mulțimile fuzzy X_1 și X_3 , notată RoS și definită prin următoarea funcție de apartenență:

$$\mu_{RoS}(x_1, x_3) = \sup_{x_2 \in X_2} \min(\mu_R(x_1, x_2), \mu_S(x_2, x_3)),$$

$$(x_1, x_3) \in X_1 \times X_3 \quad (I. 5)$$

Relațiile fuzzy împreună cu operația de compunere a acestora, așa cum a fost ea definită mai sus, stau la baza *raționamentelor aproximative de tip fuzzy*. În urma acestor raționamente se obțin concluzii cu un anumit grad de incertitudine, pornind de la o colecție de premise ce au un anumit grad de imprecizie.

De exemplu, o regulă de forma [Shen and Leitch, 1993]:

$$\text{DACĂ } x \text{ este } A_i, \text{ ATUNCI DACĂ } y \text{ este } B_i \text{ ATUNCI } z \text{ este } C_i \quad (I. 6)$$

care guvernează o relație dintre premisele A_i , B_i și concluzia C_i , poate fi transpusă sub forma unei relații fuzzy de tipul:

$$\mu_{R_i}(x, y, z) = \min(\mu_{A_i}(x), \mu_{B_i}(y), \mu_{C_i}(z)), \quad (I. 7)$$

în care A_i este o mulțime fuzzy definită pe universul de discurs al variabilei x , iar B_i și C_i sunt de asemenea mulțimi fuzzy, dar definite pe universurile de discurs ale variabilelor y și, respectiv, z .

Când este disponibil un set de astfel de reguli, se poate obține o relație R compunând relațiile fuzzy asociate regulilor:

$$\mu_R(x, y, z) = \max_{i \in \{1, \dots, n\}} \min(\mu_{A_i}(x), \mu_{B_i}(y), \mu_{C_i}(z)). \quad (I. 8)$$

Legătura dintre mulțimile fuzzy și regulile dintr-un sistem expert se face prin intermediul *variabilelor lingvistice*, astfel obținându-se o cuantificare a unor afirmații vagi. O variabilă lingvistică poate avea drept valoare un *termen lingvistic*. Acesta poate fi modelat prin mulțimi fuzzy sau operații cu mulțimi fuzzy, căpătând astfel evaluări cantitative. Variabilele lingvistice ajută la

scrierea de reguli, ele putând apare fie în partea de condiție, fie în partea de concluzie, fie în ambele. Pentru o problemă în care se folosesc variabile lingvistice, trebuie să se stabilească mulțimea termenilor lingvistici asociați fiecărei variabile.

Sistemele expert de tip fuzzy sunt capabile să rezolve probleme în care apar elemente de incertitudine, raționamentele dezvoltate de acestea având la bază *logica fuzzy*. Spre deosebire de logica clasică în cadrul căreia sunt folosite doar două valori de adevăr, logica fuzzy permite ca toate predicatul să admită valori de adevăr în intervalul $[0,1]$, oferind astfel o reprezentare mai apropiată de lumea reală.

Orice operație logică ($\&$, SAU, \Rightarrow , \Leftrightarrow) poate fi definită, în logica fuzzy, ca o aplicație de la $[0,1] \times [0,1] \rightarrow [0,1]$. Aceasta determină valoarea de adevăr a unei formule compuse în funcție de valorile de adevăr ale subformulelor legate prin operația logică respectivă, nemaifiind posibilă o definiție sub forma tabelului de adevăr așa cum se procedează în logica clasică.

Pentru a realiza un sistem expert fuzzy este necesară transpunerea regulii “modus ponens” care stă la baza realizării raționamentelor într-un sistem expert bazat pe logica clasică și care se aplică astfel:

Fapt: $x = a$

Regula: DACĂ $x = a$ ATUNCI $y = b$.

Într-un sistem expert de tip fuzzy, faptele vor fi de forma:

Fapt: $x \approx a$

fiind modelate prin mulțimi fuzzy (cu o funcție caracteristică asociată), iar regulile vor fi de forma:

Regula: DACĂ $x \approx a$ ATUNCI $y \approx b$.

În această situație, atât premiza ce activează regula cât și concluzia au un grad de incertitudine modelat prin mulțimi fuzzy. În cazul cel mai general, pot exista

mai multe reguli care să se refere la aceeași concluzie, părțile de condiție ale acestor reguli fiind diferite, ca în exemplul următor:

DACĂ $(E_1^1 \text{ \textasciitxt{ȘI} } E_2^1 \text{ \textasciitxt{ȘI} } \dots \text{ \textasciitxt{ȘI} } E_{i_1}^1)$ ATUNCI C ;

.....
 DACĂ $(E_1^n \text{ \textasciitxt{ȘI} } E_2^n \text{ \textasciitxt{ȘI} } \dots \text{ \textasciitxt{ȘI} } E_{i_n}^n)$ ATUNCI C .

După cum se poate observa, în exemplul prezentat, partea de condiție pentru fiecare regulă este de formă conjunctivă și, dacă fiecare fapt E_i^j are caracter imprecis, asociindu-se acestuia o mulțime fuzzy caracterizată de o anumită funcție de apartenență, atunci întregii părți de condiție a regulii i se asociază valoarea minimă a funcțiilor caracteristice atașate, după cum este definită operația de conjuncție în logica fuzzy.

Concluzia C este obținută prin partea de acțiune a mai multor reguli, ceea ce ar corespunde unei disjuncții. În acest caz, concluzia obținută va fi introdusă în baza de fapte asociindu-se ei un coeficient dat de valoarea maximă a coeficienților atașați părților de condiție ale fiecărei reguli, după cum a fost definită disjuncția în logica fuzzy, rezultând:

$$\mu(C) = \max(\min(\mu(E_1^1), \dots, \mu(E_{i_1}^1)), \dots, \min(\mu(E_1^n), \dots, \mu(E_{i_n}^n))) \quad (I. 9)$$

Se spune ca regula “modus ponens” din logica clasică are drept corespondent în logica fuzzy regula “max-min”.

Un Sistem bazat pe Logica Fuzzy (limba engleză: “Fuzzy Logic System” - FLS) are principala proprietate de a fi capabil să prelucreze simultan date numerice și cunoștințe euristice. Aplicarea acestui cadru unic pentru prelucrarea atât a informației numerice, cât și a cunoștințelor exprimate lingvistic, în scopul proiectării sistemelor de diagnoză a anomaliilor proceselor bazate pe model, constă în principal din:

- identificarea fuzzy a sistemelor în scopul generării simptomelor;
- efectuarea unor raționamente fuzzy în scopul evaluării simptomelor.

Structura generică a unui FLS este descrisă în figura I. 19 [Babuska and

Verbruggen, 1996; Isermann, 1998]. Acesta conține trei elemente: *fuzzificator*, *motor de inferențe de tip fuzzy* și un *defuzzificator*. Elementul de bază al FLS este motorul de inferențe de tip fuzzy care, la rândul său, este compus din baza de reguli și baza de date care formează, împreună, baza de cunoștințe a FLS, și motorul de inferențe propriu-zis. Fuzzificatorul și defuzzificatorul constituie interfețele FLS cu datele numerice cu care acesta lucrează, transformând numerele precise (exacte) în mulțimi fuzzy și invers. Această operație, de fuzzificare, este necesară activării regulilor care sunt exprimate folosind variabile lingvistice cărora li se asociază mulțimi fuzzy. Operația de defuzzificare este necesară atunci când FLS trebuie să furnizeze la ieșire valori precise.

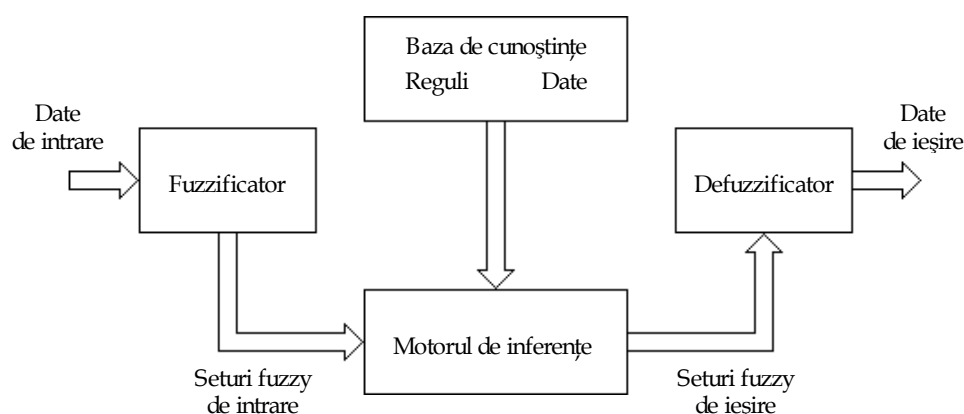


Fig. I. 19 Sistem expert bazat pe logica fuzzy

Incertitudinile asupra variabilelor și parametrilor sistemului sunt exprimate prin funcții de apartenență. Regulile care reflectă constrângeri de cauzalitate sunt exprimate sub forma unei colecții de instrucțiuni de tip DACĂ – ATUNCI. Acestea pot fi furnizate de către experți umani sau pot fi extrase din datele numerice obținute prin măsurători, ele descriind funcționarea sistemului de modelat.

