Universitatea Politehnica Bucureşti
Facultatea de Automatică şi Calculatoare

LUCRARE DE LICENŢĂ

Identificarea şi controlul metaeuristic al unei instalaţii hidraulice. Studiu de caz: instalaţie cu două rezervoare.

Absolvent: **Mitrică Răzvan-Ionuţ**

Coordonator: **Conf. Dr. Ing. Janetta Culiţă**
Prof. Dr. Ing. Dan Ştefănoiu

June 22, 2020

Cuprins

Listă de figuri

1 Introducere

1.1 Motivaţie

Apa reprezintă o resursă foarte importantă, de-a lungul mileniilor, oamenii încercând să interacţioneze în foarte multe moduri cu aceasta. Odată cu trecerea timpului a apărut şi nevoia de a putea controla acest element. Mai mult, concomitent cu dezvoltările tehnologice şi cu apariţia ramurii automaticii, a apărut nevoia controlului anumitor caracteristici ale apei, fie caracteristici proprii, fie caracteristici ca parte a unui sistem. Această problemă rămâne în continuare una care dă bătăi de cap inginerilor, mai ales în sistemele cu vase de diferite forme sau care comunică.

Sistemele fluidice sunt necesare în mod esenţial în cadrul proceselor vitale din cadrul industriilor precum, industria petro-chimică, energetică, de tratare a apei sau în agricultură. Majoritatea acestor procese se bazează pe interacţiunea dintre mai multe bazine, în care controlul nivelului de lichid şi debit dintre bazine reprezintă o problemă majoră, în special datorită faptului că pot fi descrise ca sisteme multivariabile cu dinamică neliniară. Prin urmare, procesele cu mai multe bazine au captat atenţia comunităţii inginereşti şi diferite strategii de control au fost proiectate şi implementate pe sistemele cu 4 bazine. Procesele cu mai multe bazine referite în lucrările de specialitate sunt alcătuite din 2, 3 sau 4 bazine cu formă rectangulară sau cilindrică.

1.2 Obiectiv

Scopul acestei lucrări este identificarea unui model matematic care să estimeze cât mai precis comportamentul unei instalaţii cu două rezervoare (instalaţia ASTANK 2), precum şi proiectarea unei legi de comandă care să asigure (în buclă închisă) păstrarea nivelului de lichid în bazine la anumite valori şi rejectarea perturbaţiilor. Procesul este neliniar prin natura sa, iar datele masurate furnizate de senzori, pe care le vom folosi in identificarea experimentală, sunt afectate de zgomot de masură (perturbaţii).

Setul optim de indici structurali ai modelului de identificare va fi determinat cu ajutorul unei metaeuristici care maximizeaă o funcţie criteriu bazată pe raportul semnal-zgomot al modelului. De asemenea, pentru controlul nivelului de lichid din rezervore se va proiecta câte un regulator PID pentru fiecare mărime controlată, ai căror parametri vor fi determinaţi tot printr-o metaeuristică (algoritm genetic).

1.3 Descrierea lucrării

În prima parte a lucrării este prezentată instalaţia ASTANK 2, atât din punct de vedere funcţional cât şi constructiv. Se va prezenta problematica modelării dinamicii sistemului si necesitatea de a construi un model de identificare experimental.

Partea a doua cuprinde formularea problemei de identificare şi tipurile de modele utilizate pentru modelarea dinamicii procesului, atât pentru componenta utilă, cât şi pentru componenta de zgomot a procesului. Ambele componente vor fi identificate pe

baza seturilor de date intrare/ieşire disponibile în urma unui experiment econometric. Tot în acest capitol, va fi descrisă şi funcţia criteriu folosită pentru obtinerea unei structuri optime a modelului de identificare.

În a treia parte se prezintă doi algoritmi (metaeuristici) folosiţi pentru găsirea setului optim de indici structurali care să îndeplinească maximizarea funcţiei criteriu specificate in capitolul anterior. Aceşti algoritmi (algoritm bazat pe roi de licurici şi algoritm genetic) se aplică separat pentru partea utilă, modelată cu ajutorul unui model de tip ARX, cât şi pentru partea de zgomot, care este modelată cu un model de tip ARMA. Prin urmare, se doreşte găsirea a 2 astfel de modele optimale, care, împreună, să constituie modelul global adecvat al procesului.

A patra parte se concentrează pe găsirea a 2 controllere din clasa regulatoarelor PID, ce trebuie să calculeze în buclă închisă comanda care să asigure urmărirea referinţei şi rejectarea perturbaţiilor.

Ultimul capitol cuprinde rezultatele obţinute prin aplicarea celor două metode menţionate pentru identificarea modelului optimal al instalaţiei. Se vor prezenta, comparativ, rezultatele obţinute cu ajutorul algoritmului genetic şi algoritmului bazat pe licurici. Modelele optime identificate vor fi folosite la etapa de reglare, în conjuncţie cu regulatoarele din clasa PID acordaţi cu ajutorul algoritmului genetic.

2 Prezentare instalaţie

Instalaţia ASTANK2 [2] reprezintă un sistem hidraulic care include 2 bazine deschise, de aceeaşi înălţime. Bazinele sunt alimentate cu apă dintr-un al treilea bazin (de acumulare) prin intermediul unei reţele de distibuţie şi pompare a apei. Principala caracteristică a acestei instalaţii o reprezintă bazinul teşit, care introduce neliniarităţi în proces.

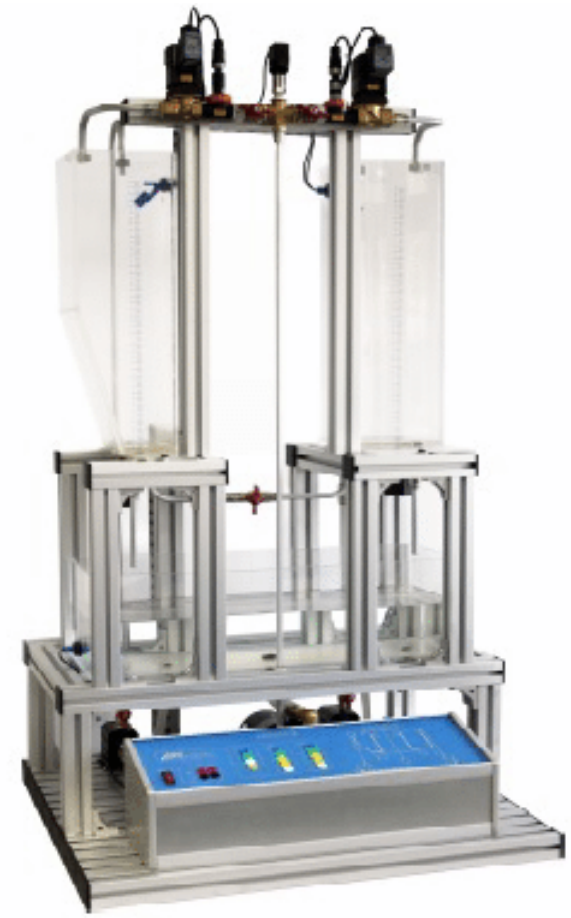

Figure 2.1: ASTANK2 [2]

2.1 Descrierea instalaţiei

Cele două bazine pot fi conectate prin intermediul conductei sau să opereze independent, în funcţie de poziţia robinetului R. Golirea celor două bazine poate fi controlată prin intermediul robineţilor R11 şi R21, care pot fi deschişi sau închişi. Astfel, numeroasele configuraţii disponibile conduc la o mare varietate de experimente ce pot fi întreprinse pe baza instalaţiei.

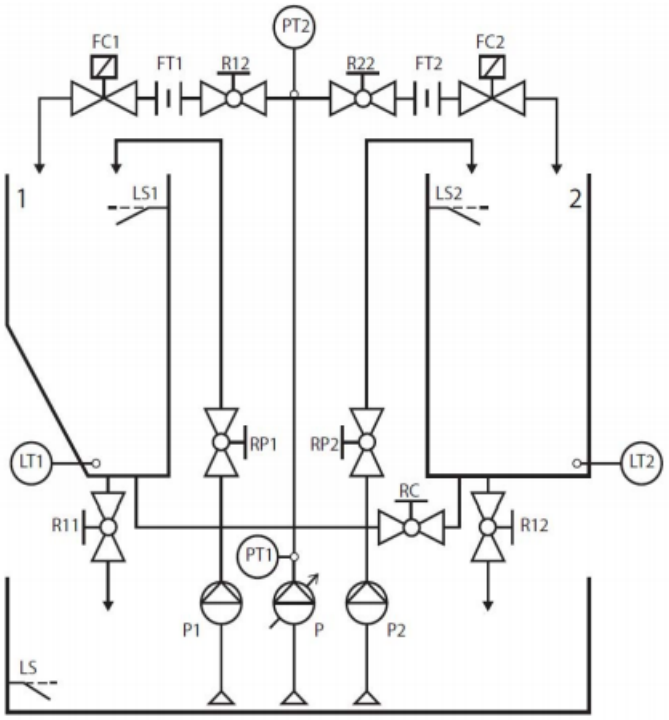

Figure 2.2: Schema arhitecturii ASTANK2 [1]

Nivelul de apă din fiecare bazin este măsurat cu ajutorul traductoarelor de nivel LT1 şi LT2. Senzorii de nivel LS1 şi LS2 trimit semnale doar în cazul umplerii la un nivel prea mare al bazinelor. Senzorul LS monitorizează atingerea unui nivel prea mic în bazinul de alimentare pentru a preveni funcţionarea în gol a pompei P.

Pompa P are rolul de a distribui apa din bazinul de alimentare în sistem, de-a lungul conductei mediane. Presiunea şi debitul de apă pompat pot fi controlate prin ajustarea frecvenţei.
Conducta verticală se ramifică în două în partea de sus, iar presiunea fluidului este monitorizată cu ajutorul traductoarelor de presiune PT1 şi PT2. Pe cele două ramificaţii orizontale, traductoarele de debit FT1 şi FT2 măsoară debitul care intră în cele două bazine. Acest debit este controlat (variat) prin intermediul electro-valvelor FC1 şi FC2, controlate în tensiune.

Sistemul de pompare este completat de alte două pompe auxiliare - P1 şi P2 - prin intermediul cărora se poate adăuga apă la debit constant în cele două bazine. Acestea pot fi folosite pentru a introduce perturbaţii în sistem.

Astfel, procesul instalaţiei ASTANK2 poate fi considerat un proces multivariabil (de tip MIMO - Multiple Input Multiple Output) cu 3 intrari (tensiunile aplicate pe cele doua electro-valve, tensiunea aplicatţpompei principale) si 2 ieşiri(nivelurile din cele 2 bazine superioare) ¸ Pentru un caz extins, pe care nu il voi aborda in aceasta lucrare, se pot considera 5 parametri de control, adica 5 marimi de intrare (tensiunea

electro-valvelor, tensiunea aplicată pompei principale şi deschiderea sau închiderea robinetelor pompelor secundare).

2.2 Configuraţia instalaţiei

După cum a fost menţionat în (2.1), există multe configuraţii posibile ale instalaţiei, în funcţie de acţionarea asupra robineţilor precizaţi. De exemplu, setarea robinetului **RC** conduce la configuraţii şi implicit la dinamici diferite. Dacă acesta este deschis, procesul de umplere-golire a rezervoarelor (în buclă deschisă) devine mai lent, adică stabilizarea nivelurilor în cele două rezervoare se face după un timp îndelungat (cca 20-30 minute). De asemenea, robinetele **RP1** şi **RP2** pot fi folosite pentru introducerea de perturbaţii în proces, în mod independent în fiecare bazin în parte.

În lucrarea de faţă a fost aleasă următoarea configuraţie:

- Toate robinetele sunt deschise de tot, robinetele de evacuare ale celor două bazine (**R11** şi **R12** fiind singurele deschise la 50 %).
- Pompele auxiliare nu au fost folosite ca intrări, acestea fiind setate pe **OFF**.
- Pompa principală alimentată la **7-9V**.
- Pentru setul de date folosit la identificare, tensiunile celor 2 electro-valve au fost variate între $\simeq \mathbf{5}\boldsymbol{V}$ şi $\simeq \mathbf{10}\boldsymbol{V}$, iar pompa principala a fost alimentata la tensiunea U=9V.

Conform acestei configuraţii, procesul este unul multivariabil cu 3 intrări şi două ieşiri. Una dintre cele 3 intrări (pompa principală) este menţinută la valoare constantă, pe când cele două electro-valve vor putea fi acţionate în tensiune.

2.3 Modelarea dinamicii procesului

Modelul analitic dezvoltat pentru instalaţia ASTANK2 se poate dovedi inadecvat pentru simulare din cauza următoarelor limitări practice [1]:

- Procesul de curgere prin conductele superioare orizontale este afectat de perturbaţii, din cauza lungimii scurte.
- Debitele de apă pompate pe cele două căi laterale, prin ajustarea tensiunilor celor două electro-valve FC1 şi FC2 nu se distribuie neapărat proporţional cu valorile acestor tensiuni. De exemplu, pentru diferenţe mai mari de 4V între cele 2 tensiuni, s-a constatat, pe baza experimentelor, ca unul dintre bazine nu va fi alimentat aproape deloc. Pentru a construi un model adecvat trebuie să se ţină cont de toate combinaţiile de tensiuni ale celor două electro-valve.
- Inversarea valorilor celor două tensiuni nu conduce neapărat la inversarea valorilor debitelor.

- Ajustarea tensiunilor celor 2 electro-valve trebuie să se facă cu evitarea salturilor mai mari de 3V, acestea putând duce la defectarea electro-valvelor.

3 Identificarea parametrilor instalaţiei

Această parte a lucrării este dedicată identificării experimentale a unui model matematic adecvat şi valid, care să aproximeze cât mai bine procesul de umplere-golire din instalaţia ASTANK2. Astfel, se doreşte găsirea unui **model de identificare**, adică a unei **relaţii matematice** intrare-ieşire care să descrie cu acurateţe caracteristicile şi funcţionarea procesului. De regulă, în abordarea identificării experimentale, procesul este văzut ca o **cutie neagră**, ce furnizează semnale de reacţie (ieşire) la stimularea cu diferite semnale de comandă. Seturile de date intrare-ieşire furnizate din proces printr-un experiment econometric (achiziţie si măsurare) constituie baza construirii unui model matematic experimental (model de identificare) al procesului. Desigur, modelul este identificat (estimat) cu ajutorul unei metode de identificare aplicate unei clase de modele de identificare stabilite.

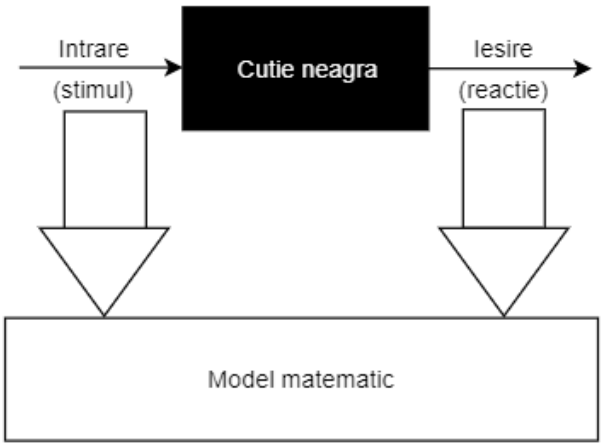

Figure 3.1: Cutie neagra

3.1 Formularea problemei de identificare

Aşadar, problema identificării experimentale poate fi formulată astfel [9]: având un proces cu structura necunoscută şi parametri necunoscuţi, se doreşte construcţia şi determinarea unui model matematic intrare-ieşire pe baza datelor achiziţionate din proces, astfel încât modelul să descrie cât mai fidel comportamenul procesului în sensul unui criteriu de adecvanţă. Parametrii modelului de identificare nu au semnificaţie fizică, ci sunt meniţi să descrie doar relaţia intrare-ieşire a procesului. Cu toate că modelul matematic poate reflecta comportamentul procesului în jurul unui anumit punct de funcţionare, deci poate avea generalitate limitată, avantajul major al identificării experimentale îl constituie implementabilitatea unor metode/algoritmi de identificare, aplicate în scopul obţinerii modelului matematic .

Construcţia modelelor de identificare se bazează pe datele experimentale furnizate de cutia negră în urma stimulării acesteia cu semnale ce au valori situate în jurul punctului de funcţionare al procesului. Astfel, procesul necunoscut, care poate fi analizat doar prin prisma dependenţei intrare-ieşire a acestuia, trebuie aproximat cu un model implementabil numeric care să reflecte caracteristicile dinamicii procesului real. Acest model este estimat prin minimizarea unei funcţii criteriu care asigură apropierea dintre modelul simulat şi procesul real.

Astfel, modelul obţinut va fi un **model experimental**. Modelele experimentale se bazează pe algoritmi inspiraţi din teoria optimizării şi teoria estimaţiei, preluând avantaje din ambele (implementabilitate, respectiv caracterizare statistică sau verificarea unor proprietăţi statistice a parametrilor modelului).

Etapele preliminare ale unui experiment de identificare sunt ([9]):

- Se studiză informaţiile preliminare ale procesului
  - Organizarea experimentului econometric
  - Achiziţia şi prelucrarea primară a datelor
- Alegerea clasei de modele de identificare
- Alegerea modelului de identificare
- Alegerea metodei de identificare

Bucla principală a experimentului de identificare constă în evaluarea comparativă a unor modele de structuri diferite (combinaţii de indici structurali ai modelului) pe baza unui criteriu de adecvanţă şi alegerea structurii optime care optimizează acest criteriu. Căutarea modelului optimal se poate face fie exhaustiv, fie printr-o metaeuristică. Rezultatul acestui procedeu este un model matematic adecvat şi valid al procesului.

De regulă, ultima etapă a unui experiment de identificare este validarea modelului, care se bazează pe paşii următori:

1. Atât procesul cât şi modelul vor fi stimulate cu acelaşemnal de stimul.
2. Se măsoară ieşirea procesului
3. Se evaluează ieşirea modelului (ieşirea simulată)
4. Se evaluează eroarea de predicţie, ca diferenţa dintre rezultatele de la **2** şi **3** şi se aplică teste de albire (de validare). Dacă eroarea de predicţie are caracteristicile unui semnal de tip zgomot alb, atunci se poate spune că modelul identificat este valid.

3.2 Modele parametrice

Modelele matematice de identificare sunt modele parametrice (sau numerice) [10], în care conceptul central este **parametrul** (de regulă, fără semnificaţie fizică). Clasa cea mai uzuală de modele parametrice intrare-ieşire liniare este clasa ARMAX (sau mai generală, RSISO). În cadrul acestei lucrări se doreşte identificarea unui **model parametric** din clasa **RSISO**.

Un model intrare-iesire poate fi scris sub forma generală:

$$y[n] = q^{-nk}G(q^{-1})u[n] + H(q^{-1})e[n] \tag{3.1}$$

Unde: **u[n]** şi **y[n]** reprezintă valorile intrării şi ieşirii procesului/modelului la un moment de timp n, **e[n]** este perturbatia care afecteaza datele masurate (zgomot de măsură), considerat ca zgomot alb de medie nulă şi dispersie unitară. **nk** este întârzierea sistemului (având, în mod uzual, valoarea 1). $\boldsymbol{G(q^{-1})}$ este funcţia de sistem a părţii utile a modelului (dinamica intrare-ieşire), iar $\boldsymbol{H(q^{-1})}$ este funcţia de sistem a părţii nedeterministe (filtrul de zgomot).

Modelele din clasa RSISO (3.2) extind modelele din clasa ARMAX datorită posibilităţii de a preciza poli diferiţi filtrelor de sistem şi de zgomot. Forma lor este :

$$\begin{cases} A(q^{-1})y[n] = \frac{B(q^{-1})}{F(q^{-1})}u[n] + \frac{C(q^{-1})}{D(q^{-1})}e[n] \\ E\{e[n]e[m]^T\} = \lambda^2\delta(n-m) \end{cases} \tag{3.2}$$

Unde:

$$\begin{cases} A(q^{-1}) = 1 + a_1q^{-1} + ... + a_{na}q^{-na} \\ B(q^{-1}) = 0 + b_1q^{-1} + ... + b_{nb}q^{-nb} \\ F(q^{-1}) = 1 + f_1q^{-1} + ... + f_{nf}q^{-nf} \\ C(q^{-1}) = 1 + c_1q^{-1} + ... + c_{nc}q^{-nc} \\ D(q^{-1}) = 1 + d_1q^{-1} + ... + d_{nd}q^{-nd} \end{cases} \tag{3.3}$$

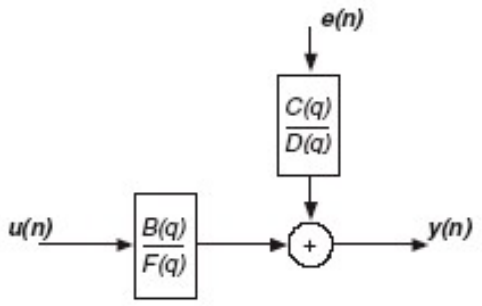

Figure 3.2: Structură Box-Jenkins

Prin particularizarea polinoamelor din modelul general RSISO se pot obţine diferite modele particulare:

- **ARMAX** - AutoRegresiv cu Medie Alunecătoare şi control eXogen (3.3): în acest caz polinoamele $F(q^{-1})$ şi $D(q^{-1})$ sunt egale cu 1, funcţia de sistem a părţii utile şi cea a părţii stocastice având poli identici, daţi de polinomul $\boldsymbol{A(q^{-1})}$. **ARMAX** este o subclasă clasei RSISO.

$$A(q^{-1})y[n] = B(q^{-1})u[n] + C(q^{-1})e[n]$$

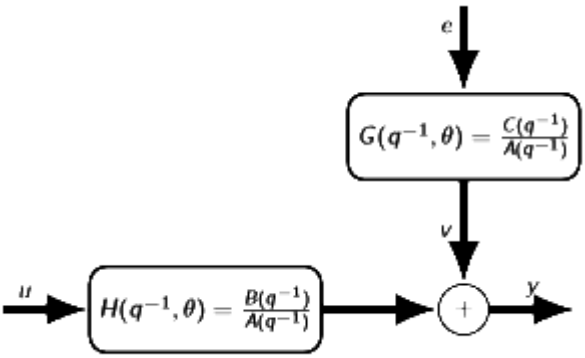

Figure 3.3: Structură ARMAX

- **ARX** - AutoRegresiv cu control eXogen (3.4): caz particular al clasei ARMAX, în care $\boldsymbol{C(q^{-1}) = 1}$

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

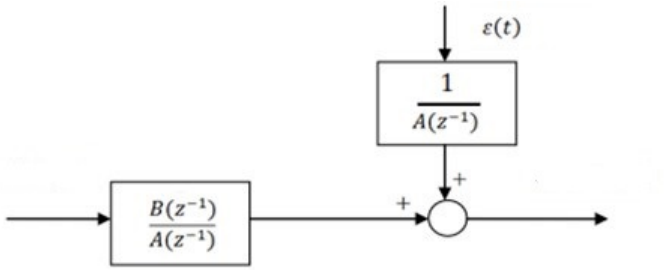

Figure 3.4: Structură ARX

Modelul **ARX** este destinat aplicatiilor de control, reprezentand atat dinamica deterministă cat şi cea stocastică a procesului.

