

11. ACORDAREA GENETICĂ A REGULATOARELOR PENTRU PROCESE CU MAI MULTE INTRĂRI ȘI CU MAI MULTE IEȘIRI

Procesele cu mai multe intrări și cu mai multe ieșiri (multiple-input, multiple output – MIMO), atunci când sunt conduse cu legi de reglare din clasa PID, necesită utilizarea a multiple regulatoare în diverse combinații. Interdependența dintre subprocese componentele poate conduce la complexitate ridicată în proiectarea și acordarea acestor regulatoare, motiv pentru care se pretează utilizarea unor tehnici multi-criteriale de optimizare în timpul acordării.

Scopul lucrării este acordarea unui set de regulatoare de tip PID pentru un robot uniciclu utilizând algoritmi genetici.

11.1 Breviar teoretic

Robotul mobil uniciclu, prezentat în lucrarea 2, este un proces MIMO care are ca intrări comenzile de translație și de rotație e_m și e_d (din care se obțin comenzile efective pentru cele două motoare, u_L și u_R). Pentru lucrarea de față, ieșirile sistemului sunt viteza de translație și viteza de rotație. După cum s-a observat în lucrarea 9, acordarea manuală a unui set de regulatoare pentru controlul acestor viteze este deosebit de dificilă și deloc intuitivă.

În cele ce urmează, se vor utiliza algoritmi genetici pentru acordarea a două regulatoare PID. Structura de reglare este ilustrată în figura 11.1. Regulatoarele PID_v și PID_ω date în formă paralel sunt

$$PID_v = K_{Rv} \left(1 + \frac{1}{T_{iv}s} + \frac{T_{dv}s}{\alpha_v T_{dv}s + 1} \right), \quad (11.1a)$$

$$PID_\omega = K_{R\omega} \left(1 + \frac{1}{T_{i\omega}s} + \frac{T_{d\omega}s}{\alpha_\omega T_{d\omega}s + 1} \right), \quad (11.1b)$$

unde coeficienții de filtrare $\alpha_v, \alpha_\omega \ll 1$ sunt pozitivi.

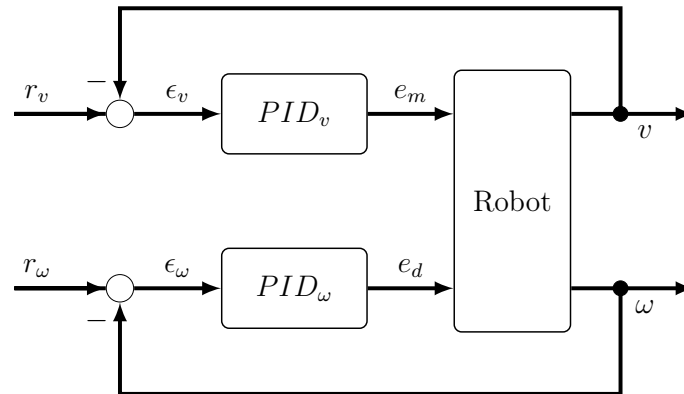


Figura 11.1: Structura de reglare a vitezei de rotație și de translație

Cele două regulatoare urmăresc obținerea unui set de performanțe specifice pentru cele două mărimi reglate (răspunsuri aperiodice la referință treaptă, cu eroare staționară zero și timpi tranzitorii aproximativ egali, de exemplu, 5 sec.). Obiectivele lor sunt independente, însă parametrii de acord ai regulatoarelor, din cauza cuplării inerente a proceselor de translație și de rotație, depind unii de ceilalți. Modificarea parametrilor unui regulator va duce la modificări în răspunsul celeilalte bucle de reglare și reciproc.

Criterii multiple în algoritmi genetici Unul dintre avantajele majore ale algoritmilor evoluționiști, și astfel, a algoritmilor genetici, este faptul că aceștia pot realiza căutări urmărind îndeplinirea concomitentă a mai multor criterii. Modelarea obiectivelor multicriteriale în cadrul algoritmilor genetici se poate face prin

- a) fuziunea criteriilor, sau prin
- b) separarea acestora.

Separarea criteriilor presupune ca toate obiectivele algoritmului genetic, i.e., funcțiile de optimizat F_1, F_2, \dots, F_n să compună formal o funcție $F = G(F_1, F_2, \dots, F_n)$, unde G este un operator care poate fi sumă, combinație liniară, operații logice, ș.a.m.d. Operatorul G trebuie formulat în așa fel încât optimul global al funcției F să reprezinte și optimele funcțiilor F_1, F_2, \dots, F_n . De obicei, soluția globală este un vector compus din soluțiile criteriilor individuale.

Fuziunea criteriilor se realizează prin găsirea unui criteriu comun tuturor funcțiilor de optimizat individuale F_1, F_2, \dots, F_n . Obiectivul global al algoritmului nu va conține explicit reprezentările criteriilor individuale. Această metodă se utilizează atunci când criteriile individuale sunt dificil de formulat sau necunoscute. Problema de optimizare devine o problemă definită pe mai multe dimensiuni, iar interacțiunea

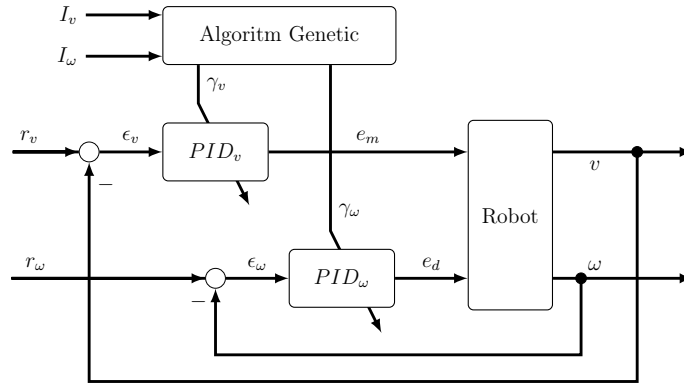


Figura 11.2: Acordarea reguletoarelor PID cu algoritmi genetici

subsistemelor componente ale problemei se încastrează în reprezentări de tip black-box.