Simptomele pot fi generate pe baza estimării ieșirilor procesului folosind modele calitative (observări bazați pe cunoștințe). Ieșirile estimate sunt comparate cu măsurătorile existente obținându-se astfel așa-numitele *reziduuri*. În cazul ideal, al absenței perturbațiilor și zgomotelor (intrări necunoscute în proces), reziduurile trebuie să fie nule în cazul unei comportări normale a procesului și nenule când apare o anomalie în sistem.

3.2 Tehnici fuzzy folosite în etapa generării simptomelor

Abordarea fuzzy pentru modelarea sistemelor dinamice neliniare poate fi interpretată ca un tip special de interpolare neliniară și anume o *interpolare bazată pe cunoștințe*. Relațiile dintre variabilele de intrare și cele de ieșire ale procesului supus atenției pot fi descrise în trei moduri diferite folosind tehnici fuzzy și anume [Frank and Marcu, 1999]:

- *modele calitative fuzzy*: folosesc o bază de reguli;
- *modele relaționale fuzzy*: folosesc un set de parametri determinați într-o fază de identificare bazată pe un set de date de antrenare;
- *modele funcționale fuzzy*: apelează la o serie de modele locale pentru a descrie comportarea sistemului în diverse puncte de operare.

În scopul proiectării unui sistem de diagnoză a anomaliilor, este necesar un model reprezentativ al procesului, ceea ce înseamnă ca este necesar un model capabil să evidențieze anomaliile de interes și care, din perspectiva asigurării robusteții schemei de diagnoză, nu este sau este slab afectat de perturbații și incertitudini (erori) de modelare. În felul acesta, modelul necesar diagnozei anomaliilor poate fi mai simplu decât cel folosit în control.

a) *Observeri calitativi fuzzy*

Metoda de modelare calitativă fuzzy poate fi privită ca o extindere a modelării matematice (analitice) convenționale. Ea se bazează pe două categorii de cunoștințe despre proces și anume:

- descrierea structurală a sistemului;
- cunoștințe despre comportarea fiecărei părți componente, aici fiind incluse legile fizice deduse din evoluții anterioare ale procesului.

În felul acesta, se construiește un model bazat pe două categorii de elemente [Kuipers, 1986]:

- *variabile calitative*: reprezintă parametrii fizici ai sistemului, fiind funcții de timp;

- *constrângeri*: definesc modul în care parametrii fizici ai sistemului se raportează unul la celălalt.

Variabilele calitative sunt definite într-un spațiu cantitativ reprezentat printr-o mulțime finită de valori limită ordonate, care reprezintă puncte distincte pe axa reală. În felul acesta se obține o secvență alternantă de puncte și intervale deschise. La orice moment de timp, o variabilă calitativă este descrisă de o pereche de valori calitative: amplitudinea calitativă și derivata, anume direcția și viteza de variație a variabilei.

Abordarea bazată pe spațiul cantitativ fuzzy pornește prin a considera un număr oarecare, dar finit, de valori calitative exprimate sub forma unor mulțimi fuzzy, cu funcții de apartenență, de exemplu de tip trapezoidal, ca în figura I. 20 . În felul acesta se realizează o descriere mai detaliată a variabilelor din proces și a relațiilor dintre ele. Variabilele sistemului dinamic și derivatele lor sunt reprezentate prin mulțimi fuzzy. Informațiile legate de dimensiune și semn sunt reprezentate prin relații fuzzy între două sau mai multe variabile.

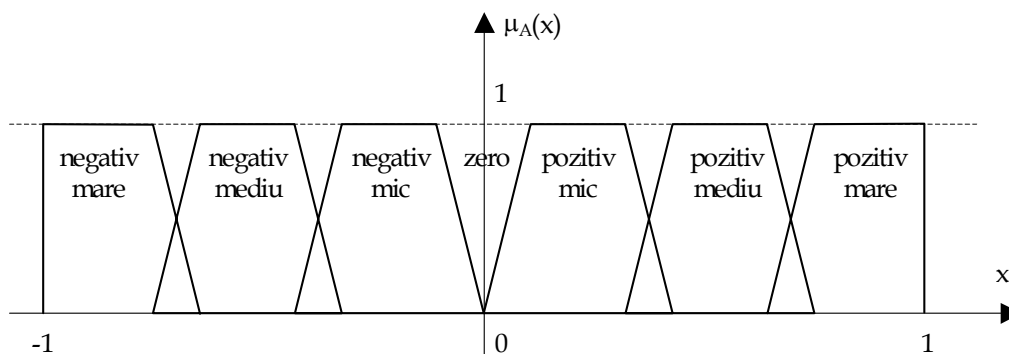


Fig. I. 20 Spațiul cantitativ fuzzy

Mulțimile valorilor posibile pe care variabilele din sistem le pot lua sunt restricționate fie prin ecuații relaționale algebrice, derivate sau funcționale. Mai exact, operațiile algebrice efectuate în spațiul cantitativ fuzzy sunt de fapt operațiile algebrice definite cu numere fuzzy (numere reprezentate cu ajutorul mulțimilor fuzzy). O relație derivativă reflectă faptul că valoarea calitativă fuzzy asociată amplitudinii unei variabile este egală cu valoarea calitativă fuzzy asociată vitezei de variație a altei variabile. Relațiile funcționale sunt

reprezentate prin relații fuzzy, în felul acesta putându-se exploata informațiile imprecise și (sau) parțial numerice despre relațiile funcționale de dependență între variabilele sistemului.

Durata rămânerii într-o anumită stare, în care variabilele sistemului au aceleași valori calitative, cât și momentul apariției unei tranziții de stare sunt, de asemenea, reprezentate prin mulțimi fuzzy în spațiul fuzzy al cantităților, în felul acesta obținându-se o ordonare a evoluției stărilor și a duratelor de timp asociate lor.

Descrierea evoluției în timp a variabilelor sistemului este integrată în model sub forma unor reguli de tip DACĂ – ATUNCI sau sub forma unor ecuații diferențiale calitative. Acestea din urmă pot fi privite ca fiind o generalizare a ecuațiilor diferențiale convenționale. Regulile descriu lingvistic relația dintre variabilele calitative.

Pentru a deduce comportarea sistemului în ansamblul său pe baza informațiilor legate de structura sa și de comportarea părților sale componente, se aplică o simulare calitativă pornind de la starea inițială a sistemului și folosind ecuațiile diferențiale calitative. Starea calitativă a sistemului este definită ca o secvență de valori calitative asociate variabilelor din proces. În felul acesta, comportarea calitativă a sistemului este definită ca o succesiune de stări, fiind descrisă printr-un graf orientat ce evidențiază posibilele stări viitoare ale sistemului, cum este cel prezentat în figura I. 21 .

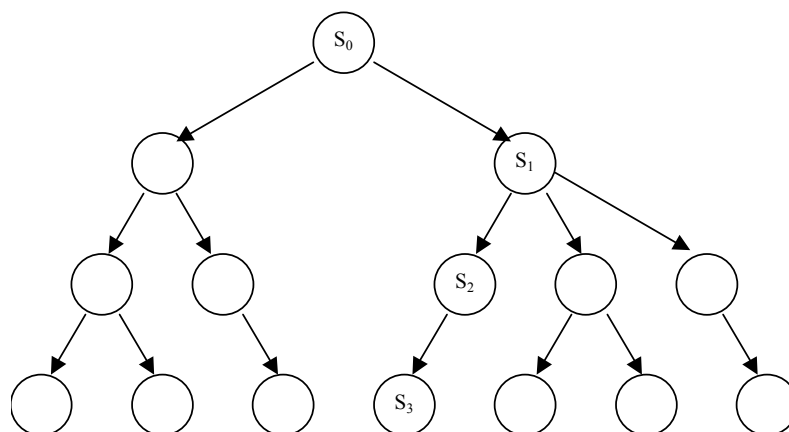


Fig. I. 21 Descrierea grafică a stărilor procesului

Prin simulare calitativă se obțin toate tranzițiile posibile exprimate în valori calitative permise pentru fiecare parametru, urmând a fi verificată consistența acestor tranziții. Tranzițiile inconsistente (imposibil de realizat) sunt eliminate din graf, acest proces fiind cunoscut sub numele de *filtrare calitativă*. Există mai multe tehnici de realizare a acestui proces și anume: filtrare bazată pe constrângeri, filtrare temporală, filtrare globală și filtrare bazată pe observații [Zhuang and Frank, 1997; Frank and Marcu, 1999].

Se realizează, apoi, descrieri complete de stări folosind seturile filtrate, aceste noi stări fiind considerate succesori ai stării curente. Dacă sunt posibile mai multe schimbări calitative, atunci starea curentă va avea mai mulți succesori, iar simularea va produce o ramificație arborescentă de stări în graf.

Un observer calitativ fuzzy, așa cum este ilustrat în figura I. 22 , folosit la generarea de reziduuri, face apel la simularea calitativă bazată pe tehnicile convenționale de filtrare, realizând în felul acesta o filtrare bazată pe observații. Comportarea calitativă simulată a unei variabile trebuie să acopere corespondentul măsurătorii sale din sistem, în caz contrar calea din graf corespunzătoare acelei comportări fiind considerată inconsistentă și putând fi eliminată.