- **ARMA** - AutoRegresiv de Medie Alunecătoare (3.5): caz particular al clasei ARMAX, în care $\boldsymbol{u[n] = 0}$

$$A(q^{-1})y[n] = C(q^{-1})e[n]$$

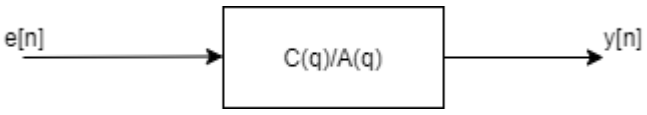

Figure 3.5: Structură ARMA

Modelul **ARMAX** este util în modelarea filtrelor de zgomot (perturbaţiilor, ca zgomot colorate).

Un model parametric este determinat de un anumit număr de parametri necunoscuţi ce trebuie determinaţi.

Este necesară determinarea atât a valorilor parametrilor necunoscuţi cât şi a numărului lor.

În această lucrare se urmăreşte determinarea unor modele parametrice optime din clasa RSISO. Presupunând iniţial cunoscută structura modelului, identificarea valorilor parametrilor modelului se va realiza folosind procedurile cunoscute de optimizare (Metoda celor ai mici pătrate MCMMP pentru modele ARX, Metoda minimizării erorii de predicţie MMEP, pentru modele ARMA si ARMAX). Avand in vedere ca structura unui model poate varia, valorile **indicilor structurali** optimali ai modelului vor fi determinate cu ajutorul metaeuristicilor. Aşadar, modelul optimal va fi obţinut prin implementarea unor metaeuristice descrise in capitolul următor.

Structura modelului ales de mine are în componenţă un filtru pentru partea utilă (model ARX), respectiv un filtru de zgomot (model ARMA), conform ecuaţiilor de mai jos (3.4)-(3.10). Procesul multivariabil se va modela prin intermediul a 2 modele de tip MISO, aferente nivelurilor de lichid din cele două rezervoare.

Partea utilă a modelului ce trebuie identificat este ilustrată în figura (3.6):

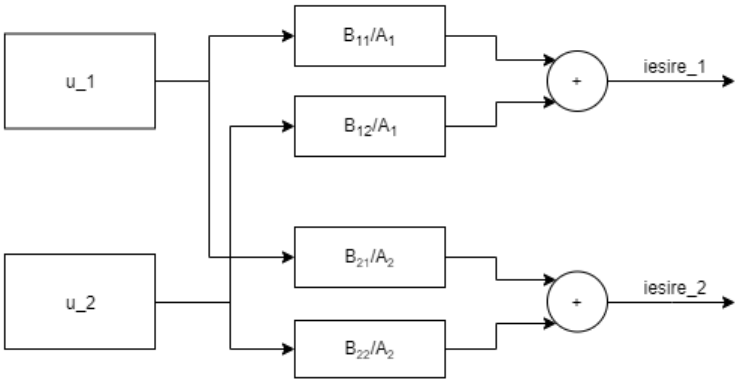

Figure 3.6: ARX MIMO

$$A(q^{-1})y[n] = B_1(q^{-1})u_1[n] + B_2(q^{-1})u_2[n] + v[n] \tag{3.4}$$

$$\begin{cases} A(q^{-1}) = 1 + a_1q^{-1} + ... + a_{na}q^{-na} \\ B_1(q^{-1}) = q^{1-nk_1}\left(b_{1,1}q^{-1} + b_{2,1}q^{-2} + ... + b_{nb_1,1}q^{-nb_1}\right) \\ B_2(q^{-1}) = q^{1-nk_2}\left(b_{1,2}q^{-1} + b_{2,2}q^{-2} + ... + b_{nb_2,2}q^{-nb_2}\right) \end{cases} \quad \forall n, m \in \overline{1, N} \tag{3.5}$$

Ieşirea simulată a modelului util, y_{ARX} este (se presupun cunoscute gradele polinoamelor din 3.5:

$$\begin{aligned} y_{ARX}[n] = &-\hat{a}_1 y_{ARX}[n-1] - \hat{a}_2 y_{ARX}[n-2] - ... - \hat{a}_{na} y[n-na] + \hat{b}_{1,1}u_1[n-nk_1] + ... \\ &+ \hat{b}_{nb_1,1}u_1[n-nk_1-nb_1+1] + \hat{b}_{1,2}u_2[n-nk_2] + ... \\ &+ \hat{b}_{nb_2,2}u_2[n-nk_2-nb_2+1] \qquad \forall \in \overline{1, N} \end{aligned} \tag{3.6}$$

Se iniţializează $y_{ARX}[n] = y[n], n \in \overline{1, \max\{na, nb_1, nb_2\}}$.

Unde A, B_1 şi B_2 sunt polinoame de grade na, nb_1 şi nb_2. Pentru fiecare dintre cele două ieşiri, zgomotul poate fi identificat cu ajutorul unui model ARMA:

$v_r[n] = y[n] - y_{ARX}[n], \forall n \in \overline{1, N}$ reprezintă zgomotul colorat rezidual ce trebuie identificat.

$$\begin{cases} v_r[n] = \frac{C(q^{-1})}{D(q^{-1})}e[n] \\ E\{e[n]e[m]\} = \lambda^2\delta_0[n-m] \end{cases} \quad \forall n, m \in \overline{1, N} \tag{3.7}$$

Unde e este zgomot alb Gaussian de medie nulă şi dispersie necunoscută λ^2, iar C şi D sunt polinoame de grade nc şi nd.

$$\begin{cases} C(q^{-1}) = 1 + c_1q^{-1} + ... + c_{nc}q^{-nc} \\ D(q^{-1}) = 1 + d_1q^{-1} + ... + d_{nd}q^{-nd} \end{cases} \tag{3.8}$$

Modelul ARMA poate fi identificat cu MMEP, ulterior estimându-se zgomotul colorat v_{ARMA}.

$$\begin{cases} v_{ARMA}[n] = -\hat{d}_1 v_{ARMA}[n-1] - ... - -\hat{d}_1 v_{ARMA}[n-nd] + c_1\hat{e}[n-1] + ... + \hat{c}_{nc}[n-nc] \\ \hat{e}[n] = v_r[n] + \hat{\alpha}v_r[n-1] + ... + \hat{\alpha}v_r[n-n\alpha] \end{cases} \tag{3.9}$$

Unde ê este estimarea unui model AR de ordin $n\alpha$, setat mult mai mare decât max{nc,nd}.

Se iniţializează $v_{ARMA}[n] = v_r[n], \forall n \in \overline{1, max\{nc, nd\}}$.

În final eroarea de predicţie (ce trebuie minimizată) este:

$$\epsilon[n] = v_r[n] - v_{ARMA}[n], \forall n \in \overline{1, N} \tag{3.10}$$

3.3 Funcţia criteriu

Presupunând stabilită structura modelului ce trebuie identificat, se poate trece la optimizarea setului de indici structurali. Această problemă se va formula ca o problemă granulară de optimizare. Pentru fiecare dintre cele 2 ieşiri ale procesului trebuie identificat modelul optim, care maximizează o funcţie criteriu. Formularea acestei funcţii are o deosebită importanţă în procesul de identificare a numărului optim de parametri pentru polinoamele modelului structural. Valorile optime ale parametrilor vor fi calculate cu ajutorul metodei MMEP.

O posibilă formulare a funcţiei criteriu este calcularea sumei patratelor erorilor de predicţie la fiecare moment de eşantionare

$$f_{criteriu} = \sum_{i=1}^{n} \left(y_{real}(i) - y_{simulat}(i)\right)^2 \tag{3.11}$$

Pentru a obţine direct un model valid, funcţia criteriu folosită la optimizarea setului de indici structurali din această lucrare constă în calcularea raportului semnal-zgomot atât pentru datele de identificare cât şi pentru cele de validare. Menţionez că setul de măsurători a fost partajat în două subseturi: pentru identificare şi pentru validare, după cum va fi explicat in secţiunea (6).

Se obţine un criteriu de cost compus, de forma:

$$\boldsymbol{f_{criteriu} = \alpha f_{ident} + (1-\alpha) f_{valid}} \tag{3.12}$$

Unde, $\boldsymbol{f_{ident}}$ reprezintă SNR-ul aferent setului de identificare, calculat ca raportul dintre norma semnalului de ieşire măsurat staţionarizat şi norma erorii de predicţie (zgomot colorat) - diferenţa dintre ieşirea măsurată şi ieşirea simulată.

Identic, în cazul lui $\boldsymbol{f_{valid}}$, se ţine cont de ieşirea măsurată staţionarizată în urma folosirii setului de validare. În acest caz, eroarea de predicţie se calculează ca diferenţa dintre date de ieşire de validare şi date simulate cu modelul de identificare estimat.

$$f_{ident} = dB\left(\frac{||y_{identificare}||}{||err_{identificare}||}\right) \tag{3.13}$$

$$f_{valid} = dB\left(\frac{||y_{validare}||}{||err_{validare}||}\right) \tag{3.14}$$

$\boldsymbol{\alpha \in (0,1]}$ - factor de ponderare a celor două seturi. Dacă se doreşte ca unul din seturi (de regulă, de identificare) să aibă ponderea mai mare, atunci se alege $\boldsymbol{\alpha > 0.5}$ Evident, ponderile celor două seturi, sunt complementare.

4 Optimizare metaeuristica

După cum am precizat anterior, în momentul identificării valorilor optime ale coeficienţilor modelului, se consideră cunoscute valorile indicilor structurali. Valorile lor optimale vor fi găsite cu ajutorul **metodelor metaeuristice** [7], care pot asigura un compromis bun între viteză şi precizie. Căutarea exhaustivă pe un spaţiu de căutare fractal este ineficientă.

4.1 Metaeuristici

Termenul *euristică* în optimizare originează din Grecia Antică, unde *heuriskein* înseamnă "a descoperi". Există două metode de căutare a unui optim într-un spaţiu dat:

- Se poate efectua o căutare **exhaustivă**, în care fiecare tot spaţiul este evaluat şi se va găsi, într-un timp destul de lung optimul global.
- Cătuare **euristică**, în care doar o mică parte din spaţiu este evaluată (în mod strategic), încercându-se găsirea unui individ suficient de bun, cât mai apropiat de optim, cu avantajul reducerii semnificative a timpului de cŭatare.

Problema de optimizare se va defini în contextul unei funcţii de cost(criteriu):

$$f : \boldsymbol{S} \rightarrow \boldsymbol{R}$$

unde, spaţiul de căutare $\boldsymbol{S}$ este un subset mărginit al $\boldsymbol{R^{nx}}$.

Funcţia criteriu $\boldsymbol{f}$ se remarcă prin variaţiile sale care împart spaţiul de căutare în multe subspaţii cu extreme proprii, astfel existând multe extreme locale. Este aproape imposibilă evaluarea derivatelor lui $\boldsymbol{f}$, de multe ori chiar neexistând.
Problema poate fi relaxată prin căutarea numai într-un subspaţiu $\boldsymbol{D}$ optimul:

$$opt_{x \in \boldsymbol{D} \subset \boldsymbol{S}} f$$

Rezolvarea problemei devine echivalentă cu rezolvarea unei probleme granulare de optimizare, în care se va încerca găsirea particulei din $\boldsymbol{D}$ cel mai apropiat de optimul global descris de $\boldsymbol{f}$.

Principalul obiectiv al tehnicilor bazate pe euristică este dezvoltarea unor mecanisme de rezolvare care să evite căutarea exhaustivă, putând fi, totodată construiţi algoritmi care să poată fi implementaţi pe calculator. Prin evitarea căutării exhaustive, gasirea optimului global nu va fi garantata, însă se încearcă reducerea semnificativă a timpului de căutare, păstrând o acurateţe suficient de bună.

Metodele euristice care pot fi implementate pe sisteme de calcul sunt cunoscute sub denumirea de **metaeuristică**. Ele se bazează pe principiul următor: căutarea optimului simulează comportamentul unui sistem biologic sau evoluţia unui fenomen

natural. Astfel, o nouă ramură a optimizării s-a dezvoltat recent, numită **Programare Evoluţionară**.

În general, toate metodele metaeuristice folosesc generatoare pseudo-aleatoare pentru a selecta anumiţi parametri sau anumite operaţii care conduc la estimarea soluţiei optime. În cazul secvenţelor pseudo-aleatoare (**PRS**), densitatea lor de probabilitate ar trebui să fie specificată a priori. Există două tipuri de densităţi de probabilitate utilizate:

- uniformă sau normală: generator de secvenţe pseudo-aleatoare cu probabilitate uniformă
- generator de secvenţe pseudo-aleatoare cu probabilitate prescrisă.

Există două categorii de metaeuristice: **locală** şi **globală**. În cadrul metodelor **locale** se presupune existenţa unui singur optim global sau căutare restrânsă la un singur subset care include optimul global. Scopul metodelor **globale** este găsirea optimului global sau aproximarea cât mai bună a sa prin cătuarea în diferite zone ale lui $\boldsymbol{S}$. De regulă, pentru a evita blocajele în optime locale, se folosesc schimbări bruşte ale subsetului explorat.

4.2 Algoritm bazat pe licurici

Licuricii sunt insecte mici zburătoare, ce fac parte din clasa Coleoptera. Pe timpul nopţii, ele emană semnale de lumină în scopul atracţiei mutuale. Algoritmul de optimizare bazat pe comportamentul licuricilor a fost introdus de Xin-She Yang, fiind inspirat de atracţia reciprocă dintre insecte şi fenomenul de atenuare naturală a luminii în funcţie de distanţă.

Licuricii sunt grupaţi într-un roi şi lăsaţi să exploreze un spaţiu de căutare, în interioul căruia vor interacţiona prin intermediul semnalelor luminoase receptate de la alţi indivizi ai roiului. Cu cât este mai strălucitor un licurici, cu atât acesta va putea fi capabil sa atragă mai mulţi indivizi. Intensitatea decreşte odată cu mărirea distanţei în cadrul spaţiului de căutare.

Zborul licuricilor este parţial aleator, iar strălucirea acestora este direct proporţională cu valorile exprimate prin criteriul de performanţă. Astfel, dacă în cadrul problemei de optimizare este căutat un maxim global, atunci, licuriciul cel mai strălucitor va fi cel care va da soluţia.

Algoritmul bazat pe licurici [7] are tendinţa de a se termina având o populaţie finală concentrată în jurul unuia sau chiar mai multor indivizi. Algoritmul efectuează o căutare paralelă, astfel în cazul anumitor spaţii de căutare populaţia se poate concentra în mai multe zone. Algoritmul tinde să pună accentul pe explorare pe tot parcursul căutării, utilizatorul neputând controla prea bine echilibrul dintre exploatare şi explorare.

Algoritmul bazat pe licurici are următorii paşi principali:

- Definirea spaţiului de căutare, a funcţiei criteriu(ce trebuie maximizată sau minimizată) şi configurarea parametrilor (mărimea populaţiei, numărul maxim de iteraţii, densitatea de probabilitate a direcţiilor perturbatoare, factorul de atracţie);
- Licuricii vor fi distribuiţi în mod uniform în cadrul spaţiului de căutare;
- Se calculează intensitatea luminoasă a fiecărui licurici;
- Se găseşte cel mai strălucitor licurici şi se iniţializează soluţia optimă;

Mai departe, se vor urma paşii conform diagramei de mai jos:

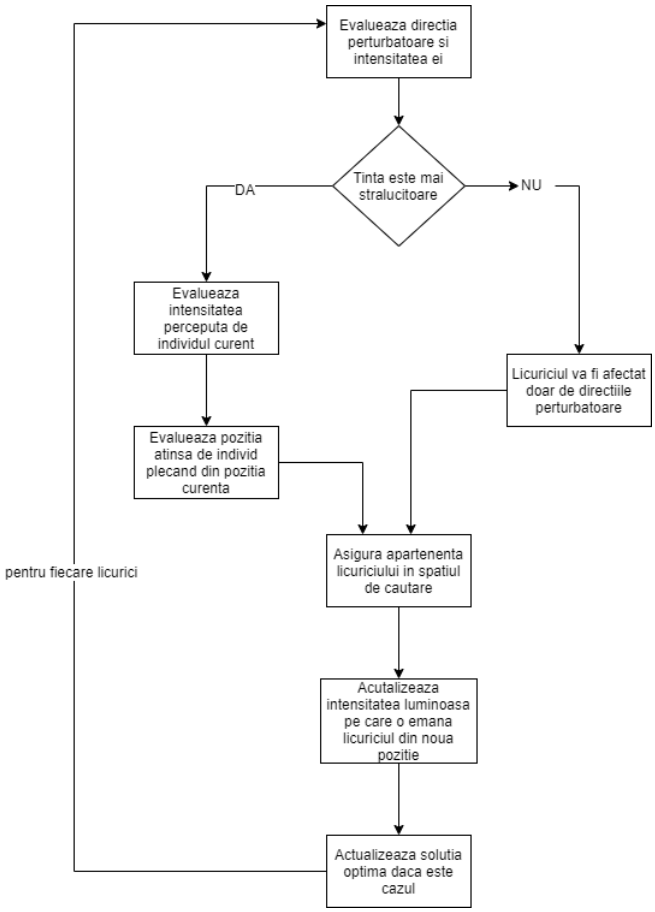

Figure 4.1: Schema de principiu

Modelul dinamic al comportamentului licuricilor este:

$$x_p^{k+1} = x_p^k + \beta_{p,q}^k(x_q^k - x_p^k) + \alpha^k sign(\rho^k - \frac{1}{2})\zeta^k(\phi_s) \tag{4.1}$$

Unde:

- $\beta_{p,q}^k$ - intensitatea luminoasă a licuriciului q percepută de licuriciul p la iteraţia k
- $\alpha^k \in [0,1]$ - număr general aleator cu densitate de distribuţie uniformă, care exprimă influenţa celorlalte surse luminoase asupra licuriciului curent în timpul zborului său
- $\rho^k \in [0,1]$ este un număr aleator folosit pentru generarea sensului direcţiei perturbatoare, în cadrul $sign(\rho^k - \frac{1}{2})$
- $\zeta^k(\phi_s) \in \boldsymbol{R}^{nx}$ - vector setat aleator (de distribuţie neuniformă), care reprezintă direcţia perturbatoare rezultată din lumina împrăştiată de ceilalţi licurici.
- ϕ_s - parametru proporţional cu dimensiunea spaţiului de căutare.

Pe baza principiului atracţiei luminii percepute, licuriciul $p \in \overline{1,P}$ zboară către licuriciul mai luminos $q \in \overline{1,P}$, poziţia sa fiind evaluată pe baza ecuaţiei (4.1). Expresia intensităţii luminoase, $\boldsymbol{\beta_{p,q}^k}$ depinde de distanţa dintre 2 licurici, exprimată prin $\boldsymbol{d(x_p^k, x_q^k)}$, care este distanţa euclidiană:

$$d(x_p, x_q) = \parallel x_p - x_q \parallel, \forall x_p, x_q \in \boldsymbol{S}$$

. Intensitatea luminoasă scade odată cu creşterea distanţei. Foarte important în cadrul algoritmului este modelul folosit pentru a exprima intensitatea luminoasă percepută de licurici. Un model propus de biologii care au studiat comportamentul licuricilor este următorul:

$$\beta_{p,q} = I_0 exp[-\gamma d^2(x_p, x_q)] = I_0 exp\left[-\gamma \parallel x_p - x_q \parallel^2\right], \forall x_p, x_q \in S \qquad (4.2)$$

În ecuaţia (4.2), I_0 reprezintă intensitatea luminoasă relativă ideală, în condiţiile unei distanţe nule dintre 2 licurici. Se va alege valoarea $\boldsymbol{I_0 = 1}$.

Parametrul $\boldsymbol{\gamma}$ este unul de importanţă majoră, acesta reprezentând un factor cheie în influenţarea vitezei de convergenţă. Deşi $\boldsymbol{\gamma}$ poate varia între $[0, \infty]$, se va alege intervalul $[0, \phi_s]$, unde $\boldsymbol{\phi_s = sup_{x,y \in S} d(x,y)}$. Adică, se va ţine cont de diametrul spaţiului de căutare. Acesta poate fi, însă, neregulat în dimensiune, spre exemplu $\boldsymbol{S} = [0,20] \times [0,50] \times [0,80]$. Astfel se va ţine cont pe fiecare axă în parte de intervalul în care pot varia parametrii.

Acest parametru trebuie ales cu grijă, deoarece el determină influenţa celorlalţi licurici asupra licuriciului curent. În cazul unei valori prea mari, spre $\boldsymbol{\infty}$, algoritmul cu licurici devine asemănător Monte-Carlo.

În timpul zborului spre licuriciul mai strălucitor, licuriciul curent poate percepe alte surse de lumină, de la alţi licurici şi îşi poate schimba direcţia în funcţie de intensitatea luminoasă acestora. Puterea acestor influenţe este cuantificată cu ajutorul factorului $\boldsymbol{\alpha^k \in [0,1]}$. Cu cât mai mare acest termen, cu atât mai predispus licuriciul la schimbarea direcţiei. Direcţia perturbatoare este reprezentată prin $\zeta^k \in [0, \phi_s]^{nx}$.

Componentele vectorului ce dau direcţia perturbatoare sunt selectate aleator, pe baza unei distribuţii de probabilitate $\boldsymbol{p}$, în funcţie de comportamentul roiului. Componentele vor fi determinate în intervalul $[0, \sigma_i]$, cu probabilităţi determinate de $\boldsymbol{p}$.