Acordarea reguletoarelor pentru structuri MIMO Pentru problema de față, structura de reglare cu acordare utilizând algoritmi genetici este dată în figura 11.2.

Cromozomul care codifică soluția problemei de reglare conține toți parametrii de acord ai reguletoarelor.

1	2	3	4	5	6
K_{Rv}	T_{iv}	T_{dv}	$K_{R\omega}$	$T_{i\omega}$	$T_{d\omega}$

Tabelul 11.1: Structura cromozomului

Criteriile de îndeplinit, pentru cele două reguletoare, sunt I_v și I_ω de tip integral. Să presupunem că utilizăm varianta ISE

$$I_v = \int_0^\infty \epsilon_v^2(t) dt, \quad (11.2a)$$

$$I_\omega = \int_0^\infty \epsilon_\omega^2(t) dt. \quad (11.2b)$$

Prin separare, noul indice devine, de exemplu

$$I = \int_0^\infty \epsilon_v^2(t) dt + \int_0^\infty \epsilon_\omega^2(t) dt. \quad (11.3)$$

Minimul criteriului I se obține atunci când ambele criterii I_v și I_ω sunt de asemenea minime.

Fuziunea criteriilor este imposibilă în cazul de față, deoarece referințele buclilor de reglare sunt distincte. Pentru fuziune trebuie considerată o altă structură de reglare, cu alt scop final, alte intrări, și alte ieșiri (mărimi reglate).

11.2 Sarcini de lucru

1. Se realizează schema Simulink a structurii de reglare din figura 11.1, obținându-se modelul *modelSRA.mdl* (consultați schema Simulink realizată în lucrarea 10).
2. Se scrie algoritmul genetic care va căuta parametrii celor două regulatoare.
 - 2.1. Urmărind pașii descriși în lucrarea 10, se creează funcția *algoritmulmeu.m*, cu diferența că vectorii *variableBounds* și *bounds* reflectă dimensiunea curentă a cromozomului (i.e., 6 elemente în loc de 3).
 - 2.2. Se creează funcția *functieobiectiv.m*

```
function[x_pop,fx_val]=functieobiectiv(x_pop,options).
```

– Se setează regulatorul PID cu parametrii dați de individul curent.

```
Krv=x_pop(1);
Tiv=x_pop(2);
Tdv=x_pop(3);
Krw=x_pop(4);
Tiw=x_pop(5);
Tdw=x_pop(6);
assignin('base','Krv',x_pop(1));
assignin('base','Tiv',x_pop(2));
assignin('base','Tdv',x_pop(3));
assignin('base','Krw',x_pop(4));
assignin('base','Tiw',x_pop(5));
assignin('base','Tdw',x_pop(6));
```

– Se evaluează răspunsul la intrări de tip treaptă.

```
[T,state,outputs] = sim('modelSRA',tsim);
%tsim reprezinta timpul de simulare
%convenabil ales
erv = outputs(:,1);
yv = outputs(:,2);
refv = outputs(:,3);
tv=outputs(:,4);
erw = outputs(:,6);
yw = outputs(:,7);
refw = outputs(:,8);
tw=linspace(0,tsim,length(refw));
```

Observația 11.1. Parametrii y_v , erv și ref_v , respectiv y_w , erw și ref_w se obțin din modelul *modelSRA.mdl* corect prin numerotarea corespunzătoare a porturilor de ieșire.

– Se evaluează eroarea pătratică medie de urmărire a referinței utilizând ecuația (11.3).

$IND = erv' * erv + erw' * erw;$

Observația 11.2. Indivizii care nu îndeplinesc cerințele de performanță (răspunsuri aperiodice și $t_{tv} \approx t_{tw} \approx 5\text{sec.}$), trebuie penalizați prin atribuirea unei valori foarte mari.

```

ytv1=find(erv>0.05, 1, 'last');
ytv2=find(erv<-0.05, 1, 'last');
ytw1=find(erw>0.05, 1, 'last');
ytw2=find(erw<-0.05, 1, 'last');
SettlingTimeV = tv(max([ytv1 ytv2]));
SettlingTimeW = tw(max([ytw1 ytw2]));
OvershootV = abs(min(min(erv),0));
OvershootW = abs(min(min(erw),0));
if (Overshoot > 0 & SettlingTimeV > 5
    & SettlingTimeW > 5
    & abs(SettlingTimeV-SettlingTimeW) > 0.1)
IND = 10e300;
end

```

– Se calculează fitness-ul individului (procedurile din toolbox-ul GAOT realizează maximizare, așadar, inversarea valorii este necesară pentru minimizare în problema de față).

$fx_val=1/IND;$

3. Se rulează funcția *algoritmuleu.m*.

Observația 11.3. În cazul în care algoritmul nu găsește o soluție convenabilă, se vor ajusta intervalele de căutare, atât la inițializare, cât și la rularea procedurii de optimizare. Pentru utilizatorii neexperimentați, se recomandă inițial setarea aceluiași intervale.

4. Se va realiza o comparație între rezultatele obținute prin această optimizare și rezultatele obținute în lucrarea 9.

5. În cadrul cromozomului, se adaugă și coeficienții de filtrare α_v , α_w și se modifică în mod corespunzător algoritmul. Se rulează din nou procedura de optimizare și se realizează o comparație cu rezultatele pasului 4.