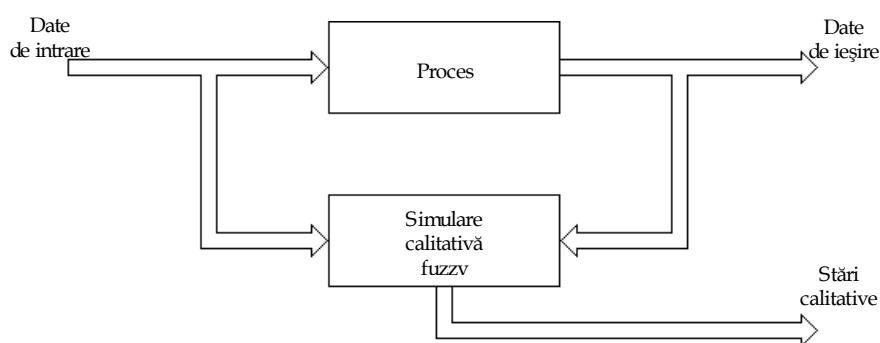


Fig. I. 22 Observer calitativ fuzzy

O anomalie produce o deviație a ieșirii sistemului într-o astfel de manieră încât corespondentul ieșirii estimate nu mai este consistent, altfel spus, o anomalie va produce o mulțime vidă de stări estimate calitativ, lucru imposibil în funcționarea normală (fără anomalii) a procesului.

b) *Observeri relaționali fuzzy* [Amann and Frank, 1997; Frank and Marcu, 1999]

Proiectarea unor observeri relaționali fuzzy pentru diagnoză se bazează pe principiile *modelării relaționale fuzzy*. Pentru aceasta se folosește o matrice relațională cuprinzând toate regulile ce descriu comportarea dinamică a sistemului. Sistemul este descompus în mai multe subsisteme MISO, structura completă a unui astfel de observor fuzzy fiind prezentată în figura I. 23 .

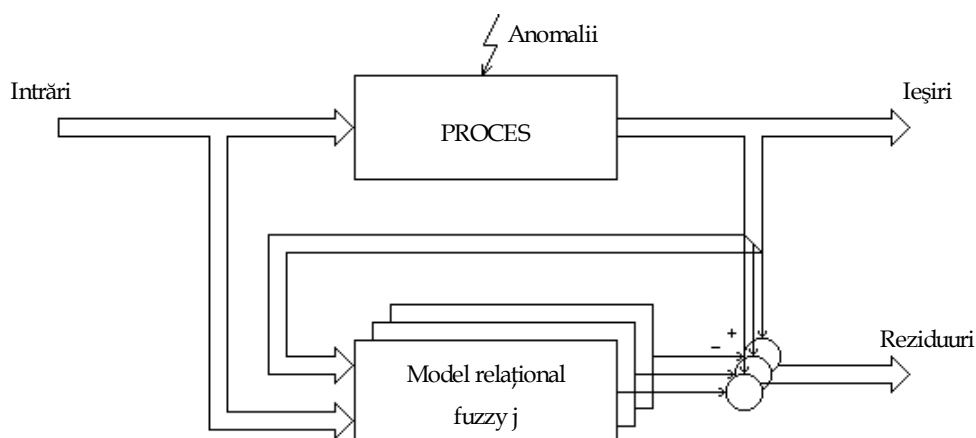


Fig. I. 23 *Observer relațional fuzzy*

Fiecare model relațional fuzzy este conceput să estimeze doar una din ieșirile procesului. Un model relațional fuzzy este compus din următoarele părți componente (figura I. 24):

- un bloc dinamic ce memorează eșantioane anterioare ale datelor de intrare – ieșire relevante;
- un bloc combinațional de forma produsului cartezian fuzzy;
- relația fuzzy însăși;
- interfața dintre măsurători și spațiul fuzzy, reprezentată prin blocurile de fuzzificare și defuzzificare.

Legătura funcțională fuzzy dintre intrările procesului și ieșirile acestuia este reprezentată, în domeniul discret, printr-o ecuație relațională fuzzy de intrare - ieșire de forma:

$$\begin{aligned}
 Y(k) &= R_0 \circ X(k) \\
 X(k) &= U(k) \times \dots \times U(k - k_u) \times Y(k - 1) \times \dots \times Y(k - k_y)
 \end{aligned}
 \tag{I. 10}$$

unde U , Y sunt variabile lingvistice asociate intrării u și, respectiv, ieșirii y ale procesului. Operatorul “ \times ” reprezintă operația logică ȘI între două variabile fuzzy, iar operatorul “ \circ ” reprezintă operația logică SAU între două variabile fuzzy. Întârzierile k_u , k_y sunt determinate printr-o procedură de identificare sau sunt cunoscute aprioric.

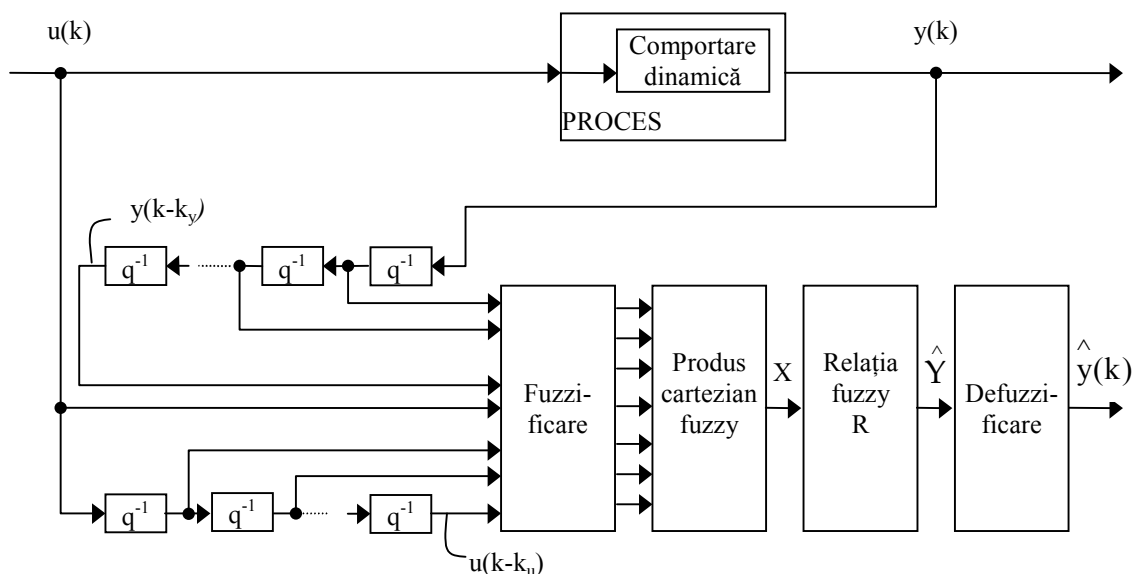


Fig. I. 24 Structura unui model relațional fuzzy

Pentru a obține matricea relațională pentru modelul nominal (fără anomalii), R_0 , trebuie rezolvată ecuația fuzzy (I. 10). O abordare alternativă este de a găsi o soluție R_a care să genereze un estimant \hat{Y} al ieșirii, suficient de apropiat de ieșirea modelului relațional Y dată de ecuația (I. 10) în sensul satisfacerii unui criteriu dat, care poate fi minimizarea erorii medii pătratice:

$$J(R) = \frac{1}{2} \cdot \sum_{k=1}^N [y(k) - \hat{y}(k)]^2,
 \tag{I. 11}$$

unde N este numărul de elemente ale setului de antrenare, $y(k)$ este ieșirea măsurată a procesului, iar $\hat{y}(k)$ este valoarea predictată, obținută după defuzzificare.

c) *Observeri funcționali fuzzy*

Un model funcțional fuzzy aproximează un proces dinamic neliniar folosind

modele liniare pe porțiuni [Nelles and Fischer, 1996; Ballé, *et al.*, 1998; Frank and Marcu, 1999]. Fiecare model local este valid într-o subregiune a spațiului de intrare. Partițiile în care modelele locale sunt valide nu sunt precis specificate, ci sunt de tip fuzzy (imprecis specificate). Din acest motiv, valabilitatea unui anumit model local într-un punct bine precizat din spațiul de intrare este exprimată printr-o funcție de ponderare, continuă, definită pe întreg spațiul de intrare, care ia valori în intervalul $[0,1]$. Valoarea 0 semnifică invaliditatea modelului, iar valoarea 1 semnifică faptul că modelul este sigur valid. Ieșirea modelului procesului se determină ca fiind o sumă ponderată a tuturor modelelor locale, aceasta conducând la o interpolare neliniară a ieșirilor modelelor locale. Premizele regulilor descriu regiuni fuzzy ale spațiului de intrare, iar părțile de concluzie reprezintă funcții exacte de intrările în model.