O posibilă distribuţie de probabilitate, care se potriveşte bine comportamentului licuricilor, se numeşte distribuţia lui Levy:

$$\boldsymbol{p}(u) = u^{-\lambda}, \forall u \in \boldsymbol{R} \tag{4.3}$$

Cu valoarea recomandată pentru $\boldsymbol{\lambda = 1.5}$

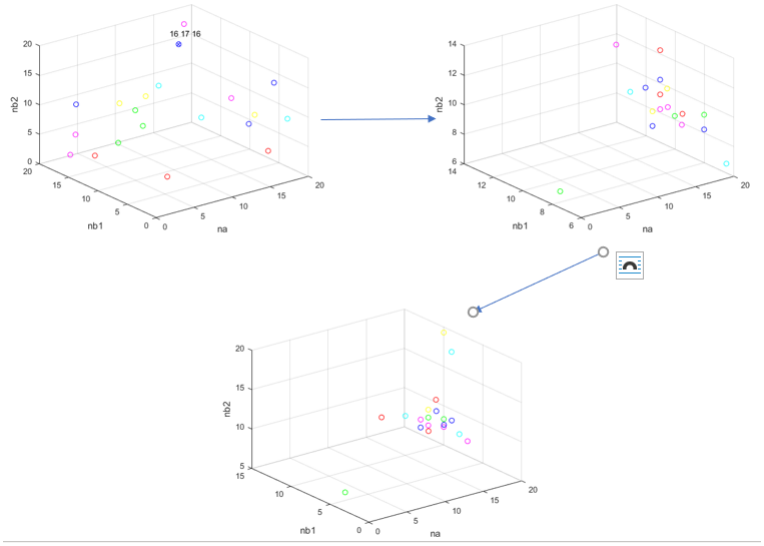

Figure 4.2: Poziţia licuricilor

Se doreşte evitarea numărului mare de indivizi care trebuie făcuţi valizi (readuşi în spaţiul de căutare în cazul în care au picat în afară). Dacă acest procedeu trebuie aplicat de prea multe ori, atunci procedura poate deveni asemănătoare unui proces de căutare Monte-Carlo [7]. Se pierde strategia de căutare, licuricii începând să aibe o mişcare browniană.

Licuricii nu pot fi blocaţi în aceeaşi poziţie indiferent de strălucire, un licurici putând să părăsească chiar şi cea mai bună poziţie întâlnită. Astfel, soluţia optimă nu este neapărat indicată de poziţia finală licuricilor, ci mai degrabă, este o soluţie întâlnită de unul din licurici pe drumul său.

La finalul algoritmului licuricii tind să se aglomereze într-una sau mai multe zone ale spaţiului de căutare, după cum se poate observa şi în figura (4.2).

4.3 Algoritm genetic

Un algoritm genetic [7], prima dată menţionat de către John Holland în 1975 reprezintă o tehnică de simulare a proceselor naturale de evoluţie şi adaptare specifice sistemelor biologice.

Tehnica de optimizare bazată pe algoritmi genetici presupune imitarea unei populaţii şi simulează în mod naiv evoluţia acesteia de la o genereţie la alta în cadrul unui mediu, în cazul nostru, **spaţiul de căutare**. Algoritmul încearcă să emuleze ideea de evoluţie, astfel de la o generaţie la alta indivizi din ce în ce mai capabili să facă parte din populaţie. Capabilitatea unui individ este exprimată sub forma conceptului de fitness.

Conceptul de **fitness** este considerat măsura fertilităţii şi capacităţii a unui individ în cadrul populaţiei, acesta justificând prezenţa şi utilizarea individului în cadrul mecanismului de selecţie naturală.

În cadrul optimizării pe baza algoritmilor genetici este folosit principiul celui mai capabil supravieţuitor. De asemenea, se promovează ideea de schimb aleator de informaţie, construindu-se astfel un proces de evoluţie care foloseşte în acelaşi timp tehnici de explorare cât şi de exploatare a spaţiului de căutare.
În contextul algoritmilor genetici, populaţia este formată din indivizi care reprezintă fiecare în parte o posibilă soluţie a problemei granulare de optimizare ce trebuie rezolvată. Un cromozom este compus din din gene care pot lua diferite valori şi care pot fi grupaţi în alele. De exemplu, un cromozom poate fi format din gene care să reprezinte o codificare binară.

În (4.3) este prezentată structura generală a algoritmului genetic. Etapele principale vor fi descrise în cele ce urmează.

Operaţiile genetice, 3 la număr, sunt mecanismele prin care se poate explora sau exploata spaţiul de căutare. În urma operaţiilor de **încrucişare**, **mutaţie** şi **inversiune** se obţin indivizi noi din spaţiul de căutare.

Încrucişarea (4.4) este operaţia genetică care are loc între doi indivizi şi presupune schimbul de informaţie genetică între aceştia pentru a crea doi moştenitori.
Moştenitorii vor avea părţi din genele ambilor părinţi combinate sub diferite forme. De exemplu, cei doi indivizi implicaţi în operaţia de încrucişare pot schimba între ei părţi din codificarea lor genetică pe baza unor pivoţi şi lungimi. O generalizare a procesului de combinare este realizată cu ajutorul unei măşti care indică ce biţi vor fi schimbaţi între ei de către cei de părinţi, după cum este prezentat mai jos:

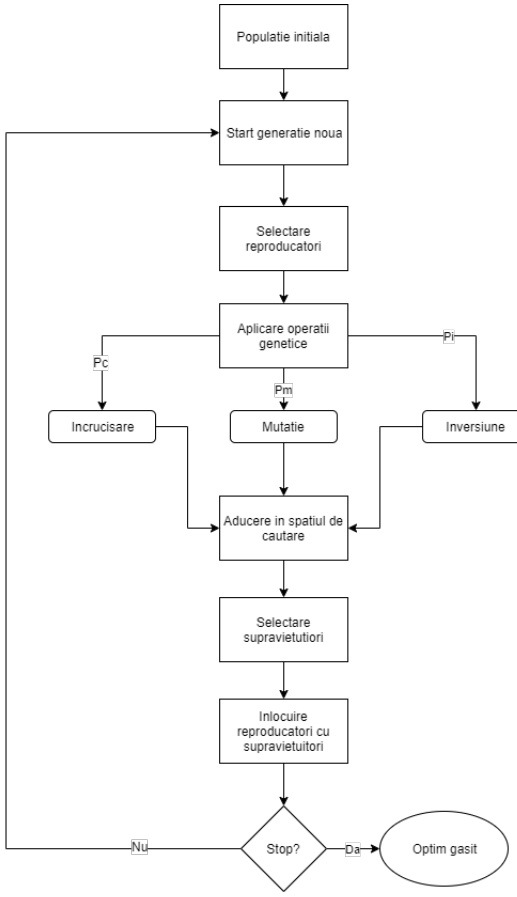

Figure 4.3: Structura generală a AG [7]

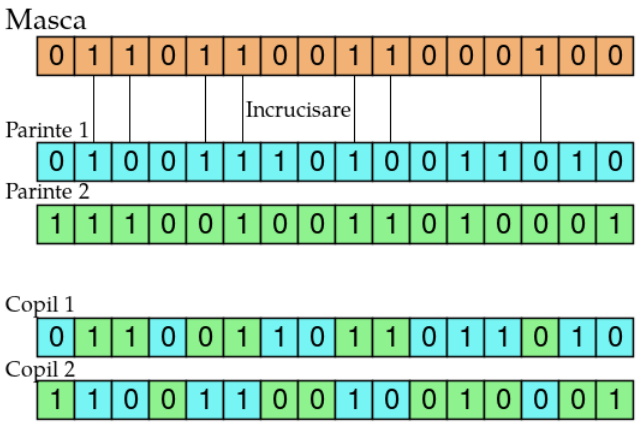

Figure 4.4: Operaţia de încrucişare

Mutaţia (4.5) reprezintă schimbarea aleatoare a unor alele din cadrul unui cromozom. Scopul acesteia este de păstra o oarecare diversitate în cadrul populaţiei,

moştenitorul rezultat fiind de cele mai multe ori mai puţin capabil decât părintele.

Mutaţia este, deci, unul din mecanismele care promovează procesul de explorare al spaţiului de căutare, prevenind împotmolirea procesului de căutare în zonele de maxim local. Pe de altă parte, un număr prea mare de mutaţii poate duce la o explorare prea intensă a spaţiului de căutare, rezultând într-un comportament oscilator al procesului de căutare.

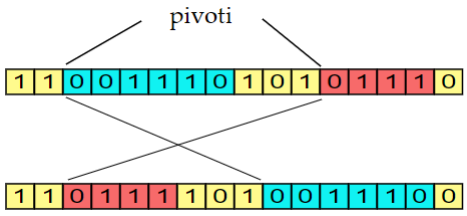

Figure 4.5: Operaţia de mutaţie

Inversiunea (4.6) este operaţia prin care o parte din cromozom îşi va schimba proprietăţile, astfel încât, pe baza unui pivot şi a unei lungimi, o parte din cromozom va fi inversată.

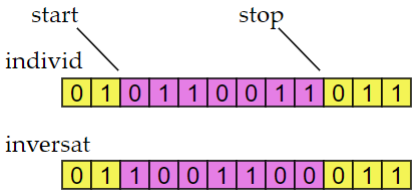

Figure 4.6: Operaţia de inversiune

Inversiunea se poate realiza prin schimbarea sensului secvenţei de interes din cromozom sau prin inversarea fiecărui bit din codificarea genetică a secvenţei, după cum se poate observa în (4.6).

În multe publicaţii operaţia de inversiune este considerată comună cu mutaţia, rezultatul acesteia fiind tot un mutant. Mutanţii au rolul foarte important de a preveni blocajul procesului de căutare şi aglomerarea în jurul unor puncte de optim local. **Încrucişarea** reprezintă motorul evoluţiei, prin această metodă obţinându-se moştenitori mai buni decât indivizii implicaţi în operaţie. Este necesară găsirea unui echilibru în utilizarea celor 3 operaţii în scopul evitării pe de o parte a blocajului în diferite subspaţii şi o evoluţie oscilatoare a populaţiei, pe de altă parte.

Un aspect foarte important de care trebuie ţinut cont în urma operaţiilor genetice menţionate este **viabilitatea** moştenitorilor. Se pot obţine indivizi care să

nu aparţină spaţiului de căutare prin aplicarea operaţiilor. Este necesară corectarea acestor indivizi şi aducerea lor înapoi în spaţiul de căutare valabil. Se pot aplica operaţii de tip *modulo* asupra genelor cromozomului pentru a ne asigura ca indivizii noi sunt viabili.

În cadrul algoritmilor genetici există două etape de selecţie: **selecţia reproducătorilor** şi **selecţia pentru supravieţuire**.

Selecţia reproducătorilor presupune alegerea din populaţia disponibilă în generaţia curentă a acelor indivizi asupra cărora se vor aplica operaţiile genetice. Maniera în care sunt aleşi reproducătorii poate fi decisivă cu privire la convergenţa algoritmului. O selecţie prea **strictă** poate duce la o populaţie dominată de indivizi din aceleaşi subspaţii, diversitatea fiind redusă. Dacă criteriul pe baza căruia se realizează selecţia este prea **permisiv**, populaţia devine puternic împrăştiată, iar evoluţia către optim devine prea lentă.

Există mai multe proceduri prin care se poate realiza selecţia reproducătorilor: selecţie după fitness, selecţie pe baza legii lui Boltzmann, selecţie pe baza clasamentului. În această lucrare se va folosi **selecţia prin competiţie** pentru alegerea indivizilor implicaţi în operaţiile de reproducere.
Paşii metodei de selecţie prin competiţie sunt următorii:

- Se alege un număr $N_r \leq N$ de poziţii vacante care vor fi puse la bătaie. Fiecare poziţie va avea un parametru $\boldsymbol{\eta \in [0.5, 1]}$ atribuit
- Pentru fiecare poziţie:
  - Se vor alege doi competitori în mod aleator din populaţie $x \in \boldsymbol{P}, y \in \boldsymbol{P}$
  - Se va calcula fitness-ul fiecărui competitor
  - Se va genera în mode aleator, pe baza unei distribuţii uniforme de probabilitate, un număr $\boldsymbol{v \in [0, 1]}$
  - Dacă $\boldsymbol{v \geq \eta}$, poziţia va fi ocupată de individul mai capabil. Individul mai slab va fi cel care va ocupa poziţia disponibilă în caz contrar.
  - După participarea la competiţie, cei 2 indivizi se întorc în populaţie pentru a putea fi aleşi din nou

Mai departe, în urma aplicării tuturor operaţiilor genetice asupra indivizilor selectaţi pentru reproducere, rezultă un grup prea mare de cromozomi care trebuie introduşi înapoi în populaţie. Mutaţia şi inversiunea se aplică asupra unui singur individ şi se obţine un moştenitor. Încrucişarea se realizează între doi părinţi, rezultând doi noi copii.

Prin urmare, se pune problema selectării indivizilor **supravieţuitori** din grupul astfel format. Există mai multe modalităţi de realizare a acestei selecţii, cea folosită în această lucrare fiind **selecţia elitist-generaţională**. Aceasta presupune selectarea supravieţitorilor direct din grupul format în urma aplicării operaţiei genetice, alcătuit

din indivizii reproducători şi moştenitorii lor.

În urma mai multor generaţii populaţia ar trebui să fie distribuită în câteva subspaţii de căutare în care se realizează exploatarea, păstrând, totodată, câţiva indivizi exploratori în diferite zone (4.7).

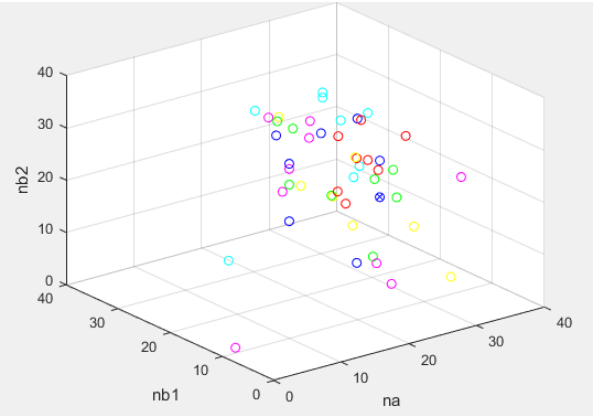

Figure 4.7: Populaţie AG

5 Control PID optimal metaeuristic

În această etapă se urmăreşte proiectarea unei legi de control în buclă închisă care să ducă la îndeplinirea unor condiţii de performanţă, anume asigurarea unor valori reduse ale suprareglajului şi subreglajului, odată cu reducerea timpului de creştere şi a timpului tranzitoriu. Aceşti indicatori de performanţă vor fi analizaţi în cazul răspunsului în buclă închisă la o referinţă de tip treaptă aplicată sistemului. Înainte de asigurarea acestor performanţe se doreşte ca eroarea staţionară să fie nulă. De asemenea, legea de control trebuie să permită rejecţia perturbaţiilor externe, fără solicitarea peste măsură a elementelor de acţionare.
Schema generală a unui sistem de control în buclă închisă este prezentată în (5.1).

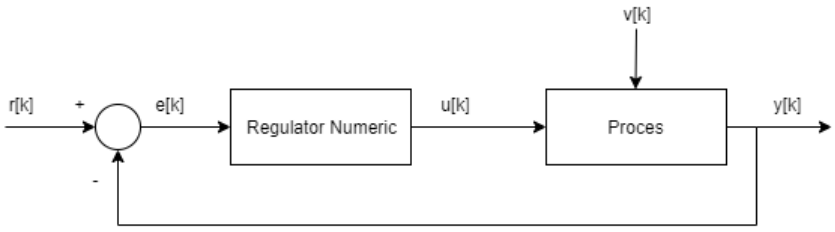

Figure 5.1: Schema de control în buclă închisă

Printre legile de control proiectate off-line se numără:

- legea de control de tip RST
- legea de control de tip PID - în varianta RST particularizat

5.1 Legea de control PID

Legea de control de tip PID [11] presupune calcularea la fiecare moment de eşantionare a valorii comenzii pe baza valorii erorii curente (componenta proporţională), erorii acumulate (componenta integrală) şi a ratei de modificare a erorii (componenta derivativă).

$$u_k = K_R \left[\epsilon_k + \frac{T}{T_i} \sum_{i=0}^{k} \epsilon_i + \frac{T_d}{T} (\epsilon_k - \epsilon_{k-1}) \right] + u_0 \tag{5.1}$$

Legea de control de tip **PID** poate fi scrisă ca o **particularizare** a legii de control de tip **RST**, în cadrul căreia polinoamele $\boldsymbol{R[z^{-1}]}$ şi $\boldsymbol{T[z^{-1}]}$ sunt egale.
Schema de principiu a unui sistem de reglare bazat pe legea de control de tip RST este prezentată în figura (5.2).

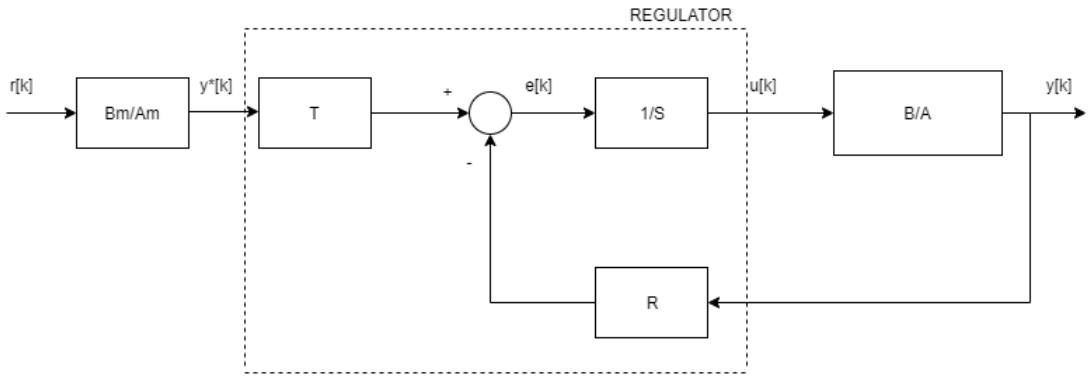

Figure 5.2: Schema de control RST

Schema de control PID obţinută prin particularizarea RST [8] este cea pentru care:

$$\begin{cases} R(z^{-1}) = T(z^{-1}) = r_0 + r_1 z^{-1} + r_2 z^{-2} \equiv R_0(z^{-1}) \\ S(z^{-1}) = (1 - z^{-1})(1 + s_1 z^{-1}) \equiv S_0(z^{-1}) \end{cases} \tag{5.2}$$

Astfel, schema de control din (5.2) devine, pentru cazul regulatorului de tip PID, echivalentă cu cea din (5.3).

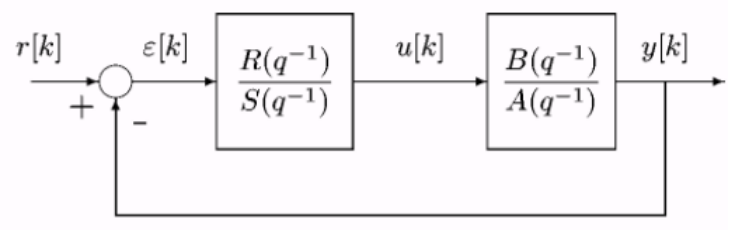

Figure 5.3: Schema de control PID

Formula legii de control **PID în varianta RST** este următoarea:

$$H_{PID}(s) \equiv K_R \left[1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} \right] \tag{5.3}$$

Prin discretizarea acesteia cu perioada de eşantionare $T = 1$, folosind **metoda dreptunghiului**, $\mathbf{\frac{1}{s} \approx \frac{T}{1 - z^{-1}}}$, legea de control devine:

$$H_{PID}(z^{-1}) = K_R \left[1 + \frac{\frac{T_e}{T_i}}{1 - z^{-1}} + \frac{\frac{NT_d}{T_d + NT_e}(1 - z^{-1})}{1 - \frac{T_d}{T_d + NT_e} z^{-1}} \right] = \frac{R_0(z^{-1})}{S_0(z^{-1})} \tag{5.4}$$

Care poate fi rescrisă sub forma:

$$H_{PID}(z^{-1}) = \frac{q_2 z^{-2} + q_1 z^{-1} + q_0}{p_2 z^{-2} + p_1 z^{-1} + p_0} \tag{5.5}$$

Unde:

$$\begin{cases} q_2 = K_R \left(T_i + \frac{T_d}{N} + \frac{T}{2} + \frac{2(N+1)T_i T_d}{NT} \right) \\ q_1 = K_R \left(T - \frac{4(N+1)T_i T_d}{NT} \right) \\ q_0 = K_R \left(\frac{2(N+1)T_d T_i}{NT} - T_i - \frac{T_d}{N} + \frac{T}{2} \right) \\ p_2 = T_i + \frac{2T_i T_d}{NT} \\ p_1 = \frac{-4T_i T_d}{NT} \\ p_0 = \frac{2T_i T_d}{NT} - T_i \end{cases} \tag{5.6}$$

5.2 Determinarea parametrilor PID

Coeficienţii celor două polinoame din (5.5) se calculează după cum este prezentat în (5.6). La rândul lor aceşti coeficienţi depind de valorile celor 3 necunoscute, $\boldsymbol{K_R, T_i, T_d}$, componentele proporţională, integratoare şi derivativă. Perioada de eşantionare, **T** şi constanta de filtrare, **N** nu sunt variabile. Astfel, calcularea regulatorului de tip PID care să asigure cel mai bun set de indicatori de performanţă, odată cu urmărirea referinţei şi rejecţia perturbaţiilor se poate realiza în mai multe feluri. Identificarea procesului se va face folosind modele din clasa ARMAX, destul de complexe. Prin urmare, utilizarea metodelor convenţionale de calculare a coeficienţilor regulatorului nu vor fi suficient de precise. Pentru a realiza optimizarea setului de parametri ai PID se va folosi algoritmul genetic prezentat în (4.3), cu modificările impuse de noul spaţiu de căutare.

Adaptarea algoritmului genetic la această problemă de optimizare impune următorii paşi:

- modificarea cromozomului - acesta trebuie să reţină valorile coeficienţilor regulatorului
- adaptarea conform cu pasul anterior a operaţiilor genetice
- modificarea funcţiei de cost pentru a exploata cerinţele de performanţă
- modificarea parametrilor de configurare ai algoritmului - spaţiul de căutare va fi semnificativ mai mare

Procesul fiind unul de tip MIMO cu două intrări, **cromozomul** va trebui să codifice informaţia pentru 2 regulatoare, deci vor exista 6 alele, câte 3 pentru fiecare regulator.