Funcția neliniară dinamică discretă cu m intrări $u_i, i = 1, \dots, m$ și o ieșire y :

$$\begin{aligned} y(k) &= f(\underline{x}(k)), \quad \underline{x}(k) = [x_1(k), \dots, x_n(k)] \\ \underline{x}(k) &= [u_1(k-1), \dots, u_1(k-k_{u,1}), \dots, u_m(k-1), \dots, u_m(k-k_{u,m}), \\ &\quad y(k-1), \dots, y(k-k_y)] \end{aligned} \quad (\text{I. 12})$$

este aproximată pe porțiuni prin modele liniare de tipul:

Regula R_i :

DACĂ $x_1(k)$ este $A_{i,1}$ ȘI $x_2(k)$ este $A_{i,2}$ ȘI ... ȘI $x_n(k)$ este $A_{i,n}$

ATUNCI $y_i(k) = w_{0,i} + w_{1,i} \cdot x_1(k) + w_{2,i} \cdot x_2(k) + \dots + w_{n,i} \cdot x_n(k)$

$i \in \{1, \dots, M\}$

unde $w_{0,i}, w_{1,i}, \dots, w_{n,i}$ sunt parametrii modelului de regresie liniară numărul i , x_1, \dots, x_n sunt intrările acestuia și $A_{i,1}, \dots, A_{i,n}$ sunt mulțimile fuzzy definite pe universul de discurs al intrărilor. Fiecare model local este valabil numai într-o anumită regiune a spațiului de intrare. Partițiile în care modelele locale sunt valide fiind imprecise (fuzzy) și nu exacte (precise), gradul de valabilitate al unui model local i , într-un anumit punct al spațiului de intrare, este exprimat sub forma unei funcții de ponderare, Φ_i , continuă, definită pe întreg spațiul de intrare și cu valori de la 0 (modelul local nu este valabil) la 1 (modelul local este sigur valabil). Astfel, ieșirea modelului global al sistemului este dată de suma

ponderată a ieșirilor celor M modele locale considerate:

$$y(k) = \sum_{i=1}^M y_i(k) \cdot \Phi_i(\underline{x}), \quad (\text{I. 13})$$

unde Φ_i este funcția de ponderare, pentru modelul i. Ecuația (I. 13) poate fi scrisă și sub forma:

$$\begin{aligned} y(k) &= \left(\sum_{i=1}^M w_{0,i} \cdot \Phi_i \right) + \left(\sum_{i=1}^M w_{1,i} \cdot \Phi_i \right) \cdot x_1(k) + \dots + \left(\sum_{i=1}^M w_{n,i} \cdot \Phi_i \right) \cdot x_n(k) = \\ &= w_0 + w_1 \cdot x_1(k) + \dots + w_n \cdot x_n(k) \end{aligned} \quad (\text{I. 14})$$

unde fiecare parametru w_i este rezultatul unei interpolări neliniare între parametrii modelelor locale $w_{i,j}$. În felul acesta se poate spune că s-a realizat o liniarizare dinamică, figura I. 25 ilustrând modul de realizare a unui astfel de model.

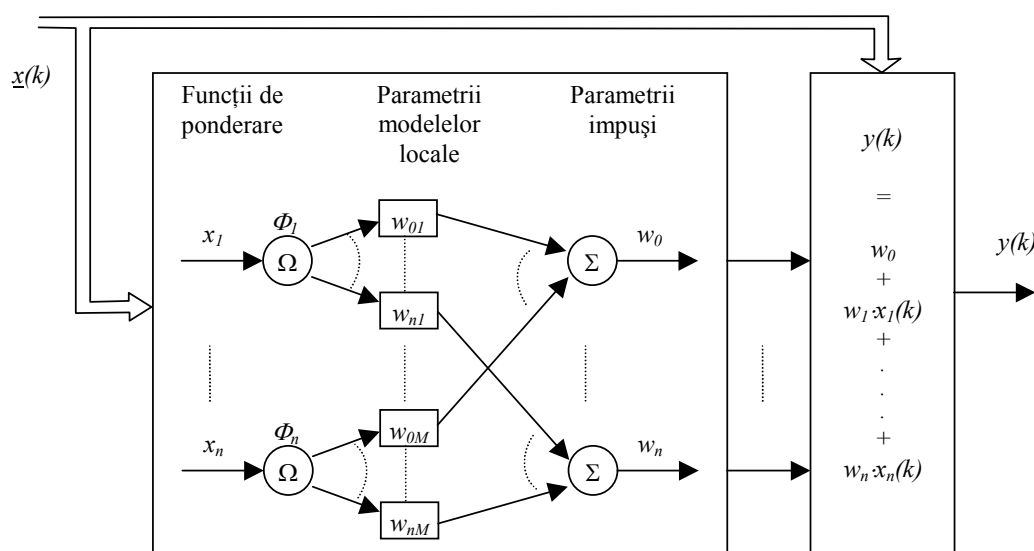


Fig. I. 25 Observer funcțional fuzzy

Pornind de la această teorie, a fost elaborat [Nelles and Fischer, 1996] un algoritm al arborelui modelelor locale (limba engleză: “Local Linear Model Tree – LOLIMOT) care generează o serie de modele funcționale fuzzy pentru fiecare submodel al procesului. Aceste modele sunt folosite pentru generarea de simptome (reziduuri), așa cum este prezentat în figura I. 26 .

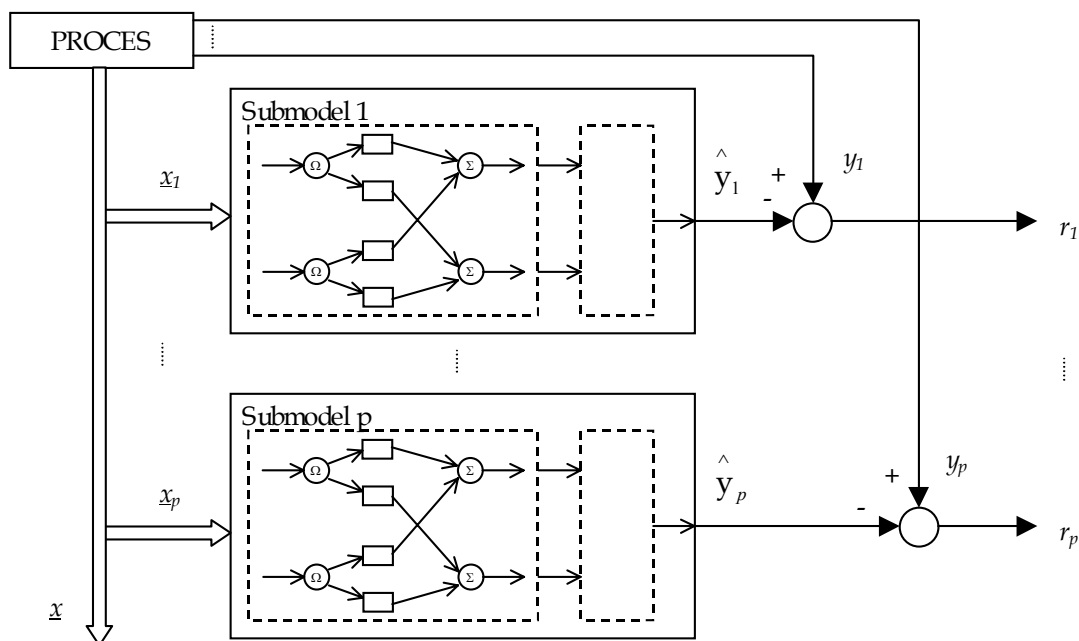


Fig. I. 26 Observer funcțional bazat pe algoritmul LOLIMOT

3.3 Tehnici fuzzy folosite în etapa evaluării simptomelor

Evaluarea simptomelor este un proces logic de luare a deciziei care transformă cunoștințe cantitative în exprimări calitative de tipul “da” sau “nu”. Scopul acestui proces este de a decide dacă și unde a apărut o anomalie în proces, încercând să se evite emiterea unor alarme false (decizii greșite). Acest proces de luare a deciziei poate fi privit și ca o problemă de clasificare, dorindu-se ca formele vectorului simptomelor să fie atribuite unor clase de funcționare anormală a procesului, prestabilite, sau clasei corespunzătoare funcționării normale. Există o serie de algoritmi de luare a deciziei, bine fundamentați teoretic, cum ar fi logica valorii de prag, teoria recunoașterii formelor și mai nou, algoritmi ce folosesc tehnica fuzzy și rețelele neuronale [Frank and Marcu, 1999]. Diferența principală dintre recunoașterea de forme și diagnoză constă în faptul că prima lucrează cu un set finit de clase, dinainte cunoscut, în timp ce a doua trebuie să-și adapteze regula de decizie la apariția unei noi clase corespunzătoare unui nou tip de comportare anormală a procesului.

Dificultatea principală a evaluării simptomelor reprezentate de către reziduuri este aceea că, în mod normal, reziduurile sunt afectate de zgomot, perturbații și

incertitudini (erori de modelare). Pentru a putea realiza o diagnoză corectă a anomaliilor este de dorit ca, în condiții ideale, reziduurile să fie nule în cazul comportării normale și nenule când apare o anomalie în proces. În practică, datorită prezenței zgomotelor și a incertitudinilor de modelare, este necesar să se folosească, în procesul de luare a deciziei, valori de prag mai mari decât zero pentru a evita apariția alarmelor false. Aceasta atrage, implicit, o reducere a sensibilității schemei de diagnoză. Ceea ce se realizează, în fapt, în ceea ce privește stabilirea valorii de prag, este un compromis între sensibilitatea detecției anomaliilor din proces și procentul de alarme false.

Proiectarea unui sistem de evaluare a reziduurilor bazat pe logica fuzzy parcurge următoarele etape:

- fuzzificarea reziduurilor;
- realizarea mecanismului de inferență;
- emiterea alarmelor de funcționare anormală.

a) Fuzzificarea reziduurilor

Prin fuzzificare se realizează atribuirea unui număr corespunzător de mulțimi fuzzy fiecărei componente a reziduului, $r_i, i = \overline{1, p}$. Această etapă are o importanță deosebită deoarece, prin această asociere, este influențată proprietatea întregii scheme de diagnoză de a localiza corect anomaliile [Frank and Marcu, 1999].