Fiecare alelă va reţine informaţia pentru un coeficient şi va fi reprezentată de un număr în baza 10, cu valoare în intervalul $\mathbf{[0, 2^m - 1]}$, unde $\boldsymbol{m}$ reprezintă precizia căutării (numărul de biţi ai unei alele). Astfel, lungimea unui cromozom va fi de **6m** biţi.

	Sir binar	Sir decimal
Kr	100001011	267
Ti	101001101	333
Td	111110100	500

Figure 5.4: Codificare parametri PID

Decodificarea parametrilor se face pe baza formulei:

$$x = x_{min} + val_{cromozom} \times \frac{x_{max} - x_{min}}{2^m - 1}$$

Astfel, în cazul valorilor din tabelul (5.4), pentru **m=9**:

$$K_R = 0 + 267 \times \frac{10 - 0}{511} = 5.22, K_R \in [0; 10]$$

$$T_i = 0 + 333 \times \frac{20 - 0}{511} = 13.03, T_i \in [0; 20]$$

$$T_d = 0 + 500 \times \frac{5 - 0}{511} = 4.89, T_d \in [0; 5]$$

În cazul nostru, se va alege pentru **m**, valoarea **6**, rezultând un cromozom alcătuit din **36 de biţi**. Prin urmare, spaţiul de căutare devine cu mult mai mare decât în cazul algoritmului genetic, coeficienţii trebuind modificaţi corespunzător.
Astfel, mărimea populaţiei devine $\boldsymbol{N = 120}$, numărul de operaţii genetice $\boldsymbol{N_{go} = 5N}$, iar numărul maxim de generaţii $\boldsymbol{N_g = 100}$.

O modificare importantă este reprezentată de alegerea unei **funcţii criteriu**. Evaluarea fitness-ului unui cromozom se realizează pe baza modelului identificat. După cum este ilustrat în schema de principiu din (5.5), sinteza regulatorului optimal se face cu ajutorul rezultatelor **stimulării cu treaptă a sistemului de reglare în buclă închisă**.

În calculul funcţiei criteriu se va ţine cont de răspunsul sistemului de reglare la treaptă. Metodele convenţionale folosite se bazează pe comparaţia acestui rşpuns şi a referinţei (răspuns ideal). Printre aceste metode se numără:

- $ISE = \int_0^\infty e^2(t)dt$
- $IAE = \int_0^\infty |e(t)|dt$
- $ITSE = \int_0^\infty te^2(t)dt$

- $ITAE = \int_0^\infty t|e2(t)|dt$

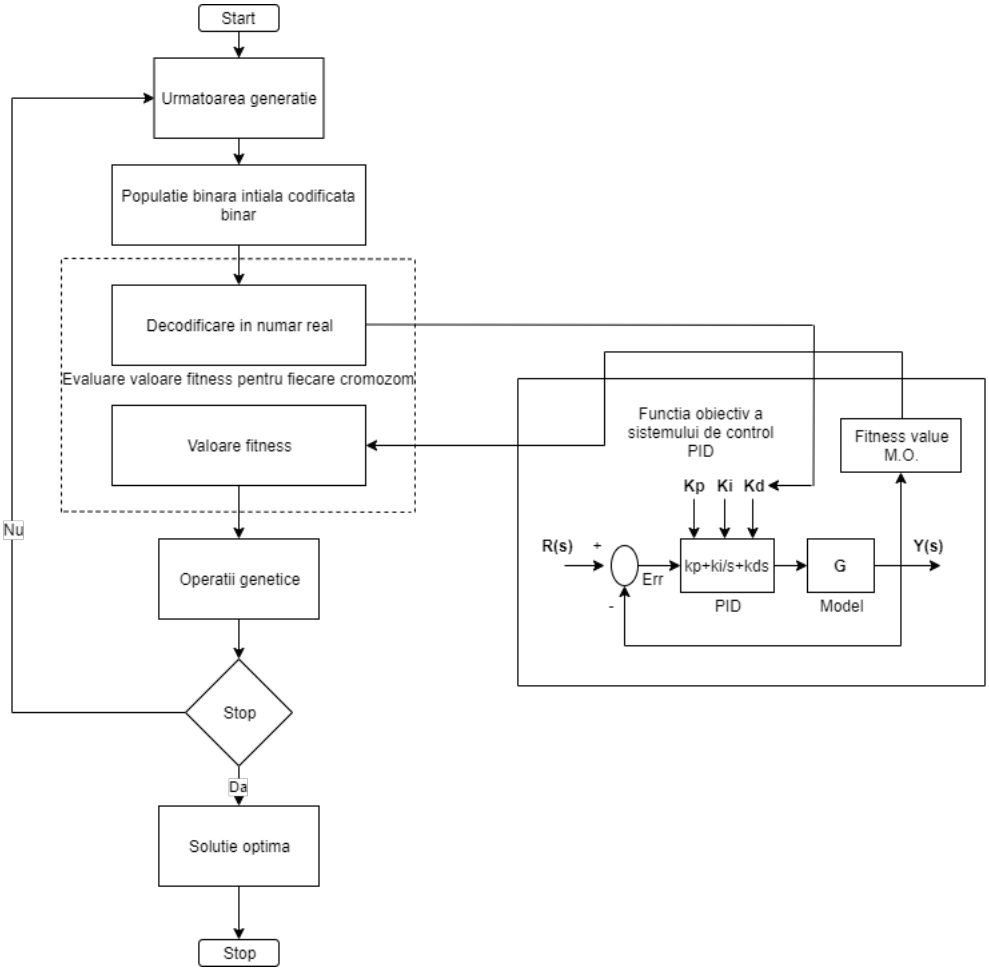

Figure 5.5: Schema de identificare a PID [4]

Din moment ce ne dorim să minimizăm eroarea, valoarea funcţiei criteriu va fi luată ca inversa valorii erorii (**indexul de performanţă**), deoarece cromozomul va fi cu atât mai bun cu cât valorile indicatorilor de performanţă vor fi mai mici, astfel se defineşte valoarea fitness-ului unui cromozom ca: $fitness = \frac{1}{Indicatori}$.

Pentru aplicaţiile în care problemele de decizie impun optimizarea mai multor obiective, se poate apela la metode de optimizare multi-obiectiv, precum ponderarea obiectivelor: $\boldsymbol{F(x) = \sum_{i=1}^{k} w_i f_i(x)}$, unde ponderile $w_i \in [0,1] si \sum_{i=1}^{k} w_i = 1$.

Conform [3], pentru optimizarea multi-obiectiv în cazul regulatoarelor PID se poate defini următoarea funcţie obiectiv care să ia simultan în calcul valorile mai multor indicatori de performanţă:

$$\boldsymbol{M.O. = w_1 * m_p + w_2 * t_s + w_3 * t_R} \tag{5.7}$$

Unde, m_p - suprareglaj maxim, t_S - timp tranzitoriu, t_R - timp de creştere.
Alte funcţii obiectiv se pot obţine prin combinarea metodelor convenţionale şi optimizarea multi-obiectiv: $F = w_1 * m_p + w_2 * MSE$

Fiind vorba de un sistem multivariabil, mai jos este prezentată schema de reglare care include cele două regulatoare PID folosite pentru controlul sistemului cu 2 intrări şi două ieşiri (modelat prin intermediul a patru modele ARX).

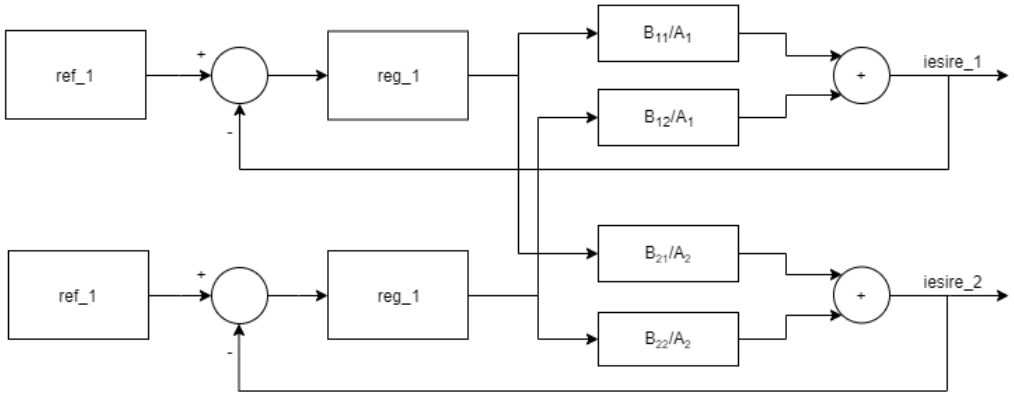

Figure 5.6: Schema de control multivariabil

6 Aplicaţie

6.1 Identificarea modelului

Înainte de a implementa algoritmii de căutare a indicilor structurali optimi în scopul estimării cât mai precise a procesului, au fost pregătite seturile de date pe baza cărora se va realiza acest proces. Au fost achizitionate seturi de date intrare-iesire, corespunzatoare mentinerii unei tensiuni constante asupra pompei principale si varind tensiunile pe cele doua electro-valve. S-au achizitionat 700 de esantioane pentru identificare si tot atatea pentru validare, cu o perioada de esantionare de Ts=1s. Setul de date de identificare cuprinde:

- doua semnale de intrare u_1, u_2, reprezentând variaţia tensiunilor pe cele două electro-valve
- doua semnale de ieşire, y_1, y_2, reprezentând nivelurile de apă din cele două rezervoare

Similar se construieşte şi setul de date de validare, cu menţiunea că cele două semnale de intrare pot fi inversate între ele, pentru obţinerea unui set de validare diferit de cel de identificare. Valorile tensiunii aplicate celor două electro-valve sunt cuprinse între 5V şi 10V. Tensiunea aplicată asupra pompei principale este constantă şi are valoarea de 9V.

6.1.1 Semnalul de stimul şi datele de identificare

Etapa de stimulare a procesului cu semnalul de intrare în vederea construirii datelor de identificare şi validare în funcţie de care se va realiza identificarea şi estimarea modelului matematic are o importanţă majoră. Apropierea modelului matematic estimat va fi limitată de precizia datelor de identificare oferite.

Astfel, este nevoie ca procesul să fie stimulat cu semnale care să surprindă cât mai exact dinamica acestuia. Trebuie identificate clasele de perturbaţii la care este supus procesul şi clasele de semnale acceptate de către proces. De exemplu, modelul poate fi determinat prin stimularea procesului cu semnale folosite în exploatarea sa uzuală, ceea ce ar conduce la o generalitate redusă modelului rezultat.

Alegerea semnalului de stimul se face astfel încât ordinul de persistenţă al acestuia să fie suficient de mare. Ordinul de persistenţă al unui semnal este cel mult egal cu perioada acestuia. Ideal, semnalul de intrare trebuie să fie unul de tip zgomot alb(neimplementabil). O soluţie implementabilă care presupune crearea unui semnal de stimul cu periodicitate (implicit, ordin de persistenţă) suficient de mare este SPA(B) - semnal pseudo-aleator (binar). Acesta presupune varierea între anumite valori (numărul lor fiind dat de rezoluţia semnalului) în manieră pseudo-aleatoare, astfel încât să se maximizeze perioada semnalului.

Este binecunoscut că semnalul de tip treaptă oferă informaţii utile cu privire la comportamentul tranzitoriu al procesului, însă nu este folosit ca semnal de stimul pentru crearea datelor de identificare, deoarece se pierd informaţii despre proces, treapta având ordinul de persistenţă 1.

Prin urmare, am folosit pentru generarea semnalelor de comanda u_1 şi u_2 semnale de tip SPA, cu paliere de lungime variabila, pentru a surprinde cât mai multe combinaţii diferite de perechi de tensiuni pe cele două electro-valve. Semnalele de comandă generate se pot observa în (6.1) şi (6.2).

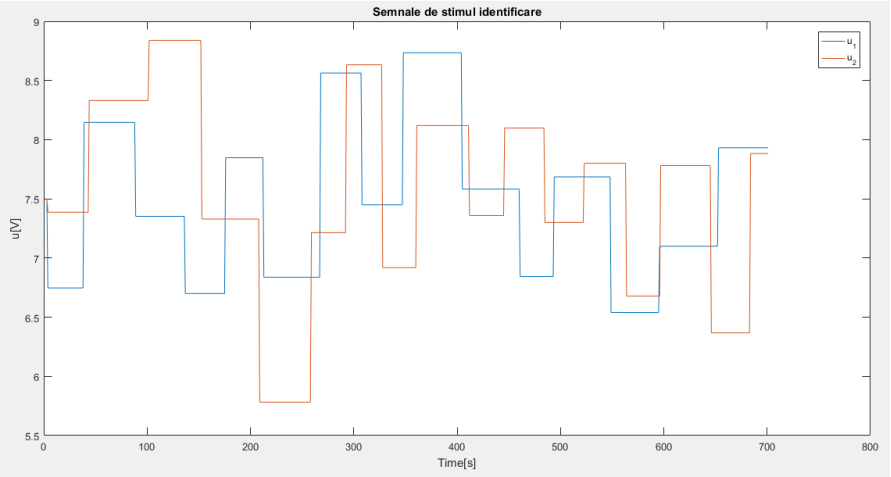

Figure 6.1: Semnalele de stimul pentru datele de identificare

Alegerea perioadei de eşantionare se face în funcţie de dinamica procesului care, în cazul ASTANK2 este lentă şi în funcţie de frecvenţa de tăiere. Conform **Teoremei de Eşantionare Kotel'nikov-Shannon-Nyquist**, frecvenţa de eşantionare minimă este egală cu dublul frecvenţei de tăiere.

$$F_s = \frac{1}{T_s} \geq F_{NYQ} = 2F_c = \frac{\Omega_c}{\pi} \tag{6.1}$$

Dacă frecvenţa de eşantionare este prea mică apare riscul pierderii de informaţie, dinamica neputând fi surprinsă întru totul.

Pe de altă parte, daca frecventţa de eşantionare este aleasă incorect, există riscul distorsionării informaţiei din cauza fenomenului de **aliasing**.
Pentru setul de date de identificare şi validare, a fost aleasă perioada de eşantionare $\boldsymbol{T_e = 1s}$.

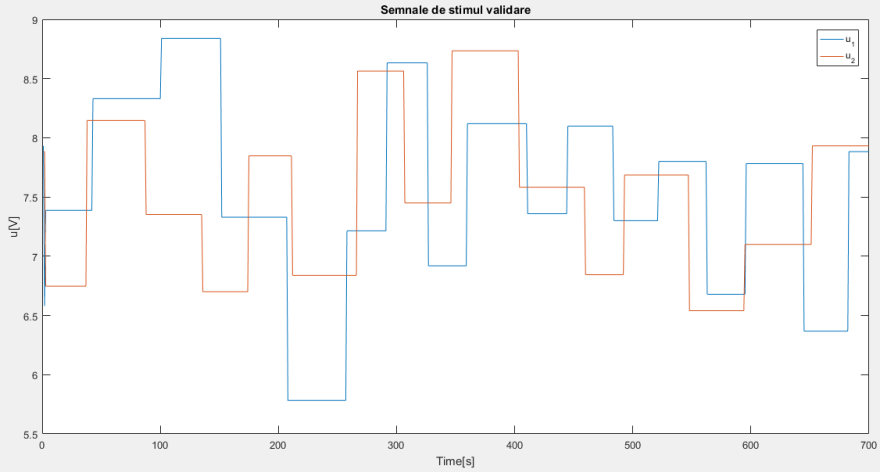

Figure 6.2: Semnalele de stimul pentru datele de valdiare

6.1.2 Funcţia fitness

Fiind stabilită structura modelului parametric ce trebuie identificat, cu partea utilă:

$$\begin{cases} A_1(q^{-1})y_1[n] = B_{11}(q^{-1})u_1[n] + B_{12}(q^{-1})u_2[n] + v_1[n] \\ A_2(q^{-1})y_2[n] = B_{21}(q^{-1})u_1[n] + B_{22}(q^{-1})u_2[n] + v_2[n] \end{cases} \quad \forall n \in \overline{1,N} \tag{6.2}$$

După cum a fost prezentat în 3.2, zgomotul colorat rezidual $v_r[n] = y[n] - y_{ARX}[n]$ va fi identificat cu ajutorul unui model ARMA:

$$\begin{cases} v_{r1}[n] = \frac{C_1(q^{-1})}{D_1(q^{-1})}e_1[n] \\ E\{e_1[n]e_1[m]\} = \lambda^2\delta_0[n-m] \\ v_{r2}[n] = \frac{C_2(q^{-1})}{D_2(q^{-1})}e_2[n] \\ E\{e_2[n]e_2[m]\} = \lambda^2\delta_0[n-m] \end{cases} \quad \forall n, m \in \overline{1,N} \tag{6.3}$$

Se modelează sistemul **MIMO 2x2** ca 2 sisteme de tip **MISO 2x1** separate, sub forma parametrică prezentată mai sus. Trebuie realizată identificarea separată a celor 2 modele, pentru fiecare ieşire în parte.

Identificarea unuia dintre modele se realizează, de asemenea, în 2 etape:

- Identificarea funcţiei de transfer a **părţii utile**: polinoamele **B** şi **A**
- Identificarea **filtrului de zgomot**: polinoamele **C** şi **D**

Obiectivul etapei de identificare este găsirea acelor polinoame menţionate în (6.2) care să filtreze semnalul de stimul şi perturbaţia astfel încât să fie obţinută o ieşire globala care să fie cât mai asemănătoare cu ieşirea reală a procesului stimulat cu

acelaşi semnal ca şi modelul de identificare.

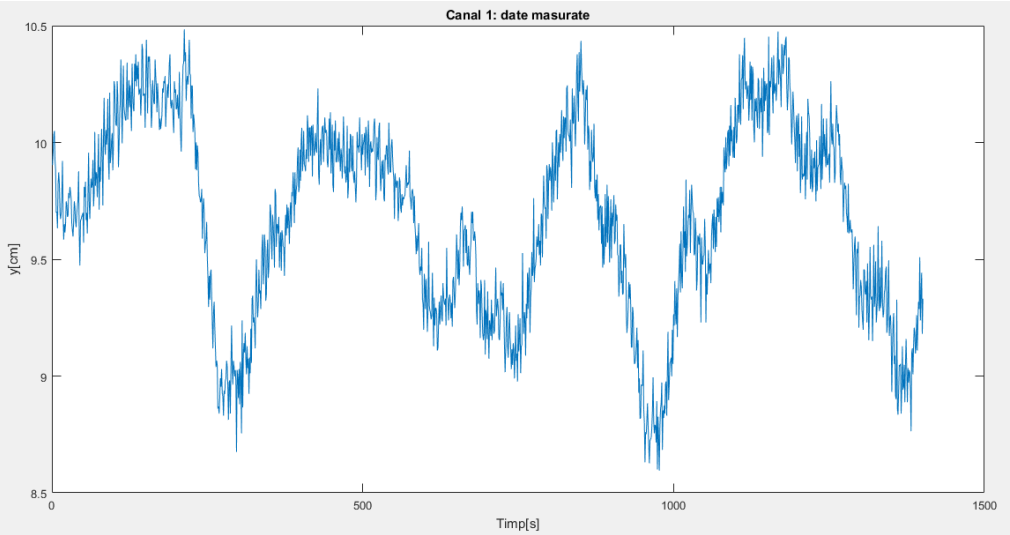

Figure 6.3: Nivelul măsurat din bazinul 1

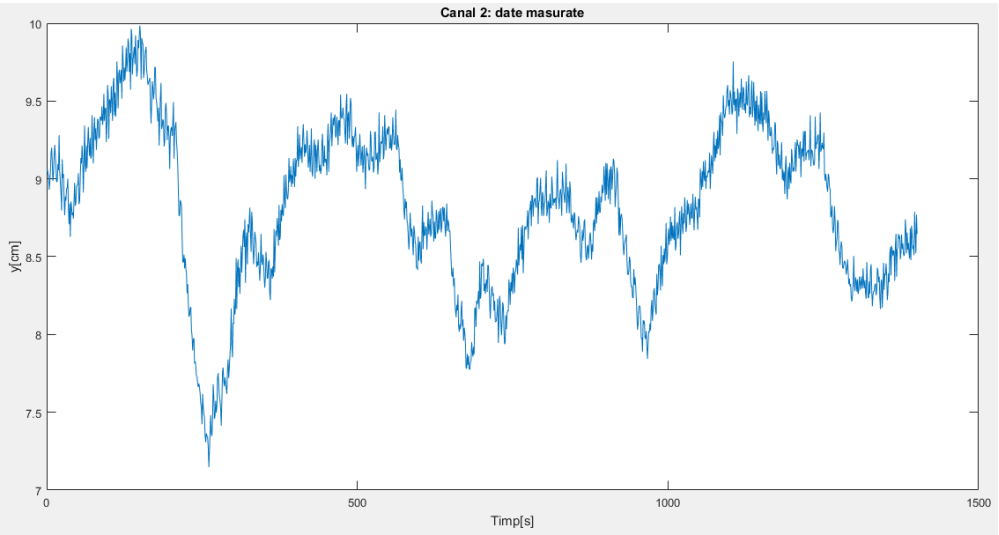

Figure 6.4: Nivelul măsurat din bazinul 2

Ne dorim cu precădere să minimizăm eroarea dintre nivelului măsurat şi partea utilă ieşirii simulate. Astfel, împărţirea pe etape a procesului de identificare este cea corectă, realizând identificarea filtrului de zgomot doar după ce funcţia de sistem a părţii utile a fost identificată iar valoarea SNR calculat cu ieşirea măsurată staționarizata şi eroarea de predicţie a fost maximizată.

În condiţiile în care modelul de identificare este compus din partea de zgomot şi partea utilă se pune problema identificării unui singur model de tip ARMAX cu min-

imizare globală erorii de predicţie. Această abordare s-a dovedit a fi greşită deoarece, deşi valoarea SNR globală era maximizată, extrăgând din răspuns partea utilă s-a putut observa că valoarea SNR a acestuia nu a fost maximizată (ilustrat în 6.5), deci eroarea de predicţie dintre ieşirea măsurată şi partea utilă a ieşirii modelului nu a fost minimizată. Acest lucru este ilustrat în următoarul grafic:

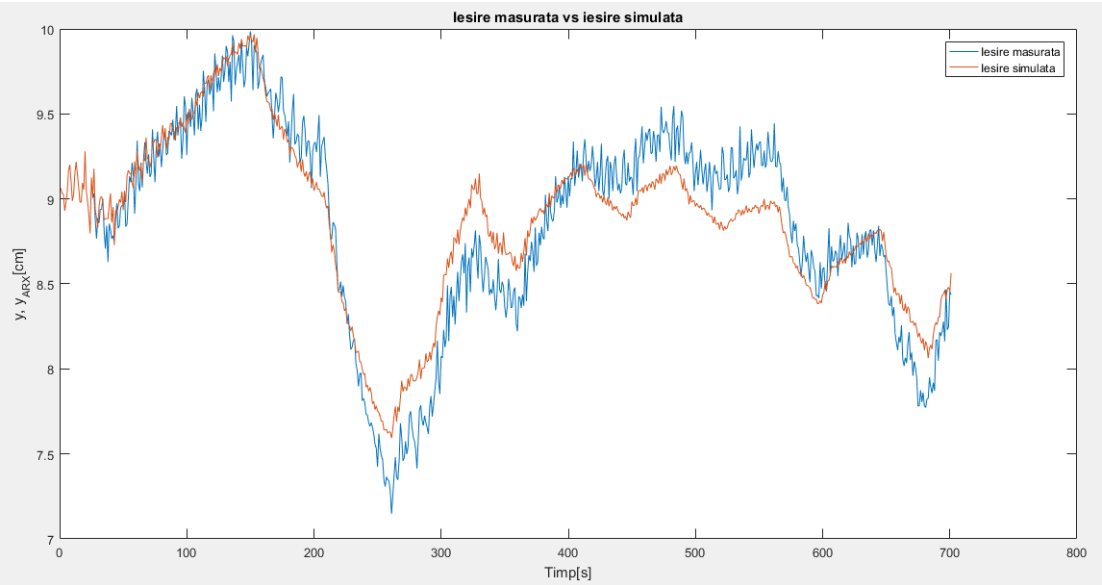

Figure 6.5: Date măsurate vs date simulate cu model ARX

Pentu răspunsul de mai sus, valoarea SNR pentru partea utilă este **6.4673 dB**. Se poate observa că eroarea de predicţie este semnificativă. Cu toate acestea, SNR pentru întreg semnalul (incluzând partea de zgomot) este **15.0003 dB**, o valoare mult mai mare. **Se poate concluziona că utilizarea unui model de tip ARMAX cu minimizare globală a erorii de predicţie nu va conduce la un SNR mai bun pentru partea utilă a ieşirii**.

Prin urmare, identificarea modelului parametric din (6.2) se va realiza, după cum s-a precizat, în 2 etape. În ambele etape funcţia criteriu are forma (definită în paragraful 3.3) :

$$\boldsymbol{f_{criteriu} = \alpha f_{ident} + (1-\alpha) f_{valid}}$$

$$f_{ident} = dB\left(\frac{||y_{identificare}||}{||err_{identificare}||}\right)$$

$$f_{valid} = dB\left(\frac{||y_{validare}||}{||err_{validare}||}\right)$$

Pentru termenul $\boldsymbol{\alpha}$, s-a ales valoarea **0.5** pentru a oferi egală importanţă setului de date de identificare şi celui de validare. Valorile nivelurilor din (6.3) şi (6.4) au fost separate în două seturi de dimensiuni egale. Pentru a realiza o identificare riguroasă, se doreşte ca ieşirea simulată să se apropie cât mai mult de valoarea întreg semnalului

real, atât a părţii de identificare cât şi a celei de validare.

- **Identificarea părţii utile**
  În calcularea valorii SNR-ului se ţine, eroarea de predicţie pentru date de identificare se calculează ca diferenţa dintre ieşirea măsurată şi ieşirea simulată cu modelul de identificare ARX, conform relaţiilor din secţiunea 3.2:

  $$A(q^{-1})y[n] = B_1(q^{-1})u_1[n] + B_2(q^{-1})u_2[n] + v_r[n]$$

  Astfel, $\boldsymbol{err_{identificare} = v_r = y - y_{ARX}}$, unde $\boldsymbol{y_{ARX}}$ este obţinut conform ecuaţiei 3.6.

- **Identificarea părţii de zgomot**

  $$D(q^{-1})v_r[n] = C(q^{-1})e[n]$$

  Această etapă presupune căutarea polinoamelor **C** şi **D** care să aproximeze cât mai bine eroarea de predicţie de la etapa anterioară.
  În această etapă: $\boldsymbol{err_{identificare} = \hat{e} = y - y_{ARX} - v_{ARMA}}$, unde $\boldsymbol{v_{ARMA}}$ este obţinut conform ecuaţiei 3.9.

6.1.3 Implementare algoritm genetic

S-a ales ca algoritmul genetic să fie implementat cu totul în cadrul acestei lucrări şi să nu se folosească funcţia **AG** din Matlab. Această decizie are fundamentul comparaţiei dintre rezultatele obţinute de această metaeuristică şi metaeuristica prezentată în (4.2). De asemenea, pentru a studia mai bine diferitele metode de aplicare a operaţiilor genetice sau de selecţie a indivizilor pentru reproducere şi pentru a oferi flexibilitate totală asupra comportamentului algoritmului s-a evitat folosirea rutinei deja implementate în suita oferită de Matlab.

După cum a fost precizat în (4.3), optimizarea folosind strategii evoluţionare, precum este cazul algoritmilor genetici [7], presupune imitarea în mod naiv a proceselor de evoluţie şi dezvoltare întâlnite în natură.

Se pleacă de la o populaţie iniţială de indivizi ce reprezintă posibile soluţii ale problemei granulare de optimizare. Fiecare individ reprezintă un cromozom alcătuit din mai multe alele ce codifică anumită informaţie. În cazul problemei de căutare a setului optim de indici structurali, un cromozom va fi alcătuit dintr-un număr de alele ce reprezintă valoarea fiecăruia din indicii structurali ai modelului parametric.

În cazul identificării modelului optim pentru partea utilă a sistemului, cromozomul va fi alcătuit din 3 alele, ce vor codifica valorile indicilor structurali **na**, **nb1** şi **nb2**. Deşi ne interesează doar partea utilă, valoarea parametrilor optimi ai polinoamelor $\boldsymbol{A, B_1, B_2}$ va fi obţinută cu ajutorul funcţiei **armax** din cutia de instrumente de identificare a sistemelor din Matlab. Această funcţie foloseşte **metoda minimizării**

erorii de predicţie pentru a optimiza parametrii celor 4 polinoame. Pe noi ne interesează doar polinoamele implicate în transferul direct de la intrare la ieşire. S-a folosit valoarea 0 pentru coeficientului corespunzător indicelui structural **nc** - ordinul polinomului **C**. Aşadar, s-a preferat căutarea modelului optim al părţii utile într-un spaţiu **3-dimensional**.

Iniţializarea algoritmului se face în 3 etape:

- Stabilirea valorilor **parametrilor de configurare**
- Formarea unei **populaţii iniţiale**
- Evaluare valoare **fitness** a fiecărui individ al populaţiei iniţiale

Printre **parametrii de configurare** cei mai importanţi se numără:

- mărimea populaţiei **N** - proporţională cu mărimea spaţiului de căutare
- numărul de poziţii vacante pentru grupul de reproducători - $\mathbf{Nr = 0.8N}$
- numărul de operaţii genetice ce trebuie realizate la fiecare iteraţie $\boldsymbol{N_{og} \in \overline{N, 6N}}$
- probabilitatea efectuării fiecăreia din operaţiile genetice - $\boldsymbol{P_c, P_m, P_i}$ - modificate pe parcursul rulării în următoarele intervale: $P_c \in [0.5, 0.9]$, $P_i = P_m \in [0.05, 0.25]$.
- metodele de selecţie pentru reproducere şi supravieţuire
- numărul maxim de generaţii - $\boldsymbol{N_g}$ - ales în funcţie de mărimea spaţiului de căutare
- factorul de supravieţuire - $\boldsymbol{S \in [2, 10]}$ - acesta scade odată cu trecerea iteraţiilor

Valorile parametrilor menţionaţi au fost alese conform literaturii de specialitate [7] şi calibrate ulterior în urma câtorva încercări, pentru a asigura un compromis între viteza de convergenţă şi precizia algoritmului.

Mărimea spaţiului de căutare este dată de valorile maxime ale indicilor structurali: $\boldsymbol{na_{max} X nb1_{max} X nb2_{max}}$. S-a ales mărimea populaţiei algoritmului $\boldsymbol{N \approx max\{na_{max}, nb1_{max}, nb2_{max}\}}$.

Iniţializarea populaţiei presupune alegerea în mod aleator a poziţiilor pe care să le ocupe indivizii în spaţiul de căutare. Aceasta se realizează conform unei distribuţii uniforme de probabilitate, fiind nevoie de un **U-PRSG**. Pentru a realiza acest lucru s-a folosit **algoritmul lui Baker** [7] de generare de numere aleatoare. Astfel, pentru fiecare indice structural, s-a generat o secvenţă aleatoare de numere din intervalul **[1, max]**, toate având aceeaşi probabilitate. Secvenţele astfel obţinute vor fi amestecate, pentru a genera fiecare cromozom în parte (conform 6.6).

	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10
na	2	1	5	9	4	7	3	6	10	8
nb1	3	1	10	8	4	2	7	6	5	9
nb2	2	4	5	3	7	1	9	8	10	6
nc	4	9	7	2	5	10	3	1	6	8

Figure 6.6: Iniţializare uniformă a populaţiei

Un exemplu de populaţie iniţială generată cu ajutorul algoritmului lui Baker este următorul:

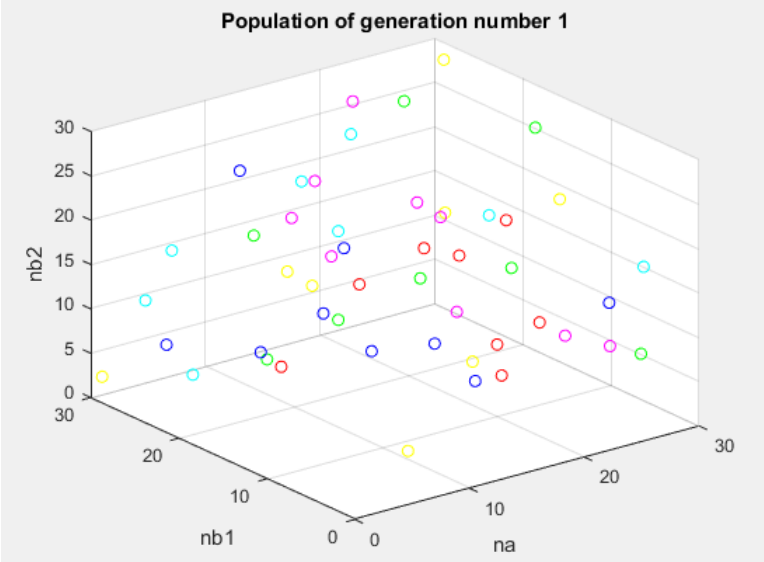

Figure 6.7: Indivizii primei populaţii

Calcularea valorii fitness-ului pentru fiecare individ în parte se face conform procedurii descrise în (6.1.2).

În acest moment poate începe procesul iterativ al algoritmului. O iteraţie reprezintă trecerea de la o generaţie la alta. O iteraţie este alcătuită din următorii paşi:

- calcularea valorii maxime a fitness-ului întâlnit în cadrul populaţiei
- selectarea indivizilor participanţi la operaţiile de reproducere
- aplicarea operaţiilor genetice asupra indivizilor selectaţi la etapa **2**, în diferite combinaţii
  - verificarea viabilităţii indivizilor noi formaţi
  - verificarea că indivizii noi pot fi adăugaţi în populaţie fără a introduce duplicate
  - calcularea fitness-ului noilor indivizi
- selecţia supravieţuitorilor

- poziţionarea moştenitorilor înapoi în populaţie
- verificarea condiţiilor de **stop**

Procesul de selecţie a indivizilor asupra cărora sunt aplicate operaţiile genetice este realizat prin metoda **selecţiei prin competiţie** [5]. Pentru fiecare poziţie vacantă din grupul de reproducători procesul de selecţie constă în alegerea din populaţia de indivizi a 2 competitori, pe baza unei distribuţii uniforme de probabilitate. Fiecare poziţie are asignată o valoare proprie $\eta \in [0, 1]$. Pentru competiţia curentă se generează o valoare aleatoare $v \in [0.5, 1]$. Pe baza comparaţiei dintre aceste valori, pierzătorul sau câştigătorul competiţiei este ales şi trecut în grupul de reproducători. Individul participant care nu a fost ales este marcat în scopul interdicţiei de a mai concura până când nu au participat toţi indivizii populaţiei:

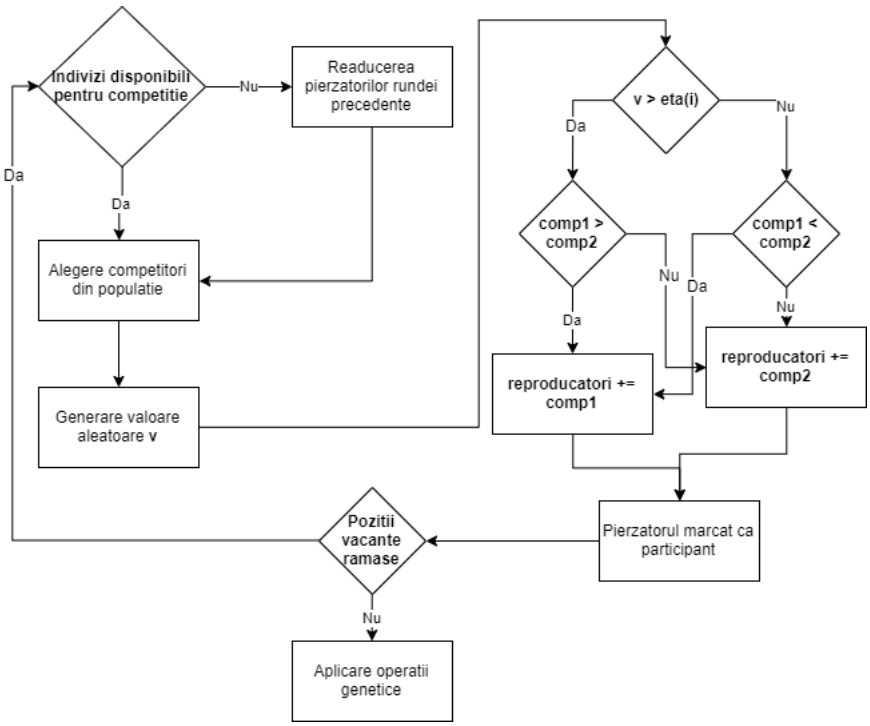

Figure 6.8: Selecţia prin competiţie

Mai departe, operaţiile genetice prezentate în (4.3) sunt aplicate asupra indivizilor din grupul de reproducători format:

Se stabilesc probabilităţile aplicării fiecărei operaţii în parte:

$P_c = \left[\frac{\left[\frac{iter}{5}\right]}{10}\right]$; $P_m = \frac{1-P_c}{2}$; $P_i = P_m$

Probabilitatea aplicării operaţiei de încrucişare creşte odată cu trecerea iteraţiilor pentru a promova exploatarea din ce în ce mai mult.

După stabilirea probabilităţilor se va aplica **algoritmul lui Baker** pentru a genera o secvenţă de operaţii genetice care să respecte distribuţia de probabilitate pre-

scrisă anterior.

Pentru aplicarea operaţiilor genetice se vor codifica binar alelele cromozomilor implicaţi. Am stabilit că alelele sunt reprezentate de valorile indicilor structurali în baza 10. Rezultatele operaţiilor genetice vor fi secvenţe binare pe care trebuie să le decodificăm pentru a obţine valorile corespunzătoare în baza 10. Moştenitorii astfel obţinuţi vor fi făcuţi viabili prin repoziţionarea în spaţiul de căutare (dacă au ieşit din limitele acestuia). Dacă indivizii cei noi nu sunt soluţii deja existente în populaţia curentă va fi evaluată valoarea fitness-ului lor.

Următorul pas este întoarcerea celor mai capabili indivizi pentru generaţia următoare. Acest lucru se realizează prin alegerea celor mai buni 2 indivizi în urma fiecărei operaţii de încrucişare. Deoarece prin mutaţie şi inversiune se obţin, de cele mai multe ori, indivizi mai puţin capabili, în urma aplicării acestor operaţii, individul mutant rezultat va fi returnat automat în populaţie, pentru a încuraja exploatarea.

Înainte de a trece la iteraţia următoare se va verifica dacă cel mai puternic individ este acelaşi sau nu. Testul de stop constă în verificarea dacă numărul maxim de iteraţii a fost atins sau dacă individul cel mai puternic a supravieţuit suficient de multe iteraţii fără să se fi găsit o soluţie mai bună.

Rezulatele obţinute în urma căutării setului de indici structurali într-un spaţiu de căutare de dimensiune $[na, nb_1, nb_2] = [30, 30, 30]$ şi $[nc, nd] = [100, 100]$ sunt prezentate pentru fiecare ieşire în parte. Sunt prezentate evoluţia fitness-ului părţii utile (6.9-prima ieşire şi 6.13-a doua ieşire), graficul părţii utile (6.10 şi 6.14), graficul semnalului de ieşire total (parte utilă + parte de zgomot) (în 6.11 şi 6.16) şi zgomotul alb rezidual (6.12 şi 6.15).

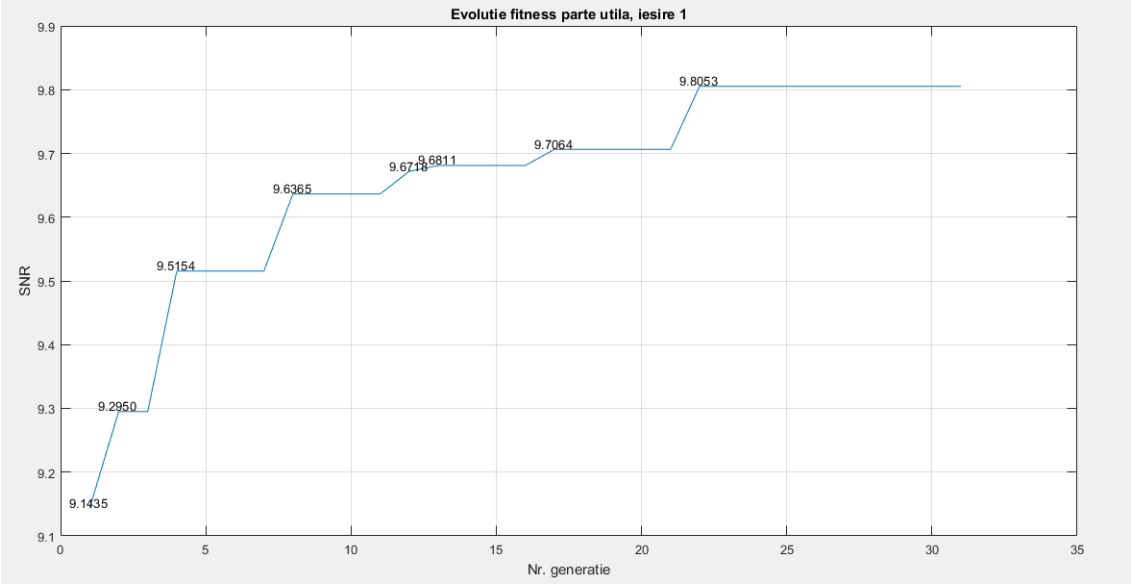

Figure 6.9: Evolutie fitness parte utilă iesire 1

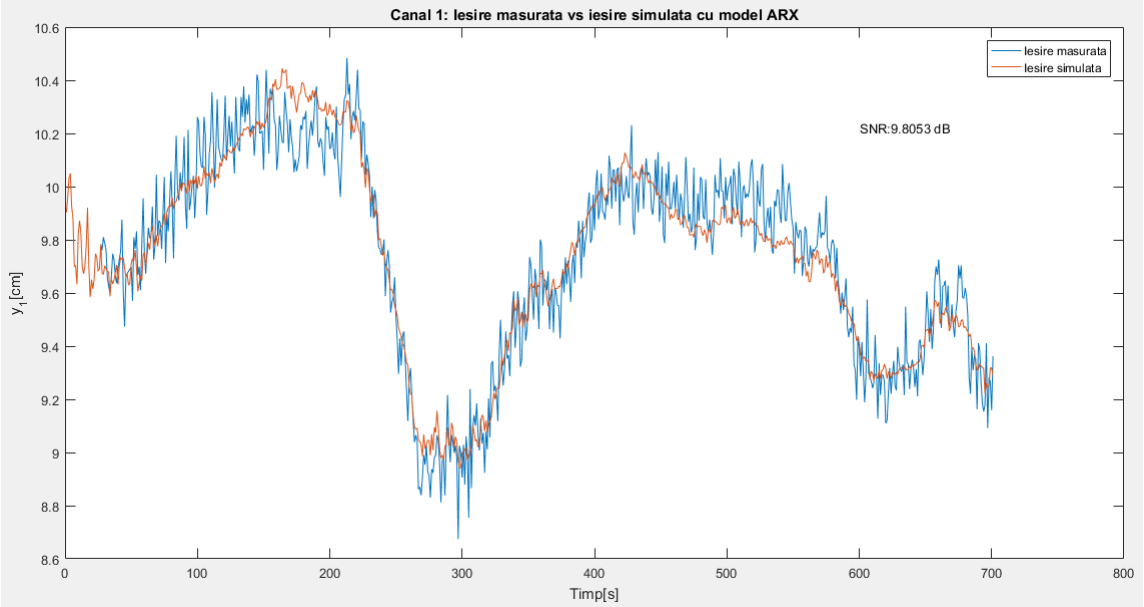

Figure 6.10: Canal 1: Ieşire măsurată vs ieşire simulată cu model ARX

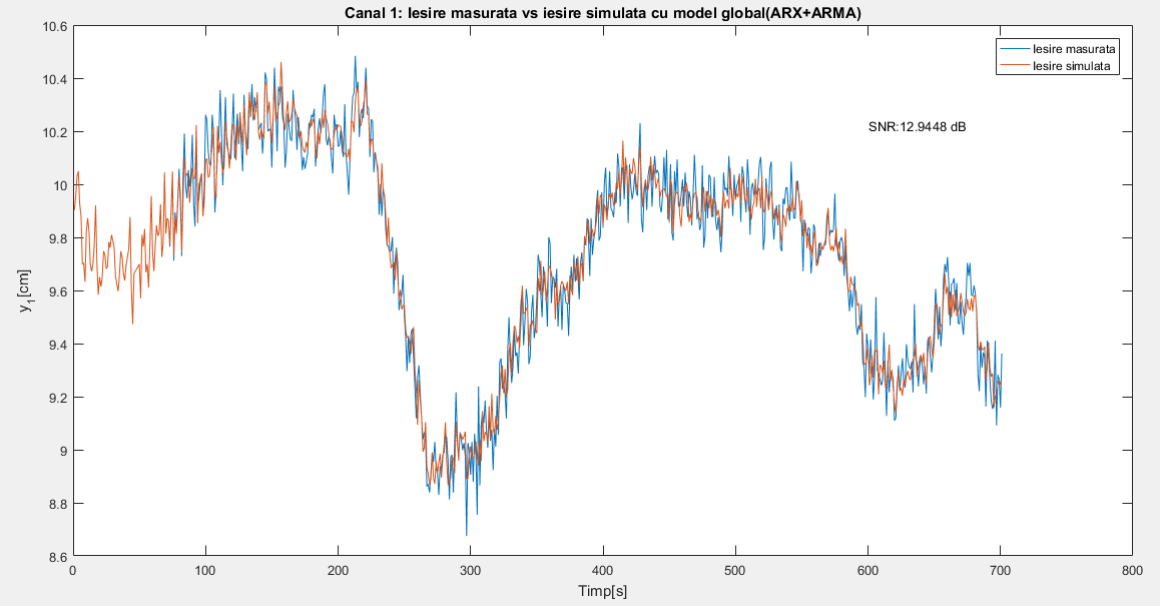

Figure 6.11: Canal 1: Ieşire măsurată vs ieşire simulată cu model global

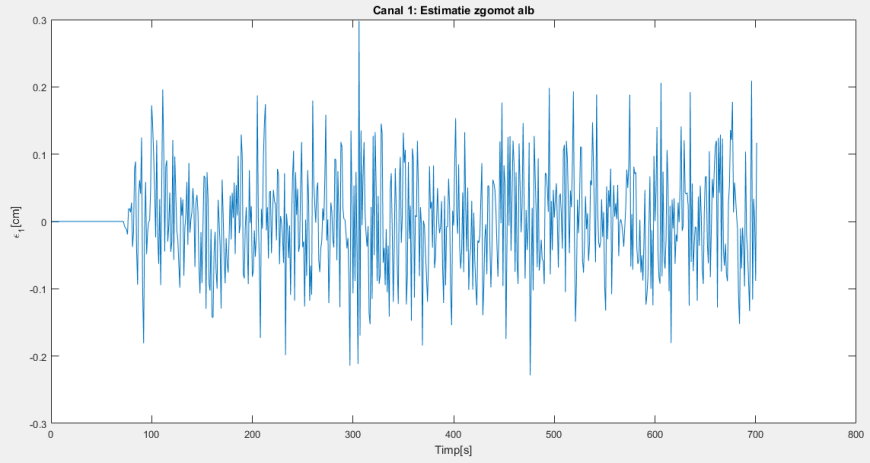

Figure 6.12: Zgomot alb rezidual ieşire 1

Algoritmul genetic a fost folosit pentru identificarea ambelor modele MISO. Indicii structurali ai filtrului de zgomot au fost determinaţi într-o manieră asemănătoare cu determinarea filtrului părţii utile. Au fost obţinute rezultatele prezentate în graficele (6.9), (6.10) şi 6.11).
SNR pentru partea utilă are valoarea 9.8053 dB, pentru setul de indici structurali [na=21 nb1=25 nb2=27]
SNR pentru întreg semnalul de ieşire simulat este 12.9448 dB, pentru setul de indici structurali [nd=96 nc=24]