Presupunând că se atribuie componentei r_i a reziduului, s mulțimi fuzzy $r_{i,k}, k = \overline{1, s}$, atunci se realizează următoarea transformare:

$$r_i \rightarrow r_{i,1} \circ r_{i,2} \circ \dots \circ r_{i,s}, \quad r_i \rightarrow [0,1],$$

unde “ \circ ” este operatorul fuzzy de compunere. Principiul care stă la baza fuzzificării reziduului este descris în figura I. 27 . Acest proces poate fi privit și ca o procedură de generare a unei valori de prag robuste. Așa cum se observă și în figura I. 27 , pentru a caracteriza absența sau prezența unei anomalii în sistem s-au folosit două mulțimi fuzzy.

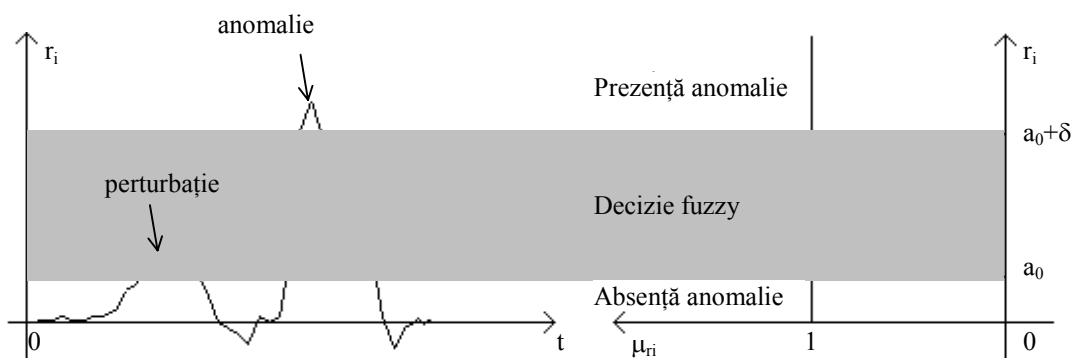


Fig. I. 27 Principiul fuzificării reziduului

Parametrul a_0 trebuie să fie stabilit proporțional cu amplitudinea zgomotului și cu efectele incertitudinilor de modelare. Parametrul δ poate fi ales ca fiind varianța zgomotului din proces, a perturbațiilor și a erorilor de modelare variabile în timp. Funcția de apartenență folosită în figura I. 27 este cea mai simplă formă de fuzificare a reziduului, însă se poate extinde folosind mai multe mulțimi fuzzy care să descrie prezența sau absența anomaliilor în proces.

O altă abordare este cea a adaptării funcțiilor de apartenență și a regulilor implicate. Acest mecanism este necesar atunci când condițiile de operare ale sistemului se schimbă. Dacă se realizează, apoi, defuzificarea, se obține o valoare de prag adaptivă. Valoarea de prag este adaptată în funcție de variațiile intrării u și ieșirii y ale procesului, în termenii unor reguli între mulțimi fuzzy caracterizate de funcții de apartenență corespunzătoare. Relația rezultată pentru valoarea de prag adaptivă este:

$$J(u, y) = J_0 + \Delta J(u, y), \quad (\text{I. 15})$$

în care $J_0 = J(u_0, y_0)$ reprezintă valoarea de prag constantă corespunzătoare punctului de operare (u_0, y_0) în care s-a ținut cont doar de perturbațiile staționare (de exemplu zgomotul de măsură), fără a se ține cont de prezența vreunei anomalii. Elementul $\Delta J(u, y)$ reflectă efectele erorilor de modelare datorate devierii procesului din punctul de operare.

b) Mecanismul de inferență

În general, scopul procedurii de luare a deciziei este acela de a evidenția, pe baza unui set R de reziduuri, anomalia f_j , dintr-un set F de anomalii, care a

apărut în funcționarea procesului. Pentru a rezolva această problemă, este necesar să se creeze un set de condiții de tip fuzzy prin compunerea tuturor combinațiilor de mulțimi fuzzy $r_{i,k}$ asociate tuturor celor p componente ale reziduului. Regulele ce trebuie evaluate sunt de tipul [Frank and Marcu, 1999]:

DACĂ (efect = $r_{i,1}$) ȘI DACĂ (efect = $r_{i,2}$) ...,
ATUNCI (cauza = f_j).

Aceste reguli formează, împreună cu valorile reziduului, baza de cunoștințe a sistemului expert pentru diagnoză.

Evaluând aceste reguli, devine posibil în condiții precise, să se determine anomalia corespunzătoare fiecărei combinații de reziduuri. Altfel spus, condițiile fuzzy realizează o transformare din spațiul reziduurilor în spațiul anomaliilor, cu ajutorul regulilor din baza de cunoștințe. Acest lucru poate fi ilustrat sub forma unui graf orientat sau arbore al anomaliilor (figura I. 28), în care f_j sunt anomaliile, iar r_i sunt reziduurile.

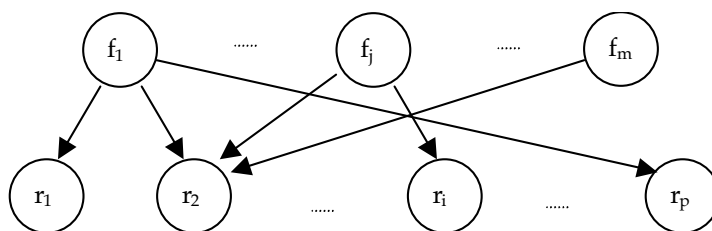


Fig. I. 28 *Reprezentarea procedurii de decizie printr-un arbore al anomaliilor*

Baza de cunoștințe corespunzătoare unui proces complex trebuie să conțină și informații despre conexiunile dintre subsistemele componente ale procesului. Aceste cunoștințe pot fi exprimate sub forma unor reguli fuzzy cu scopul de a descrie comportarea anormală a sistemului într-o manieră fuzzy (imprecisă).

Pentru a asigura existența unei soluții unice a mecanismului de inferență, trebuie verificată consistența fiecărei reguli, lucru ce poate fi efectuat concomitent cu efectuarea raționamentelor [Kiupel and Frank, 1996].

O altă abordare pentru evaluarea reziduurilor (simptomelor) folosește un arbore de clasificare [Füssel, *et al.*, 1997]. Metoda, numită și a *Arborelui de Clasificare*

cu *Auto-Antrenare* (limba engleză: “Self-Learning Classification Tree” – SELECT), combină capacitatea de a învăța, specifică rețelelor neuronale, cu transparența raționamentelor efectuate oferită de sistemele fuzzy. În figura I. 29 este reprezentat un exemplu de arbore rezultat în urma aplicării procedurii SELECT, în care $S_i, i = \overline{1,3}$, sunt simptomele (reziduurile).

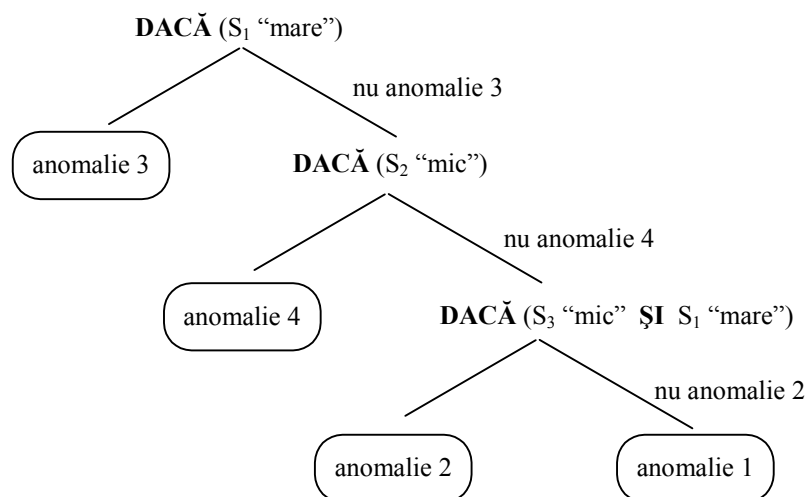


Fig. I. 29 Arbore de clasificare SELECT

Procedura constă în următoarele etape [Frank and Marcu, 1999]:

1. crearea unor funcții de apartenență folosind, de exemplu, proceduri de grupare nesupervizată [Bezdek and Pal, 1992];
2. selectarea unei reguli fuzzy pentru a evidenția anomalia cea mai simplă de separat;
3. toate datele aparținând situației anormale deja diagnosticată (la pasul anterior) sunt eliminate din setul de antrenare, creându-se un nou set de antrenare ce conține date corespunzătoare unui număr mai mic cu o unitate de clase decât anterior; se revine la pasul precedent și în mod iterativ se selectează noi reguli asociate altor clase de anomalii.
4. se creează astfel un arbore de clasificare secvențial, în care se testează pe rând diferite comportări anormale.
5. adăugarea unor reguli cunoscute a-priori în vârful arborelui obținut;
6. optimizarea parametrilor, astfel încât să se obțină precizie de clasificare optimă.

c) Emiterea alarmelor de funcționare anormală

Etapa finală a diagnozei anomaliilor este evidențierea în mod corespunzător, către operator, a situațiilor de funcționare anormală. Aceasta se poate face printr-o procedură de defuzzificare [Ayoubi, 1996a], deoarece în urma etapei de inferență se emit o serie de semnale prezentate într-o formă fuzzy, câte un semnal pentru fiecare clasă de comportare anormală și unul pentru comportarea normală. Aceste semnale se numesc *semnale fuzzy de indicare a anomaliei*. Un astfel de semnal reprezintă gradul de compatibilitate al anomaliilor posibile cu o anumită comportare anormală a sistemului. La ieșirea etapei de inferență nu se mai asociază nici o mulțime fuzzy, decizia finală legată de emiterea alarmei de prezență a unei anomalii rămânând, deci, să fie luată de către operator. Mai concret, în loc să se folosească forma standard:

... ATUNCI anomalie = mare

în care “mare” este una din mulțimile fuzzy asociate ieșirii, se folosește următorul format:

... ATUNCI anomalie = anomalie_j

unde “anomalie_j” este singura mulțime fuzzy asociată anomaliei f_j . Aceasta se aplică similar tuturor anomaliilor considerate. Mulțimea fuzzy “anomalie_j” are o valoare a funcției de apartenență care este aceeași cu a ieșirii compuse rezultată în urma evaluării reziduurilor, așa cum este ilustrat în figura I. 30 .