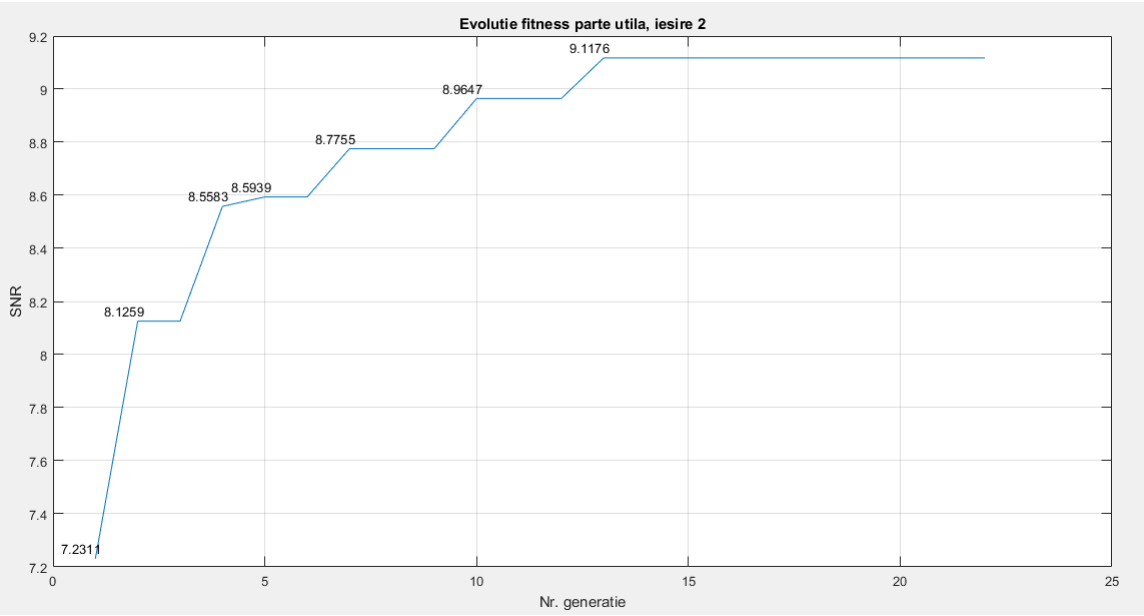

Figure 6.13: Evolutie fitness parte utilă ieşire 2

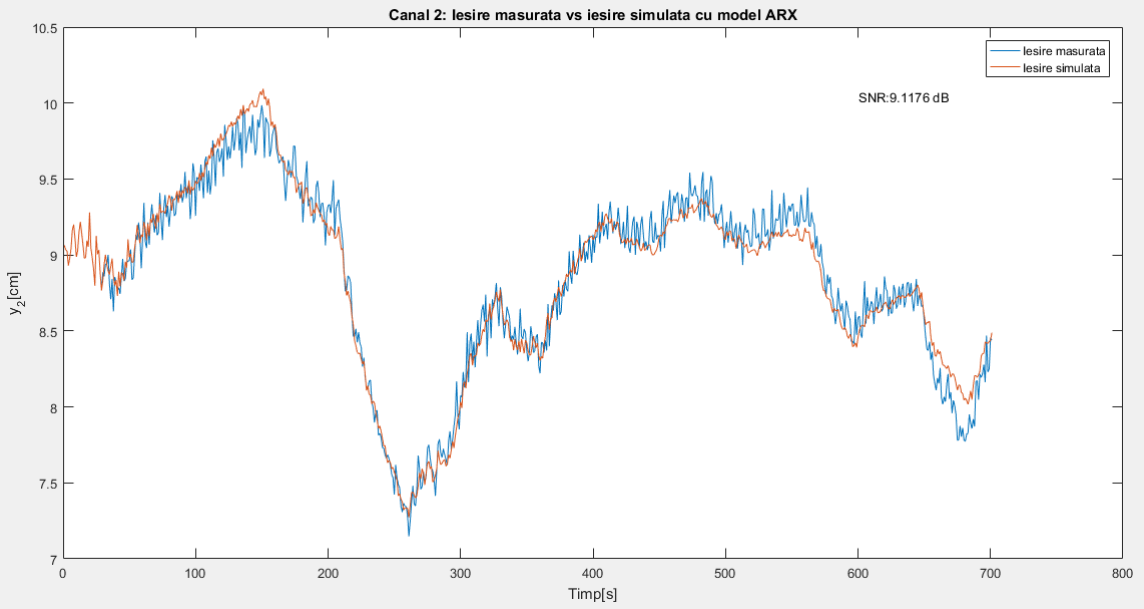

Figure 6.14: Canal 2: Ieşire măsurată vs ieşire simulată cu model ARX

Pentru ieşirea a doua, procesul MISO de umplere al celui de-al doilea bazin, bazin ce prezintă forma trapezoidală, rezultatele identificării nu au fost la fel de bune, cel puţin pentru partea utilă a semnalului.
SNR pentru partea utilă are valoarea 9.1176 dB, pentru setul de indici structurali [na=25 nb1=19 nb2=29]
Pe de altă parte, acest bazin este mai puţin afectat de zgomote, astfel că semnalul total simulat are un SNR mai bun decât în cazul primului bazin.
SNR pentru partea utilă are valoarea 15.1436 dB, pentru setul de indici structurali [nd=89 nc=9]

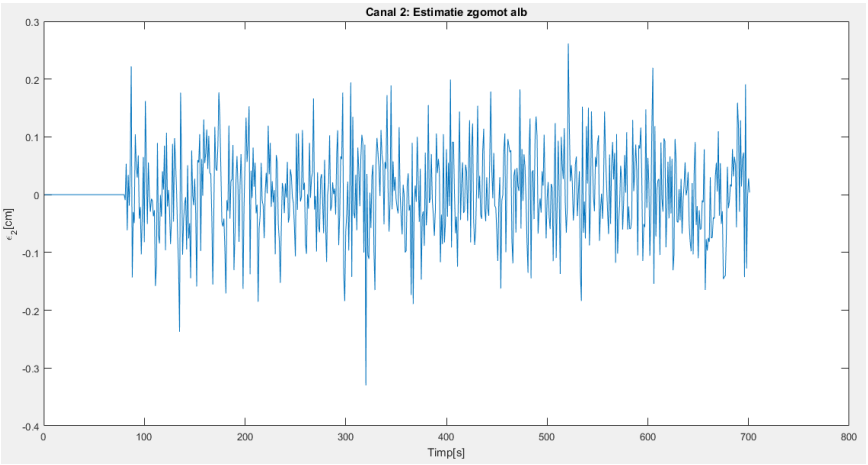

Figure 6.15: Zgomot alb rezidual ieşire 2

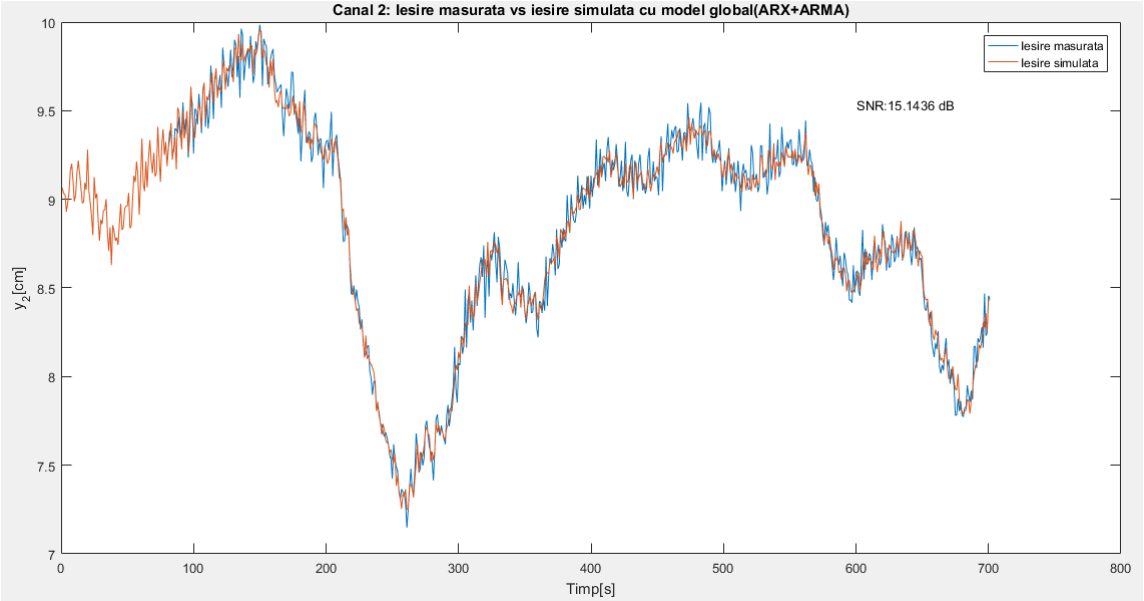

Figure 6.16: Canal 2: Ieşire măsurată vs ieşire simulată cu model global

6.1.4 Implementare algoritm licurici

Algoritmul bazat pe licurici [7] a fost prezentat în (4.2). Acest algoritm este bazat pe mecanismele de atracţie ce se desfăşoară într-o populaţie de licurici. În contextul acestui algoritm, o posibilă soluţie a problemei de identificare va fi reprezentată printr-un licurici ce are un comportament şi nişte proprietăţi specifice (ex. emite lumină, este atras de alţi licurici din grup).

Iniţializarea populaţiei se va face asemănător cu iniţializarea populaţiei din prima generaţie a algoritmului genetic (6.1.3). Se foloseşte, deci, o distribuţie uniformă de probabilitate pentru a genera valoarea indicilor structurali. Un licurici este responsabil cu reţinerea informaţiei despre un model, anume valorile indicilor structurali de interes. De asemenea, fiecare licurici are capacitatea de a emite fasciculi de lumină în toate direcţiile, direct proporţional cu valoarea fitness-ului modelului identificat pe baza indicilor structurali reţinuţi.

Licuricii vor explora spaţiul de căutare, schimbându-şi poziţia după fiecare iteraţie indiferent de valoarea fitness-ului lor de la iteraţia curentă. Astfel, este posibil ca valoarea cea mai bună găsită de algoritm să nu se găsească în cadrul populaţiei finale. Este posibil ca soluţia optimă găsită de algoritm să fi fost întâlnită chiar şi în cadrul uneia din primele iteraţii în drumul unuia din licurici. Valoarea cea mai bună care a fost întâlnită este reţinută separat.

Se calculează valoarea funcţiei fitness în punctele în care se află fiecare licurici al primei populaţii.

După aceasta poate începe procesul iterativ de căutare.

Fiecare iteraţie a algoritmului se desfăşoară conform următorilor paşi:

- Pentru fiecare licurici în parte se va calcula poziţia sa viitoare:
  - **Cacularea intensităţii luminoase** a fasciculilor emişi de fiecare licurici.
  - Se va considera o **subpopulaţie de posibili licurici ţintă**, formată din toţi licuricii mai strălucitori
  - Se formează o **distribuţie de probabilitate** direct proporţională cu intensitatea luminoasă a licuricilor
  - Se foloseşte algoritmul lui Baker pentru a genera **licuriciul ţintă**
  - Determinarea **poziţiei viitoare** pe baza licuriciului ţintă şi a perturbaţiilor din partea celorlalţi licurici:
    - Se calculează **perturbaţia** datorată fasciculilor de lumină pe care licuriciul curent îi percepe
    - Se calculează **influenţa licuriciului ţintă**
    - Rezultatele de la cei 2 paşi anteriori se combină pentru a obţine **noua poziţie** a licuriciului
  - Se aduce în spaţiul de căutare licuriciul, dacă este cazul
  - Se **calculează fitness-ul** acestuia şi se reţine în cazul în care este mai mare decât cel mai bun întâlnit de până acum
- Licuriciul aflat în noua sa poziţie va ocupa **noul loc în cadrul populaţiei**, în funcţie de fitness-ul său.

Intensitatea luminoasă a licuricilor se calculează plecând de la valoarea fitness-ului acestora, în funcţie de care sunt sortaţi în cadrul populaţiei. Fiecare licurici va avea o intensitate cuprinsă între **0.2**, pentru cel mai mic fitness şi **1**, pentru cel mai mare fitness.

Poziţia viitoare a fiecărui licurici se va calcula pe baza influenţei licuriciului ţintă şi a fasciculilor perturbatori, după cum este ilustrat în (6.17).

Licuriciul ţintă, determinat pe baza intensităţii luminoase, va influenţa în mod decisiv traiectoria individului curent, care va încerca să se deplaseze pe direcţia acestuia şi se va apropia cu atât mai mult cu cât intensitatea ţintei este mai mare.

$$x_p^{k+1} = x_p^k + \beta_{p,q}^k (x_q^k - x_p^k) + disturbance$$

Unde,

$$\beta_{p,q} = I_0 e^{-\gamma * distance}$$

$\boldsymbol{I_0}$ - intensitatea luminoasă licuriciului ţintă, iar ***distance*** - distanţa dintre cei doi licurici în spaţiul de căutare.

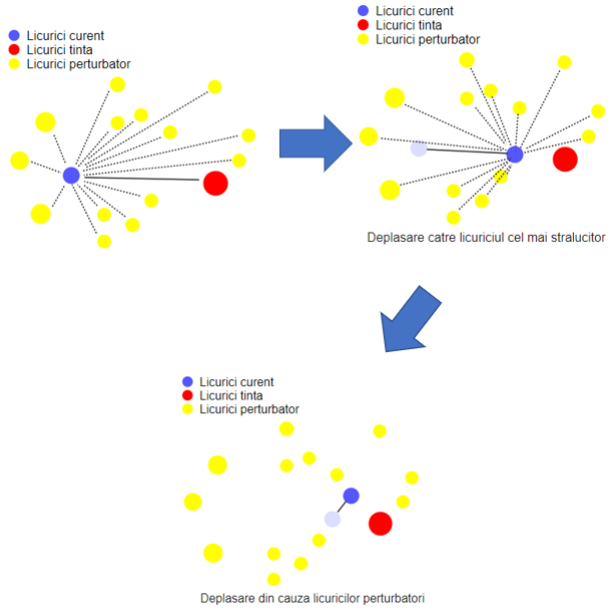

Figure 6.17: Determinare poziţie viitoare

Perturbaţia datorată luminii percepute din mediu se determină prin:

- determinarea direcţiei perturbatoare, $\boldsymbol{\zeta}$ - se realizează folosind algoritmul lui Baker şi **distribuţia lui Levy** pentru a determina valorile de deplasare de-a lungul fiecărei dimensiuni a spaţiului.
- determinarea în mode aleator a sensului perturbaţiei
- generarea unui factor aleator, $\boldsymbol{\alpha}$ care cuantifică influenţa celorlalţi licurici.

$$disturbance = \alpha^k sign(\rho^k - \frac{1}{2})\zeta^k(\phi_s)$$

Unde, ζ este un vector unidimensional cu lungimea dată de numărul dimensiunilor spaţiului de căutare. Fiecare componentă a lui ζ se află în intervalul $[1; max_{indice}]$.

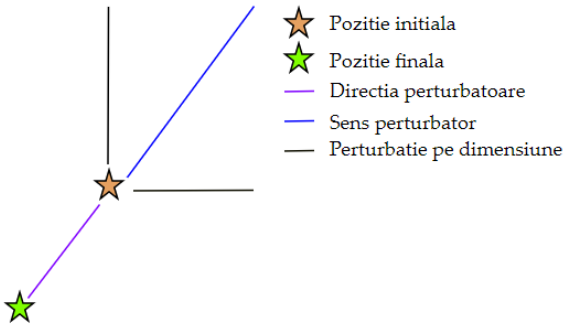

Figure 6.18: Perturbatia intensităţii luminoase a mediului

De exemplu, în (6.18), este ilustrat cum se determină perturbaţia luminoasă în cazul unui spaţiu de căutare 2-dimensional.

Asemănător cu 6.1.3 se prezintă rezultatele obţinute în urma rulării algoritmului bazat pe roi de licurici pe un spaţiu de căutare similar cu cel din cazul algoritmului genetic. Evoluţia fitness-ului părţii utile, graficele cu partea utilă a semnalului şi cu semnalul total, alături de zgomotul alb rezidual sunt prezentate în (6.19)-(6.26).

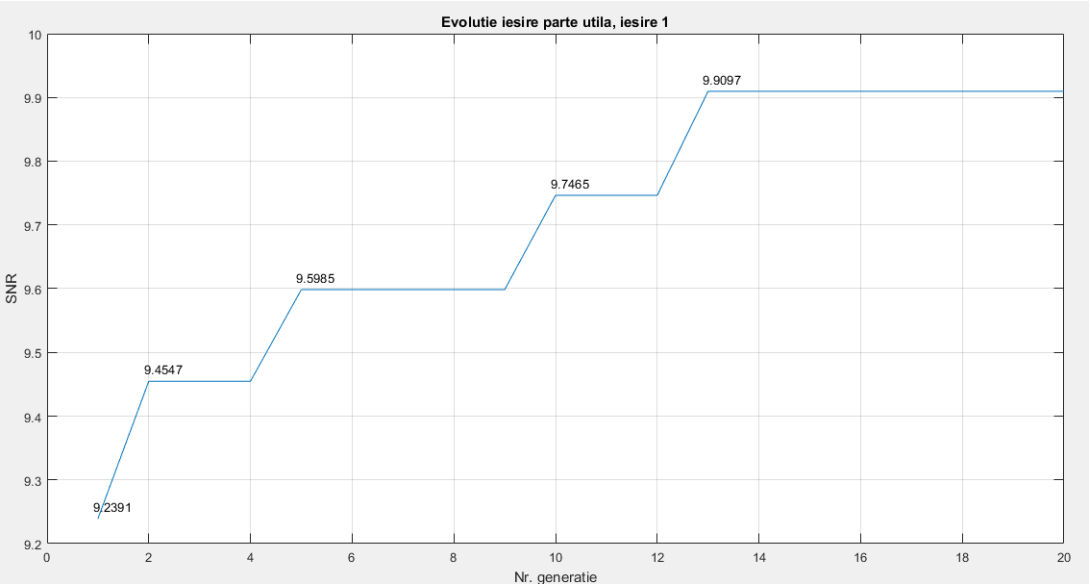

Figure 6.19: Evolutie fitness parte utila iesire 1

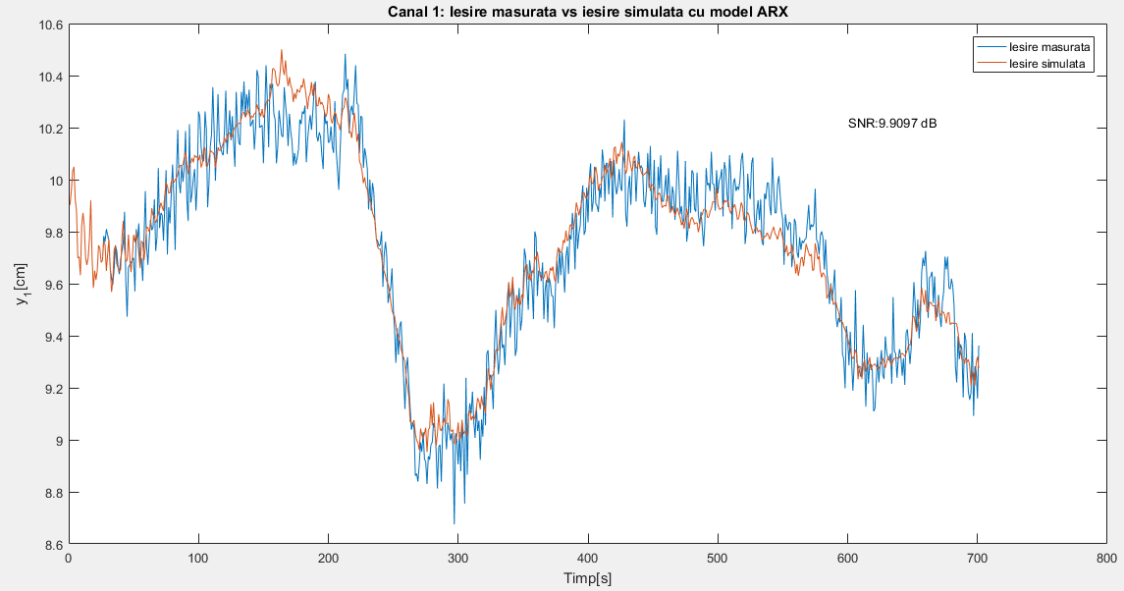

Figure 6.20: Canal 1: Ieşire măsurată vs ieşire simulată cu model ARX

Modelul identificat pentru prima ieşire are indicii structurali [**na=18 nb1=20 nb2=26**] pentru partea utilă, cu valoarea fitness-ului **9.9097 dB**. Evoluţia pe parcursul generaţiilor este prezentată în (6.19).
Modelul părţii de zgomot are indicii structurali [**nd=90 nc=16**], iar fitness-ul ieşirii simulate totale pentru primul bazin este **12.6560 dB**.