Funcțiile de apartenență corespunzătoare mulțimii fuzzy atașate unei anume clase de forme ω_j , $\mu_j = \mu(f_j)$, este o funcție scalară de timp formată din porțiuni continue întrerupte de salturi abrupte. Oscilațiile funcției în jurul valorii medii reflectă modificări în sistem, dar care sunt în interiorul aceleiași clase de comportare. O scădere a valorii acestei funcții indică faptul că sistemul părăsește clasa j , în timp ce o creștere a acesteia indică faptul că sistemul se îndreaptă către această clasă de comportare. O evoluție a sistemului din clasa ω_j către clasa ω_i este evidențiată, la nivelul funcțiilor de apartenență, printr-o scădere a funcției μ_j asociată cu o creștere a funcției μ_i , dacă clasa ω_i este o clasă de comportare cunoscută. Dacă evoluția se face către o clasă de comportare

necunoscută, ω_x , atunci se observă scăderea valorii tuturor funcțiilor de apartenență asociate claselor de comportare cunoscute [Frank and Marcu, 1999].

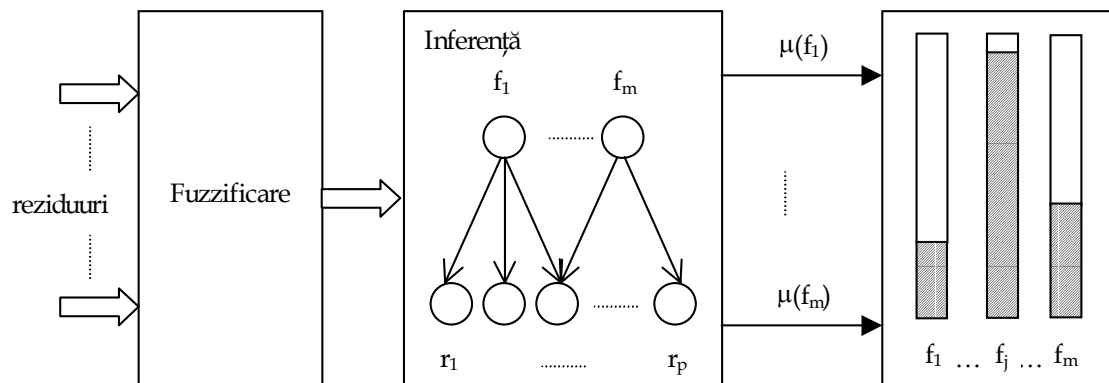


Fig. I. 30 Evaluarea reziduurilor și emiterea alarmelor

Sinteza elementară a sistemelor MIMO

Introducere

Scopul acestui material-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 comoratarea 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.

1 Lege de comandă, stabilizabilitate, alocabilitate

A) Descriere teoretică

Considerăm un sistem liniar

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t) \end{cases} \quad x(0) = x_0 \quad (5.1)$$

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 5.1 Pentru un sistem (A, B, C, D) dependența

$$u = Fx + Gv \quad (5.2)$$

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 mărării de stare !

După implementarea comenzii (5.2) sistemul în buclă închisă devine

$$\begin{cases} \dot{x}(t) = (A + BF)x(t) + BGv(t), \\ y(t) = (C + DF)x(t) + DGv(t) \end{cases} \quad x(0) = x_0 \quad (5.3)$$

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

$$R = [b \quad AB \quad \dots \quad A^{n-1}b]$$

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 alege $\tilde{F} \in \mathcal{R}^{m \times n}$ 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 invariante 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}

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

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

$$(A - BF)x_j = \lambda_j x_j, \quad j = 1 : n \quad (5.4)$$

dacă și numai dacă

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

$$X = [x_1 \quad x_2 \quad \dots \quad x_n] \quad (5.5)$$

este nesingulară;

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

iii)

$$(\lambda_j I - A)x_j \in \text{Im}B, \quad j = 1 : n. \quad (5.6)$$

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

$$B = QR = [Q_1 \quad Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1 \quad (5.7)$$

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

$$\Lambda \stackrel{\text{def}}{=} \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \quad (5.8)$$

cu notația (5.5), relația (5.4) se scrie

$$(A - BF)X = X\Lambda. \quad (5.9)$$

Ținând seama de (5.7), (5.9) se descompune în

$$Q_2^T (X\Lambda - AX) = 0 \quad (5.10)$$

și

$$F = R_1^{-1} Q_1^T (A - X\Lambda X^{-1}). \quad (5.11)$$

Relația (5.11) transferă problema fixării unei soluții a problemei de alocare a valorilor proprii într-o problemă de selecție a matricii X a vectorilor proprii care este supusă restricțiilor 5.6. Aceste restricții se pot scrie și în forma

$$x_j \in \mathcal{S}_j = \text{Ker} [Q_2^T (\lambda_j I - A)], \quad j = 1 : n. \quad (5.12)$$

În cazul în care perechea (A, B) este controlabilă subspațiul liniar \mathcal{S}_j din (5.12) are dimensiunea $\dim \mathcal{S}_j = m$.

Cu scopul de a aprecia sensibilitatea valorilor proprii ale unei matrici $A \in \mathcal{R}^{n \times n}$ 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 matricii A prin

$$c_i = \left| \frac{d\lambda_i(\varepsilon)}{d\varepsilon} \right|_{\varepsilon=0} \stackrel{\text{def}}{=} \left| \lambda_i'(0) \right|. \quad (5.13)$$

O margine superioară a lui c_i este dată de

$$(c_i)_{\max} = \frac{\|y_i\| \|x_i\|}{|y_i^T x_i|} \quad (5.14)$$

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.

Robușetea *întregului* spectru de valori proprii $\lambda(A)$ al unei matrici $A \in \mathcal{R}^{n \times n}$ 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 robușeții spectrului de valori proprii al unei matrici simple (i.e. având un set complet de vectori proprii liniar independenți) numărul de condiție la inversare al matricii 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, robușetea spectrului Λ al matricii $A - BF$ 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 (5.7) a lui B vectorii proprii ai matricii $A - BF$ aparțin subspațiilor liniare $\mathcal{S}_j, j = 1 : n$ definite în (5.12).

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

$$x_k = e_k \quad (5.15)$$

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

$$x_{k,k+1} = \frac{1}{\sqrt{2}} (e_k \pm ie_{k+1}) \quad (5.16)$$

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

$$L \stackrel{\text{def}}{=} X\Lambda X^H = \text{diag}(\Lambda_1, \Lambda_2, \dots, \Lambda_q) \quad (5.17)$$

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 (5.15), (5.16) a vectorilor proprii, conform (5.11), este

$$F = B^{-1}(A - L). \quad (5.18)$$

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

$$\nu_2 \stackrel{\text{def}}{=} \kappa_2(X) = \|X\|_2 \|X^{-1}\|_2 \quad (5.19)$$

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 (5.19) capătă expresia

$$X = SD, \quad (5.20)$$

unde

$$S = [S_1 \ S_2 \ \dots \ S_n] \in \mathcal{R}^{n \times mn}, \quad (5.21)$$

$$D = \text{diag}(d_1, d_2, \dots, d_n) \in \mathcal{R}^{mn \times n}. \quad (5.22)$$

Putem reformula problema de mai sus în următorii termeni.

Fiind dată matricea S cu structura (5.21), să se găsească X din (5.22) care minimizează ν_2 din (5.19) în raport cu toți D de forma (5.22).