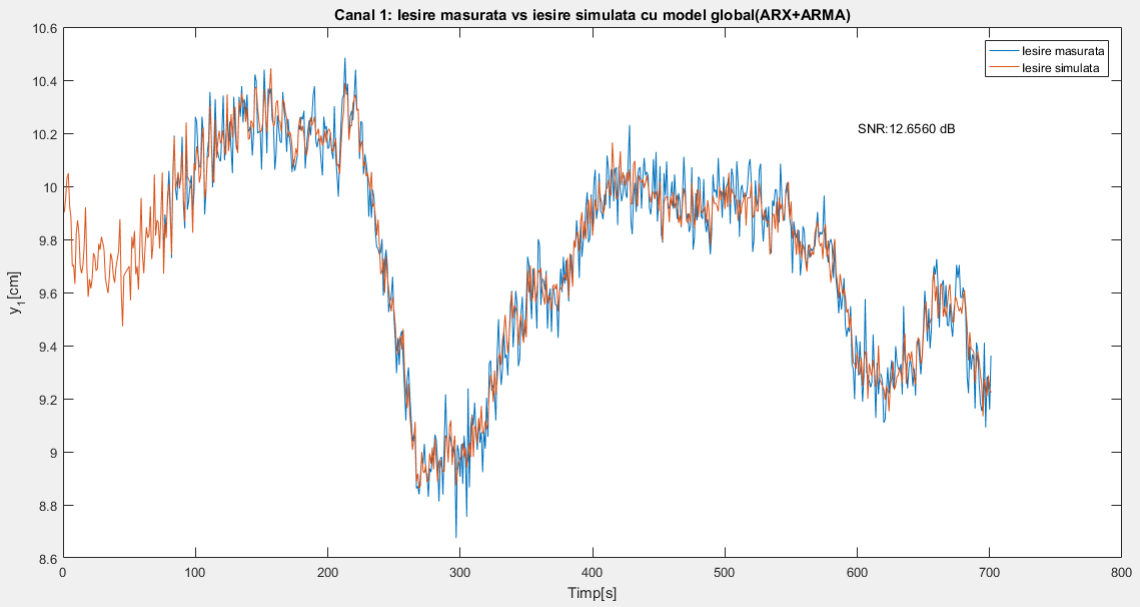

Figure 6.21: Canal 1: Ieşire măsurată vs ieşire simulată cu model global

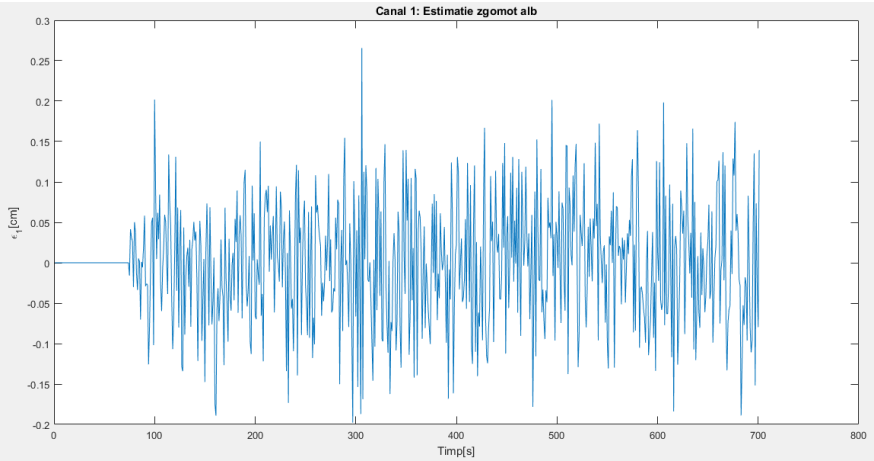

Figure 6.22: Zgomot de identificare iesire 1

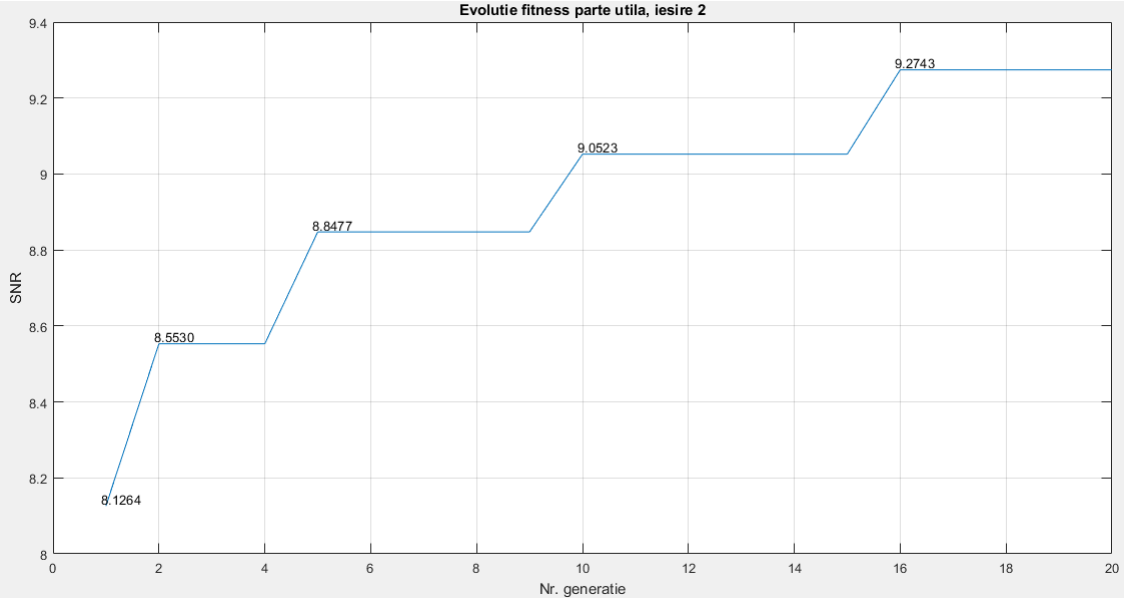

Figure 6.23: Evolutie fitness parte utila iesire 2

În cazul părţii utile a modelului nivelului din al doilea bazin, evoluţia fitness-ului maxim este ilustrată în (6.22), valoarea maximă găsită fiind de **9.2743 dB**, pentru setul de indici structurali [**na=23 nb1=21 nb2=27**].
Filtrul de zgomot care maximizează SNR-ul întreg semnalului simulat este obţinut pentru indicii structurali [**nd=81 nc=77**], iar valoarea SNR este **15.2971 dB**.

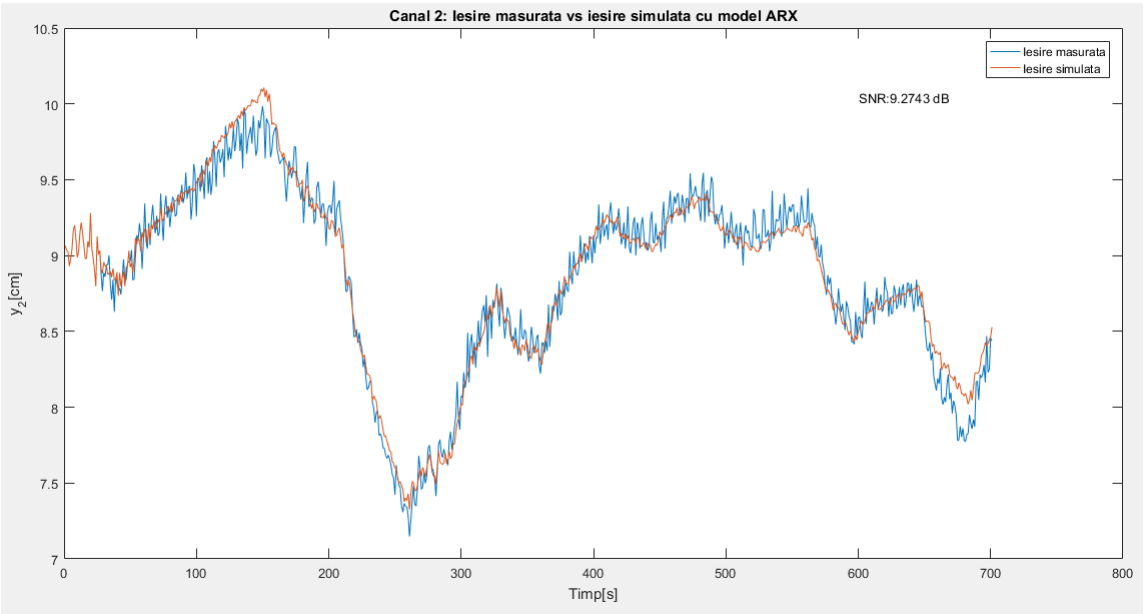

Figure 6.24: Canal 2: Ieşire măsurată vs ieşire simulată cu model ARX

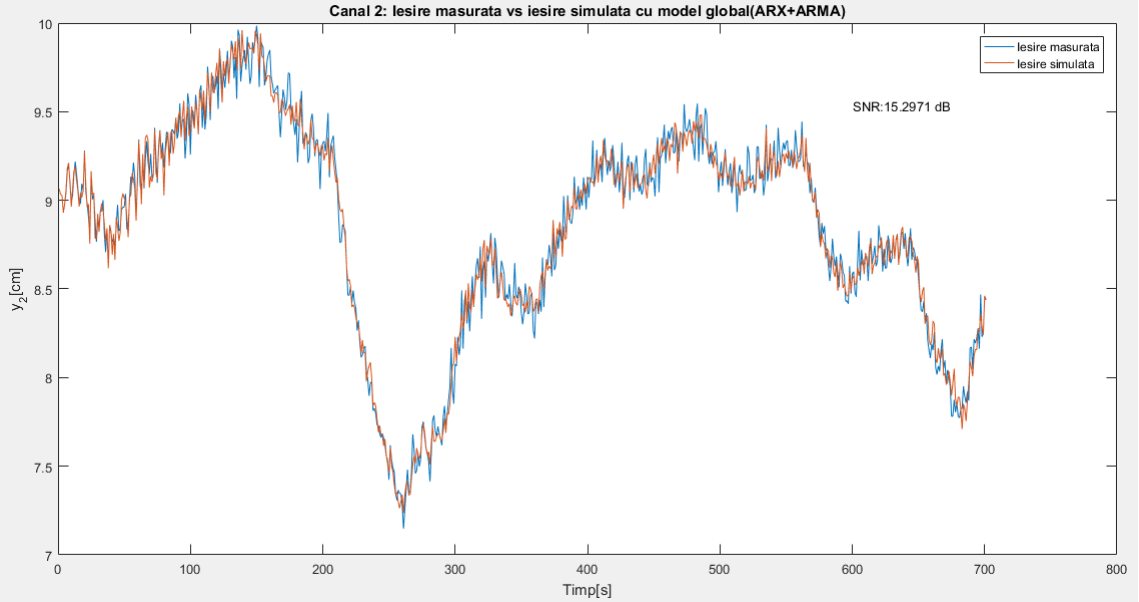

Figure 6.25: Canal 2: Ieşire măsurată vs ieşire simulată cu model global

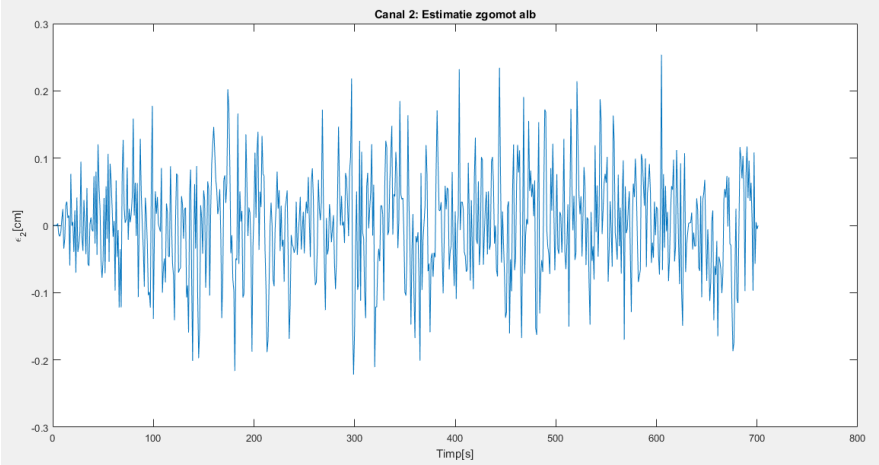

Figure 6.26: Zgomot de identificare iesire 2

6.1.5 Comparaţie algoritmi

Pentru a observa mai bine comportamentul celor doi algoritmi (consistenţa rezultatelor de la o rulare la alta, capacitatea de explorare, respectiv cea de exploatare), aceştia au fost rulaţi de mai multe ori.

Mai jos sunt ataşate rezultate ale căutării optimului pentru partea utilă în scop comparativ, în diferite spaţii de căutare, de diferite dimensiuni:

Spatiu de cautare	Algoritm	Indici structurali	Iesire	Fitness
[12 12 12]	FA	[10 2 8]	1	8.9132
[12 12 12]	FA	[2 3 1]	2	6.5527
[12 12 12]	GA	[11 10 2]	1	8.9645
[12 12 12]	GA	[2 3 1]	2	6.5527
[20 20 20]	FA	[17 12 6]	1	9.0300
[20 20 20]	FA	[9 1 16]	2	8.3580
[20 20 20]	GA	[17 4 17]	1	9.5164
[20 20 20]	GA	[8 2 16]	2	8.3215

Figure 6.27: Tabel rulări

Conform rezultatelor prezentate în tabel (6.27), se pot trage următoarele concluzii:

- Pentru spaţii de căutare suficient de mici ambii algoritmi vor găsi modele foarte bune, datorită timpului mai mic necesar realizării unei căutări exhaustive
- Algoritmul bazat pe licurici se foloseşte mai mult de explorare, prin urmare, se poate ca acesta să ofere rezultate mai bune decât algoritmul genetic, după cum s-a remarcat în urma căutărilor prezentate în (6.1.3) şi (6.1.4)
- Algoritmul genetic este mai consistent, oferind rezultate satisfăcătoare la fiecare rulare
- Algoritmul bazat pe licurici este capabil să găsească rezultate chiar mai bune decât algoritmul genetic, însă poate da greş mai des.

6.2 Proiectarea comenzii

Parametrii celor două regulatoare PID responsabile pentru urmărirea referinţei şi rejecţia perturbaţiilor au fost identificaţi tot printr-o tehnică metaeuristică, după cum este prezentat în (5). Pentru codificarea parametrilor, au fost aleşi 6 biţi pentru fiecare alelă. Algoritmul genetic a fost implementat asemănător cu cel folosit pentru identificarea modelului. Populaţia iniţială fost generată tot cu ajutorul algoritmului lui Baker.

Conform schemei din (5.5), identificarea parametrilor optimi ai regulatoarelor se face folosind dinamica modelului identificat. Rezultatele obţinute în buclă închisă sunt folosite pentru calcularea fitness-ului ce corespunde perechii de regulatoare. În (6.28) este prezentată schema Simulink corespunzătoare identificării parametrilor, respectiv schema ce ilustrează conexiunile celor 2 pompe cu cele 2 bazine.

Cele 4 funcţii de transfer discrete prezente în schemă modelează fiecare dinamică a ieşire - intrare din proces. Pentru identificarea parametrilor nu s-a luat în calcul decât partea utilă a modelelor identificate.
Au fost folosite 2 blocuri de saturaţie a ieşirilor, deoarece valorile negative ale nivelului de apă nu au sens. Regulatoarele PID au fost implementate tot sub formă de funcţie de transfer discretă, fiecare dintre ele fiind responsabil cu staţionarizarea erorii în câte o buclă de reglare.

Ponderile funcţiei criteriu din (5.7) trebuie alese de către proiectant în funcţie de forma dorită a răspunsului indicial. Cei 3 indici de performanţă pe baza cărora se face optimizarea sunt suprareglajul, timpul de creştere şi timpul tranzitoriu. Scopul regulatoarelor căutate este de a urmări referinţa suficient de rapid, pastrând, totodată, un suprareglaj redus. În mod natural, valorile ponderilor timpilor de creştere şi tranzitoriu ar trebui să fie cu atât mai mici cu cât procesul este mai lent.
În cazul procesului de faţă, timpul tranzitoriu este de aproximativ 200-250 secunde. Pentru o reducere cu 50 de secunde a timpului tranzitoriu, un suprareglaj cu 5 procente mai mare reprezintă un compromis acceptabil. De asemenea, conform [3], ponderea pentru timpul tranzitoriu se poate alege de 3 ori mai mare decât cea pentru timpul de creştere. Astfel, valorile alese pentru cele 3 ponderi din (5.7) sunt: $\boldsymbol{w_1 = 0.88}$, $\boldsymbol{w_2 = 0.09}$, $\boldsymbol{w_3 = 0.03}$.

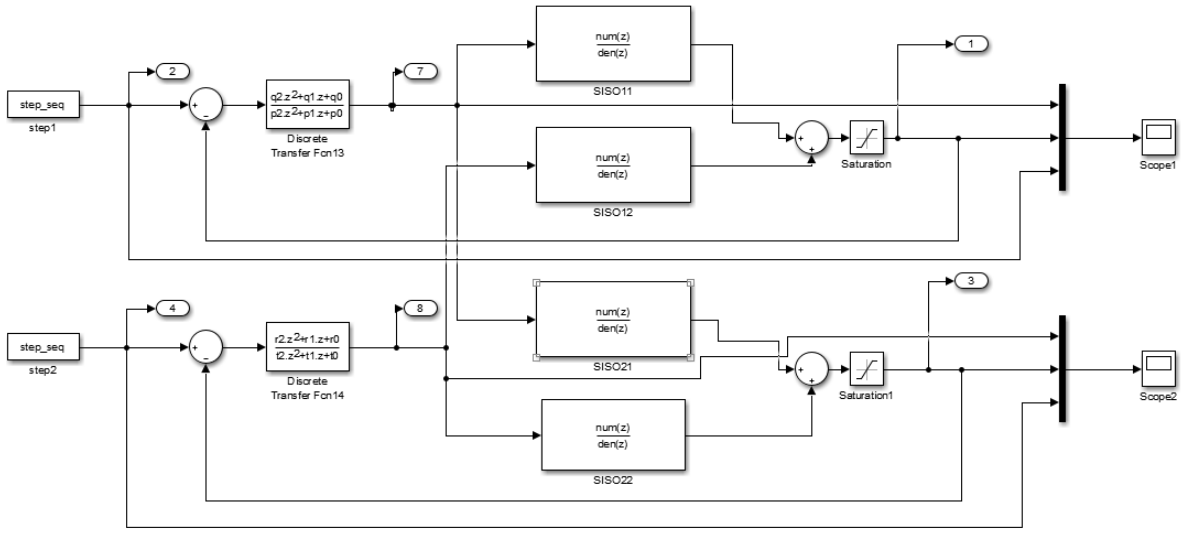

Figure 6.28: Schema de control

Pentru a simplifica procesul de căutare a parametrilor PID, au fost alese 2 modele cu indici structurali mai mici, păstrându-se, totodată, o apropiere satisfăcătoare a modelelor de procesul real. Astfel, polinoamele părţii utile, folosite pentru identificarea celor două regulatoare PID sunt:

$$A_1(z^{-1}) = 1 - 1.262z^{-1} + 0.3794z^{-2} - 0.2731z^{-3} + 0.1182z^{-4} - 0.001059z^{-5} - 0.1054z^{-6} +$$

$$0.2272z^{-7} - 0.1462z^{-8} + 0.09446z^{-9} - 0.1955z^{-10} + 0.1683z^{-11}$$

$$B_{11}(z^{-1}) = -0.000172z^{-1}+0.02248z^{-2}+0.02239z^{-3}-0.05996z^{-4}+0.02913z^{-5}-0.04914z^{-6}+$$
$$0.04197z^{-7} - 0.02665z^{-8} + 0.06131z^{-9} - 0.03964z^{-10}$$

$$B_{12}(z^{-1}) = -0.03789z^{-1}+0.04715z^{-2}-0.005016z^{-3}-0.04278z^{-4}+0.0391z^{-5}+0.008225z^{-6}+$$
$$0.02026z^{-7} - 0.04589z^{-8} + 0.02064z^{-9}$$

$$A_2(z^{-1}) = 1 - 1.226z^{-1} + 0.03877z^{-2} - 0.2438z^{-3} + 1.158z^{-4} - 0.7117z^{-5}$$

$$B_{21}(z^{-1}) = -0.0004325z^{-1}-0.03159z^{-2}+0.03693z^{-3}+0.02989z^{-4}-0.03213z^{-5}-0.009769z^{-6}+$$
$$0.01514z^{-7} - 0.03496z^{-8} + 0.02684z^{-9}$$

$$B_{22}(z^{-1}) = 0.04656z^{-1} - 0.07365z^{-2} + 0.07072z^{-3} - 0.02001z^{-4} + 0.03516z^{-5} - 0.1141z^{-6}+$$
$$0.09288z^{-7} - 0.01776z^{-8} + 0.05582z^{-9} - 0.06639z^{-10} + 0.009004z^{-11}$$

Filtrul de zgomot este adăugat conform figurii (6.29), iar zgomotul alb care este aplicat la intrarea acestuia are dispersia calculată pe baza modelului de zgomot. Eroarea de predicţie dintre ieşirea filtrului de zgomot şi zgomotul colorat de identificare (diferenţa dintre ieşirea reală procesului şi partea utilă a ieşirii simulate) reprezintă zgomotul alb. Varianţa zgomotului alb are valoarea **0.0072** pentru primul model şi **0.0070** pentru al doilea model.