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

$$\chi_j = \text{Im} X_j = \text{Im} [x_1 \ x_2 \ \dots \ x_{j-1} \ x_{j+1} \ \dots \ x_n] \quad (5.23)$$

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,

$$X_j = Q_j R_j = [\bar{Q}_j \ \bar{y}_j] \begin{bmatrix} \bar{R}_j \\ 0 \end{bmatrix} \quad (5.24)$$

factorizarea **QR** a matricei $X_j \in \mathcal{R}^{n \times (n-1)}$ din (5.23), 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

$$x_j^{\text{nou}} = \frac{S_j d_j^*}{\|S_j d_j^*\|} \quad (5.25)$$

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

$$x_j^{nou} = \frac{S_j S_j^T \bar{y}_j}{\|S_j S_j^T \bar{y}_j\|}. \quad (5.26)$$

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 5.1 (Date matricea $S \in \mathcal{R}^{n \times nm}$ din (5.21) 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. $X_j = [X(:, 1 : j-1) \quad X(:, j+1 : n)]$
 2. Se calculează ultima coloană $\bar{y}_j = Q_j(:, n)$ a matricei ortogonale Q_j din factorizarea **QR** (5.24) a matricei X_j .
 3. Se calculează x_j^{nou} din (5.25) conform secvenței
 1. $w = S_j^T \bar{y}_j$
 2. $w = S_j w$
 3. $x_j^{nou} = w / \|w\|$
 4. $X = [X(:, 1 : j-1) \quad x_j^{nou} \quad X(:, j+1 : n)]$
 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 5.2 (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. $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$

2. Dacă $m = n$
atunci

1. $X = I_n$
2. Se rezolvă sistemul linear matriceal $BF = A - \Lambda$ în raport cu F (vezi 5.18)

altfel

1. Se calculează factorizarea **QR** (5.7) 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 (5.12)
5. $S = [S_1 \ S_2 \ \dots \ S_n]$
6. Se calculează setul optim de vectori proprii pentru sistemul în circuit închis $X = \text{alloc_vec}(S, n_iter, tol)$
7. Se calculează matricea A_0 a sistemului în circuit închis prin rezolvarea sistemului linear matriceal $A_0 X = X \Lambda$
8. Se calculează matricea de reacție F , conform (5.11), prin rezolvarea sistemului matriceal superior triunghiular $R_1 F = Q_1^T (A - A_0)$.

C) Sarcini de lucru

1. 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) .

2. 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. $f^T \leftarrow f^T U$

3. Implementați în MATLAB algoritmul de alocare bazat pe forma Schur.
4. Implementați în MATLAB algoritmul de alocare cu minimizare de normă.

5.2 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

$$\begin{cases} \dot{w}(t) = Jw(t) + Hy(t) + Mu(t) \\ \hat{x}(t) = Kw(t) + Ny(t) + Pu(t) \end{cases} \quad (5.27)$$

care să îndeplinească 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 estimăția 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 5.4 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

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

funcție de care se deosebesc mai multe tipuri de estimatori.

Estimatori de tip 1: N=0

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

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

și putem alege $K = I$, $V = I$, ramanand de satisfacut doar conditia

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

Aceasta 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

$$\begin{cases} \dot{w}(t) &= (A - HC)w(t) + Bu(t) - HDu(t) + Hy(t) \\ \hat{x}(t) &= w(t) \end{cases} \quad (5.28)$$

sau forma echivalenta

$$\dot{\hat{x}} = A\hat{x} + Bu + L(C\hat{x} + Du - y) \quad (5.29)$$

in 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 $C = [O \quad I_p]$.

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 identic 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)

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

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

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

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

rezultând

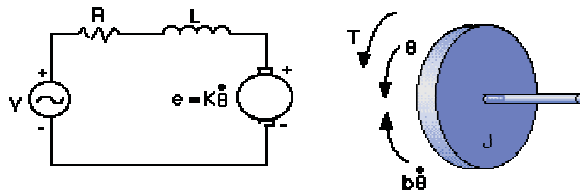
$$\begin{cases} \dot{w}(t) &= (A_1 + V_2 A_2) w(t) + H y(t) + (B_1 + V_2 B_2 - HD) u(t) \\ \hat{x}(t) = \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{bmatrix} &= \begin{bmatrix} I_{n-p} \\ O \end{bmatrix} w(t) + \begin{bmatrix} -V_2 \\ I_p \end{bmatrix} y(t) + \begin{bmatrix} V_2 D \\ -D \end{bmatrix} u(t) \end{cases} \quad (5.30)$$

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

1. Considerăm un motor de curent continuu



pentru care avem următorii parametri:

$$\text{- momentul de inerție al rotorului } J = 5E - 5kg * m^2/s^2$$

- factorul de amortizare al sistemului mecanic $b = 1E - 4Nm/s$
- constanta motorului $K = Ke = Kt = 0.1Nm/Amp$
- rezistenta din circuitul de comanda pe indus $R = 1ohm$
- rezistenta din circuitul de comanda pe indus $L = 1E - 3H$
- intrare V : tensiunea de alimentare a infasurarii rotorului
- iesire $theta$: pozitia unghiulara a arborelui motorului
- rotorul si 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

$$\begin{aligned} J \ddot{\theta} + b \dot{\theta} &= K i_a \\ L \frac{di_a}{dt} + R i &= V - K \dot{\theta} \end{aligned}$$

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];

- (a) 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)

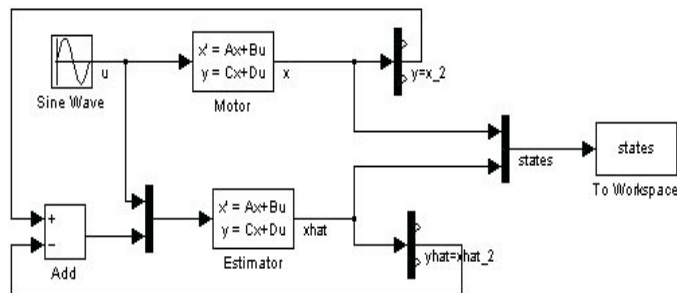
% redefinim C si D pentru a avea la iesire toate starile motorului -
% vom folosi doar y = x_2 pentru estimator
C = [1, 0, 0; 0, 1, 0; 0, 0, 1];
D = [0; 0; 0];

% 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];

```

(b) Creați schema Simulink a sistemului și a estimatorului.

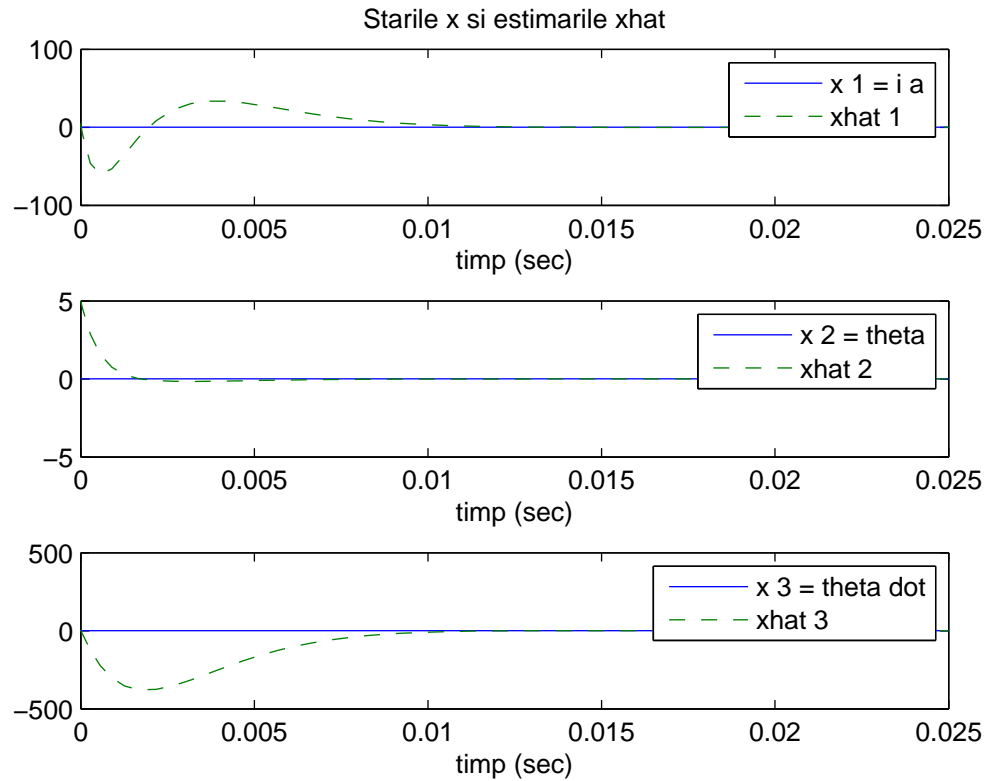


(c) Reprezentați grafic stările și estimatele lor.

```

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');

```



2. 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 $C = [O \quad I_p]$. T se alege de forma $T = \begin{bmatrix} \bar{C} \\ C \end{bmatrix}$, unde \bar{C} este de dimensiune $(n - p \times n)$.

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

```
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 = inv(T)*K;
>> N = inv(T)*N;
>> P = inv(T)*P;
```

5.3 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 (5.28). Obținem atunci ecuațiile dinamice ale compensatorului - numit compensator Kalman -

$$\begin{cases} \dot{\hat{x}} &= (A + LC) \hat{x} + Bu + LDu - Ly \\ u &= F \hat{x} \end{cases} \quad (5.31)$$

sau inca

$$\begin{cases} \dot{\hat{x}} &= (A + LC + BF + LDF)\hat{x} - Ly \\ u &= F\hat{x} \end{cases} \quad (5.32)$$

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 in reactie inversa cu sistemul original si aplicand transformarea de similaritate $T := \begin{bmatrix} I & O \\ -I & I \end{bmatrix}$ obtinem pentru matricea de stare a sistemului rezultat in bucla inchisa

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

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 stabiliza 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 $M(s)$:

$$\tilde{T} = \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array} \right] = \left[\begin{array}{cc|c} A & O & B \\ -C & O & O \\ \hline O & I_p & O \end{array} \right]. \quad (5.33)$$

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 $T(s)$ (și de același ordin cu $T(s)$).

Obținem pentru ecuațiile estimatorului de ordin redus

$$\begin{cases} \dot{w} &= (A + LC)w + (A + LC)L\tilde{y} + Bu \\ \hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} &= \begin{bmatrix} I_{n-p} \\ O \end{bmatrix} w + \begin{bmatrix} L \\ I_p \end{bmatrix} \tilde{y} \end{cases} \quad (5.34)$$

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

$$u = \begin{bmatrix} F & \tilde{F} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = F(w + L\tilde{y}) + \tilde{F}\tilde{y}. \quad (5.35)$$

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

$$\text{rank} \begin{bmatrix} -A & -B \\ C & O \end{bmatrix} = n + p. \quad (5.36)$$

Deci în ipotezele (A, B) stabilizabilă, (C, A) detectabilă și (5.36) 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

$$\begin{aligned} \dot{w} &= (A + LC)w + (A + LC)L\tilde{y} + Bu \\ \hat{x}_1 &= w + L\tilde{y} \\ \hat{x}_2 &= \tilde{y} \\ u &= F(w + L\tilde{y}) + \tilde{F}\tilde{y} \end{aligned} \quad (5.37)$$

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

$$\begin{aligned} \dot{\hat{x}}_1 &= (A + LC + BF)\hat{x}_1 + B\tilde{F}\hat{x}_2 + L\epsilon \\ \hat{x}_2 &= \epsilon \\ u &= F\hat{x}_1 + \tilde{F}\hat{x}_2 \end{aligned} \quad (5.38)$$

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

$$K(s) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right] = \left[\begin{array}{cc|c} A + LC + BF & B\tilde{F} & L \\ \hline O & O & I \\ \hline F & \tilde{F} & O \end{array} \right]. \quad (5.39)$$

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

$$\bullet \text{rank} \begin{bmatrix} -A & -B \\ C & O \end{bmatrix} = n + p$$

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.

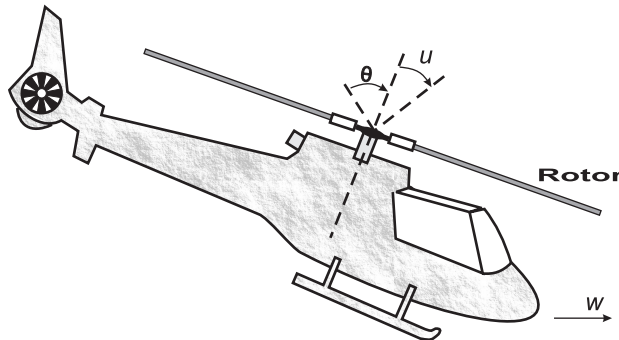
Pasul 3. Compensatorul $K(s)$ este dat de

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

B) Exemple

1. Modelul liniarizat al mișcării longitudinale a unui elicopter poate fi descris de următorul sistem de ecuații:

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



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:

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

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 $\Lambda(A - KC) = P_{est}$.

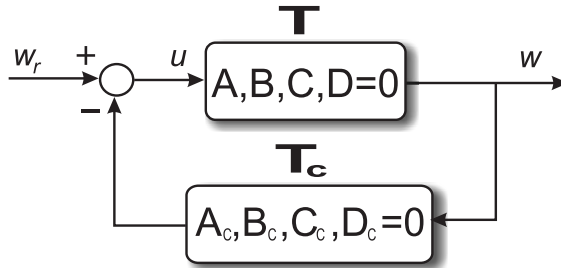
Pentru exemplul în cauză folosiți următoarele date pentru P_{aloc} și P_{est} :

$$P_{aloc} = [-0.6565 \quad -1-j \quad -1+j]$$

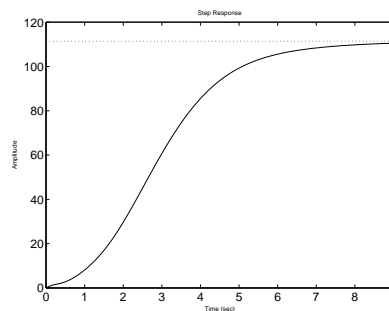
$$P_{est} = [-8 \quad -4+\sqrt{3}*j \quad -4-\sqrt{3}*j]$$

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 $\Lambda(A - KC) = P_{est}$.

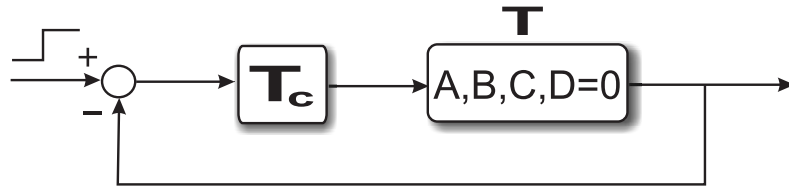
Verificați stabilitatea sistemului H în buclă închisă, dat de $H=feedback(T, T_c)$.



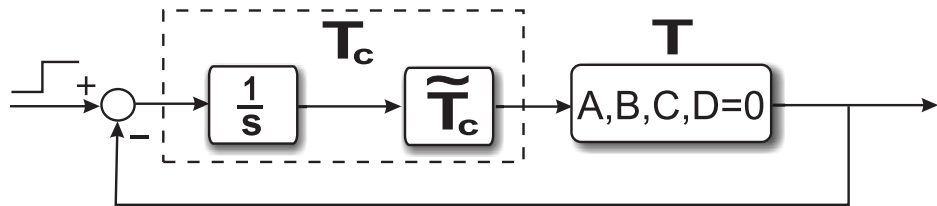
Comparați polii lui H cu elementele vectorilor P_{aloc} și P_{est} . Ce constatați? Trașați răspunsul sistemului în buclă închisă la intrare treaptă.



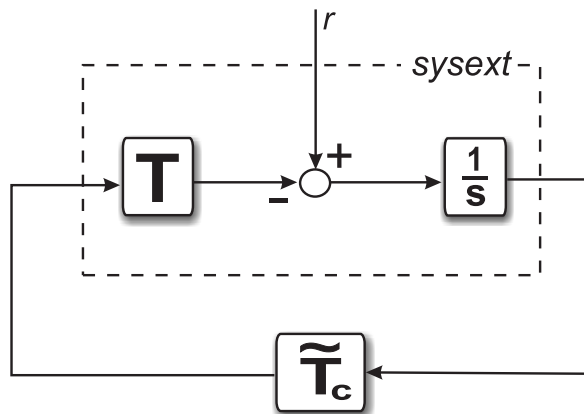
- d) **Problema reglării la referință treaptă unitară.** Pentru rezolvarea problemei reglării, ideea este de a introduce modelul intern al referinței de tip treaptă ($\frac{1}{s}$) în funcția de transfer a căii directe, deci în regulator (dacă ea nu este deja conținută în modelul sistemului).



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.)

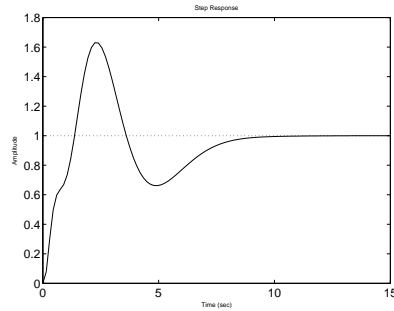


Așadar pentru `integ=ss(0,1,1,0)`, respectiv `T=ss(A,B,C,0)` apelăm funcția de la c) cu parametri de intrare: `sysext=series(integ,T)` și vectorii \tilde{P}_{aloc} și \tilde{P}_{est} de lungime 4 (adică ordinul matricei A a lui `sysext`). Conform explicațiilor anterioare, acest apel va returna o realizare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

2. 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);

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

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

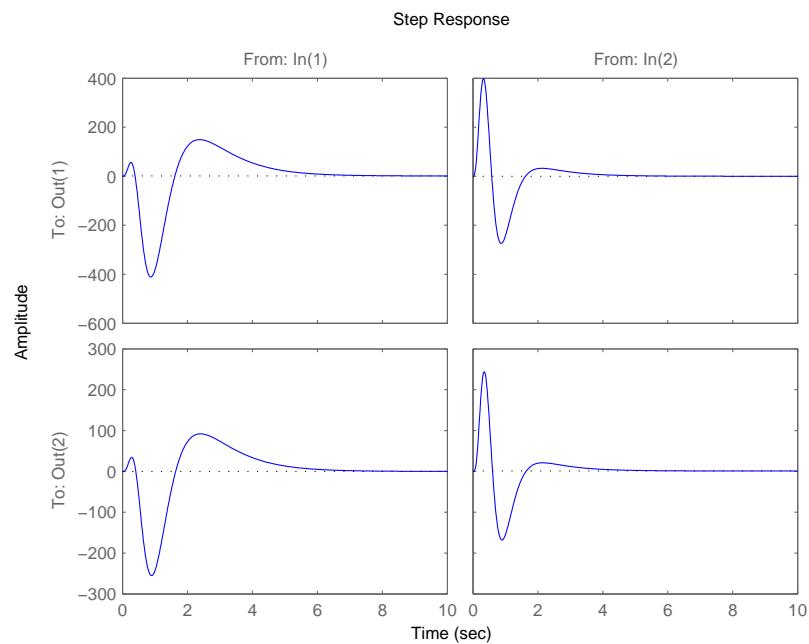
[l1,c1] = size(A1);
[l2,c2] = size(A2);

Ak = [A1 A2;zeros(c1+c2-l1,c1) zeros(c1+c2-l1,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.

5.4 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))/G_tilda(1,1);
    K_tilda = [tmp1 tmp2];
end

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

```