De asemenea, o perturbaţie de 40% pentru primul model, respectiv pentru al doilea model a fost aplicată pentru a testa capacitatea de rejecţie a sistemului de control.

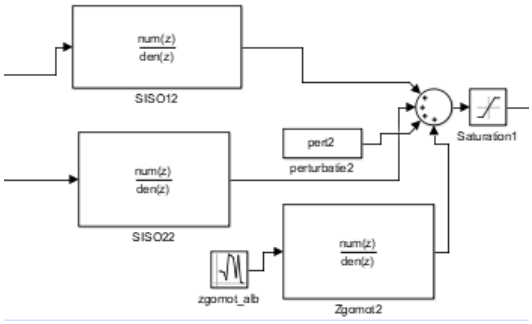

Figure 6.29: Perturbatie + zgomot

Coeficienţii regulatoarelor identificate sunt următorii:
$\boldsymbol{K_r = 0.3582, Ti = 21.3725s, Td = 0.019s}$, pentru primul PID
$\boldsymbol{K_r = 0.3137, Ti = 8.6275s, Td = 0.0196s}$, pentru al doilea PID

În graficele (6.30) - (6.35), valorile de pe axa Oy sunt exprimate în procente. **100%** reprezintă **10*V*** pentru valoarea comenzii , respectiv **10*cm*** pentru valoarea nivelului din al doilea bazin. Pentru nivelul din primul bazin, 100% reprezintă **11*cm***.

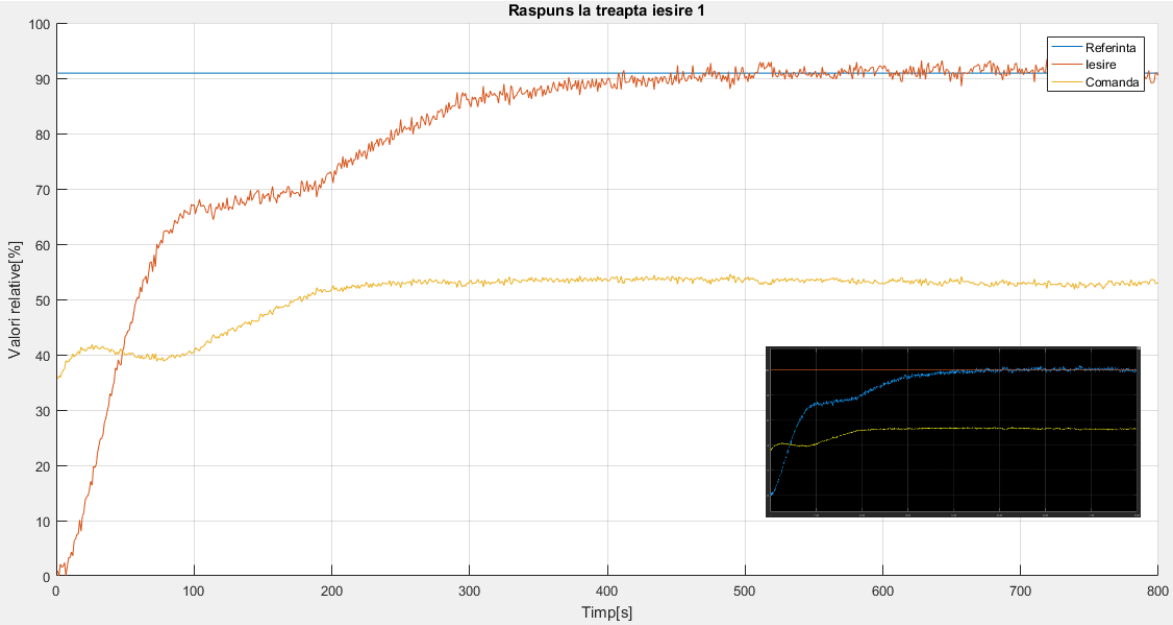

Figure 6.30: Răspuns la treaptă ieşire y_1

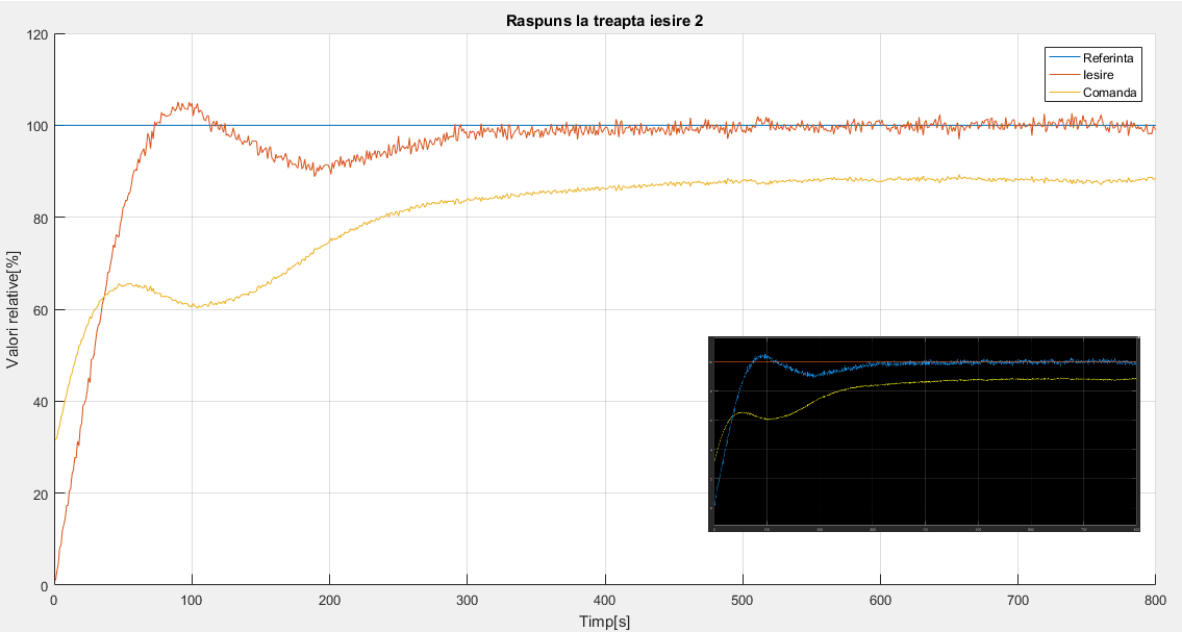

Figure 6.31: Răspuns la treaptă ieşire y_2

Valorile indicatorilor de performanţă ce ne interesează sunt următoarele:
$t_t = 290s, t_c = 210s, \sigma = 1.6\%$ pentru evoluţia primului nivel.
$t_t = 295s, t_c = 50s, \sigma = 5.75\%$ pentru evoluţia celui de-al doilea nivel.

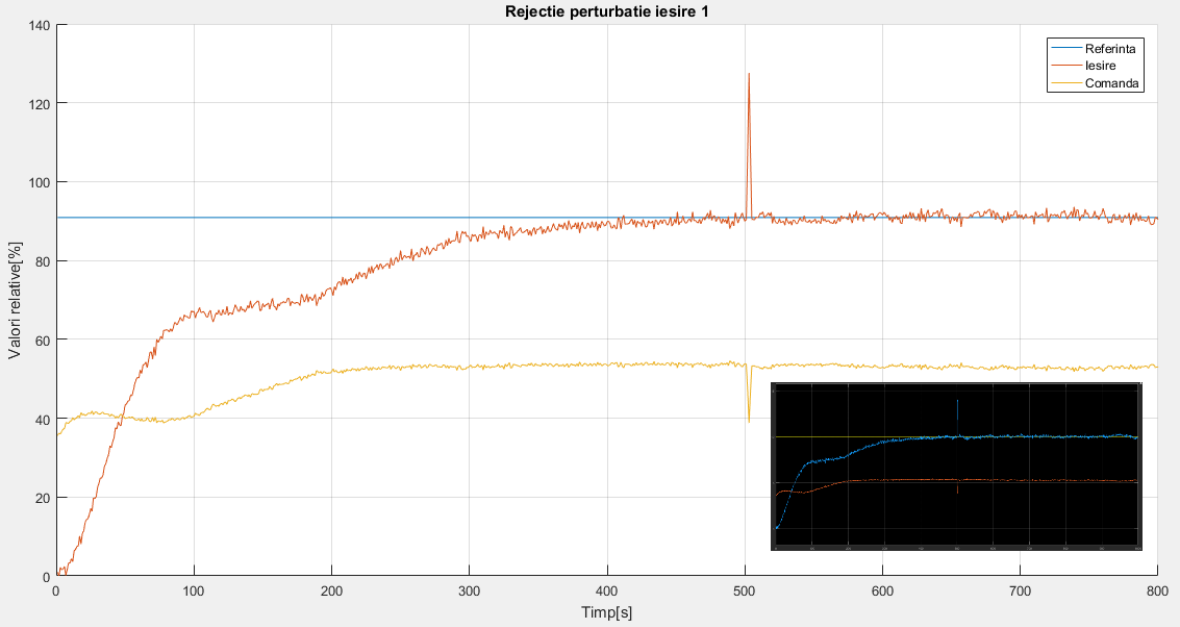

Figure 6.32: Rejecţie perturbaţie ieşire y_1

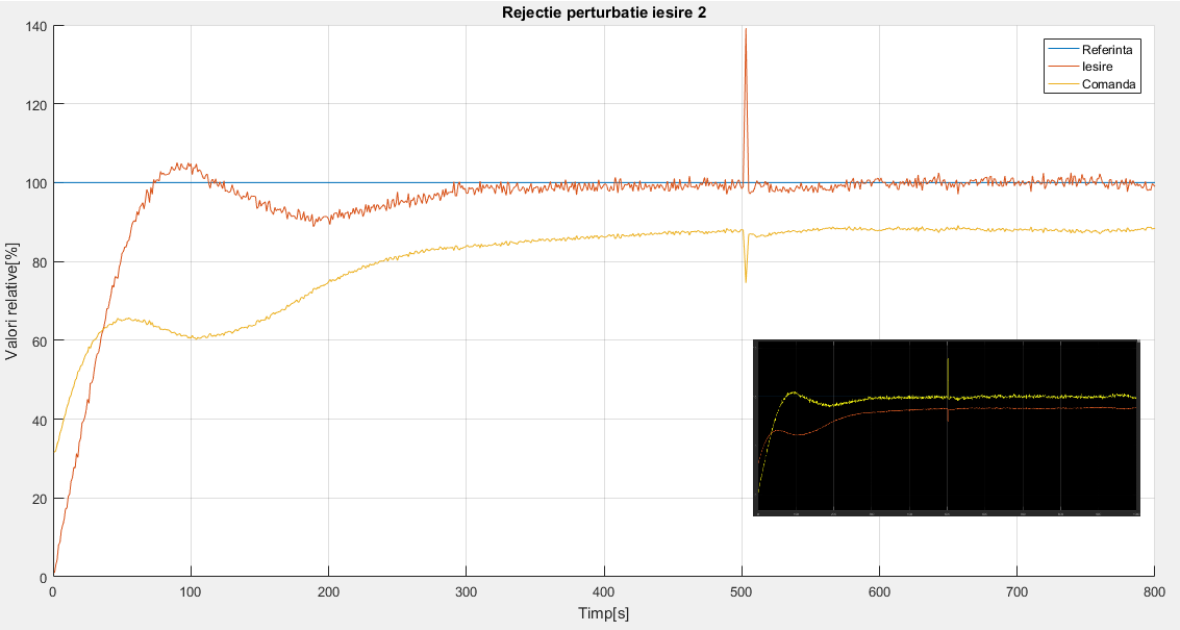

Figure 6.33: Rejecţie perturbaţie ieşire y_2

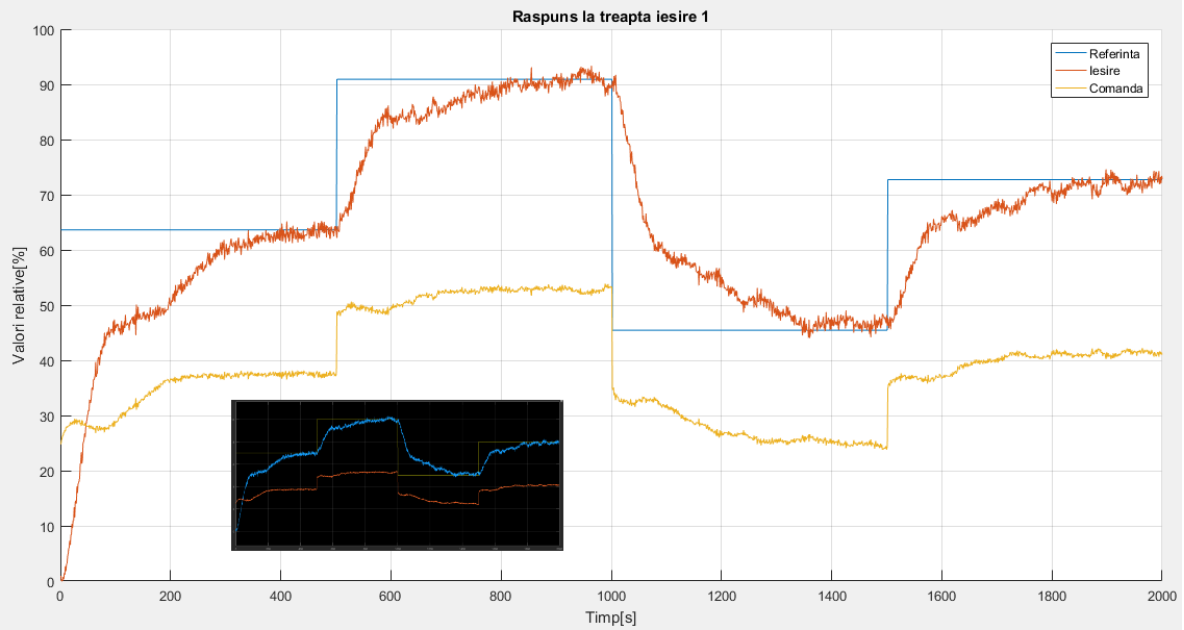

Figure 6.34: Succesiune trepte ieşire y_1

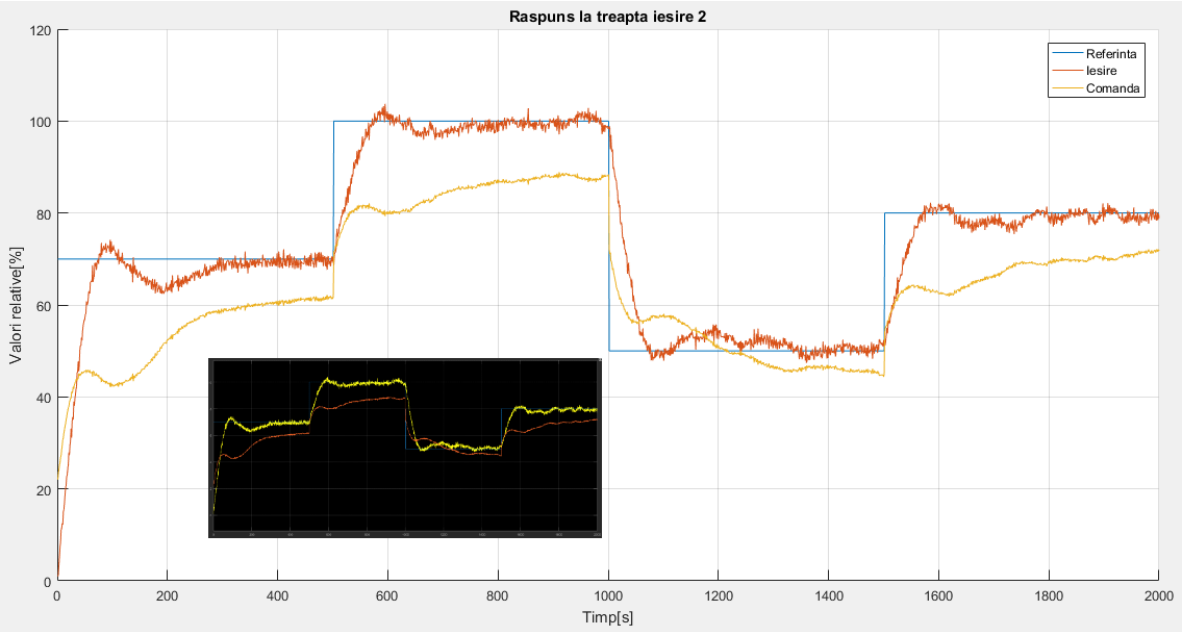

Figure 6.35: Succesiune trepte ieşire y_2

7 Concluzii

În cadrul acestei lucrări s-a cŭrmărit identificarea unui model matematic din clasa **RSISO** care să surprindă cât mai fidel dinamica functionarii instalaţiei
ASTANK2 si proiectarea legii de comanda care sa asigure performante impuse in bucla inchisa si rejectia perturbatiilor. S-a ales ca estimarea modelului de identificare să se realizeze în două etape, rezolvand succesiv doua probleme de optimizare: maximizarea SNR-ului calculat pentru partea utilă modelului şi respectiv maximizarea SNR-ului aferent filtrului de zgomot rezultat din estimarea zgomotului colorat de la etapa anterioara (prin diferenţa dintre ieşirea reală şi cea simulată cu modelul util).

Au fost implementate două metaeuristici, pentru a optimiza structura unui model util de tip ARX şi a unui model de zgomot de tip ARMA. Totodata, s-a realizat o analiza comparativa a performantelor oferite de cele doua metaheuristici. Modelele astfel identificate au fost folosite pentru a proiecta algoritmul de control al instalaţiei, care să respecte performanţele precizate. Au fost realizate mai multe simulări ai celor doi algoritmi în scop comparativ. S-a observat că algoritmul genetic este mai fiabil, obţinându-se valori satisfăcătoare la fiecare rulare. Pe de altă parte, pentru spaţii de căutare mai mari, algoritmul bazat pe licurici a dat rezultate mai bune datorită caracteristicii sale exploratorii. Cu toate astea, algoritmul bazat pe licurici s-a dovedit mai predispus la a eşua în a găsi rezultate satisfcătoare, la anumite rulări.

S-a urmărit controlul instalaţiei folosind mediul de programare Matlab şi mediul de programare grafică Simulink. Regulatoarele PID folosite pentru control au fost proiectate plecând de la modelele identificate, folosind algoritmi genetici pentru optimizarea coeficienţilor acestora. Bucla de reglare pentru controlul nivelului în bazine este alcătuită din 2 astfel de regulatoare PID, procesului fiind multivariabil cu 3 intrări, dintre care una constantă.

O perspectivă viitoare de cercetare o reprezintă identificarea unor modele pe stare care să permită proiectarea de legi de control mai performante, precum controlul optimal sau predictiv. De asemenea, o posibilă îmbunătăţire asupra legii de control ar fi proiectarea de regulatoare robuste care să permită controlul unor modele ce cuprind diferite tipuri de incertitudini. O altă direcţie viitoare de cercetare o reprezintă identificarea în buclă închisă a modelului cu tot cu proiectare de regulator.

Mulţumiri

În final, aş dori să îi mulţumesc profesorului coordonator de proiect, Conf.Dr.Ing. Janetta Culiţă pentru sprijin, sfaturi preţioase şi informaţii utile. De asemenea, doresc să îi mulţumesc domnului Prof.Dr.Ing. Dan Ştefănoiu pentru referinţele bibliografice furnizate şi pentru comentariile utile.

Bibliografie

[1] Janetta Culita, Dan Stefanoiu, and Alexandru Dumitrascu. Astank2: analytical modeling and simulation. In *2015 20th International Conference on Control Systems and Computer Science*, pages 141-148. IEEE, 2015.

[2] Janetta Culita and Dan Stefanoiu. Multi-model identication of pumping system in astank2 plant. In *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, pages 59-66. IEEE, 2017.

[3] Jyoti Ohri , Naveen Kumar, Minakshi Chinda. An Improved Genetic Algorithm for PID Parameter Tuning. In *Proceedings of the 2014 International Conference on Circuits, Systems, Signal Processing, Communications and Computers (CSSCC '14)*, pages 191-198.

[4] MOHD S. SAAD, HISHAMUDDIN JAMALUDDIN, INTAN Z. M. DARUS. PID Controller Tuning Using Evolutionary Algorithms. In *WSEAS Transactions on Systems and Control*, pages 139-149.

[5] Yongsheng Fang and Jun Li. A Review of Tournament Selection in Genetic Programming. In *Advances in Computation and Intelligence*, pages 181-192. Springer, 2010.

[6] Nur Farahlina Johari, Azlan Mohd Zain, Noorfa Haszlinna Mustaffa, Amirmudin Udin. Firefly Algorithm for Optimization Problem. In *Applied Mechanics and Materials Vol. 421 (2013)*, pages 512-517.

[7] Borne, P.; Popescu, D.; Filip, F.G. ; Ştefănoiu, D. *Optimization in Engineering Sciences Exact Methods*, 1st ed., London, U.K.: John Wiley & Sons & ISTE Press, 2013

[8] C. Lupu C. Petrescu B. Ciubotaru C. Dimon D. Popescu, D. ŞTEFĂNOIU. *Automatică Industrială*, Editura AGIR, 2006.

[9] D. Ştefănoiu, J.Culiţă, P.Stoica. *Fundamentele modelării şi identificării sistemelor*, Ed. Printech, 2005

[10] J. Culiţă, D. Ştefănoiu. *Modelarea analitică şi experimentală a sistemelor*, Ed. Printech, 2008.

[11] I. Dumitrache. *Ingineria reglării automate*, Editura Politehnica Press, 2010