

Laborator 1

Reglarea manuală. Structuri de reglare convenționale.

1.1 Tema

Aceasta lucrare se constituie într-o primă aplicație a cunoștințelor introductive de programare în mediul LABVIEW în domeniul reglării. Se vor verifica posibilitățile studenților de a realiza o reglare manuală performantă a unor procese cu dinamică diferită precum și performanțele ce se pot obține cu structuri convenționale de reglare (i.e. sisteme SISO cu legi de reglare tipizate) a acelorași procese, folosind tehnici experimentale de acordare a reglatoarelor.

1.2 Considerații preliminare

Punctul de vedere al **teoriei** sistemelor pentru automatizarea unor procese industriale complexe presupune proiectarea unui compensator multivariabil de o complexitate comparabilă pe baza unui model matematic suficient de exact al procesului, model care se obține, în general, în urma unui proces de identificare costisitor și dificil. Proiectarea compensatorului multivariabil necesită ea însăși un efort deosebit de ridicat. Mai mult, compensatorul multivariabil este un produs unicat, destinat exclusiv conducerii procesului pentru care a fost proiectat iar operarea sa și stăpânirii unor situații de avarie etc. sunt foarte dificile datorită complexității.

De aceea, în **practica** reglării proceselor multivariabile, se utilizează, aproape în exclusivitate, reglarea descentralizată care, cel puțin într-o primă fază, privește procesul multivariabil ca o colecție de procese monovariabile (SISO: Single Input Single-Output). Sistemele de reglare automată realizate într-o astfel de viziune ar fi o colecție de sisteme SISO convenționale care beneficiază de multiple avantaje:

1. utilizarea unor controllere (algoritmi de reglare) tipizate, de uz curent;
2. existența unor proceduri de acordare experimentală;
3. o experiență uriașă în operarea unor astfel de sisteme (inclusiv posibilități de comandă manuală) etc.

De aceea, prezenta lucrare de laborator își găsește o justificare deplină în contextul pregătirii ingineriști pentru exploatarea sistemelor de automatizare industriale.

1.3 Reglarea manuală

Reglarea manuală se poate constitui într-un exercițiu util pentru studenții din mai multe rațiuni:

- permite cunoașterea ”pe viu” a noțiunii de ”dinamică” a procesului;
- permite înțelegerea ideii de ”acordare a controllerului” (fiecare proces se conduce altfel);
- permite evidențierea proceselor ”greu de reglat” (proces de ordin superior, cu timp mort, cu interacțiuni între canale etc.); în general, procesele greu de condus manual sunt greu de condus și automat;
- înțelegerea caracterului permanent al reglării etc.

În esență, programul LabVIEW pentru efectuarea reglării manuale trebuie să asigure:

- calculul și afișarea permanentă a mărimii reglate;
- afișarea evoluției mărimii reglate și (eventual) a mărimii de comandă (accesul la istoria desfășurării reglării permite obținerea unor performanțe mai bune);
- posibilitatea de a modifica on-line mărimea de comandă;
- modificarea la momente aleatoare a perturbației (independent de acțiunea de reglare);

iar experimentul constă în modificarea mărimii de comandă astfel încât mărimea reglată să ajungă la o valoare impusă (referința) și să stea acolo!

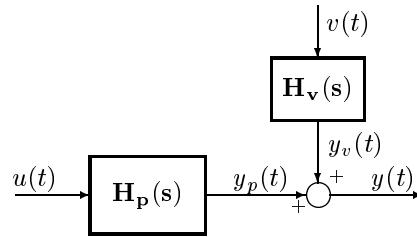


Figure 1.1: Modelul structural al procesului

Pentru a fi util experimentul de reglare manuală trebuie repetat pe procese diferite.

Pentru a simplifica elaborarea programelor LabVIEW propunem ca pe canalul de perturbație avem o dinamică fixă descrisă de funcția de transfer de ordiinel 1:

$$H_v(s) = \frac{k_v}{T_v s + 1} = \frac{0.1}{2s + 1} \quad (1.1)$$

sau în forma discretizată (vezi cursurile de Teoria Sistemelor sau de Calcul Numeric în Automatică)

$$H_v^d(z) = (1 - z^{-1}) \mathcal{Z} \left[\frac{H_v(s)}{s} \right] = \frac{k_v * (1 - e^{-\frac{h}{T_v}})}{z - e^{-\frac{h}{T_v}}} = \frac{0.1 * (1 - e^{-0.5h})}{z - e^{-0.5h}} \quad (1.2)$$

unde h este perioada de eșantionare. Prin urmare componenta perturbatoare la ieșire se va calcula cu

$$y_v(k) = a_v * y_v(k-1) + b_v * v(k-1), \quad y_v(0) = 0, \quad k = 0, 1, \dots \quad (1.3)$$

unde

$$a_v = e^{-0.5*h}, \quad b_v = 0.1 * (1 - e^{-0.5h}).$$

De asemenea, pentru uniformitate, se va considera în toate cazurile $h = 0.5$ secunde.

Pentru canalul de comandă propunem luarea în considerare a următoarelor cazuri:

1. Proces de ordinul 1.

Funcția de transfer pe canalul de comandă va fi în acest caz

$$H_p(s) = \frac{k_p}{T_p s + 1} = \frac{\sqrt{2}}{T_p s + 1} \quad (1.4)$$

considerându-se succesiv 3 valori distincte pentru T_p și anume $T_p = 1$ sec., $T_p = 10$ sec. și $T_p = 100$ sec..

Relațiile de calcul vor fi (vezi mai sus), evident,

$$y_p(k) = a * y_p(k-1) + b * u(k-1), \quad y_p(0) = 0, \quad k = 1, 2, \dots \quad (1.5)$$

unde

$$a = e^{-\frac{h}{T_p}}, \quad b = \sqrt{2} * (1 - e^{-\frac{h}{T_p}}).$$

și

$$y(k) = y_p(k) + y_v(k), \quad k = 0, 1, \dots \quad (1.6)$$

2. Proces de ordinul 2.

În acest caz, comportarea pe canalul de comandă va fi descrisă de funcția de transfer

$$H_p(s) = \frac{k_p}{(T_1 s + 1)(T_2 s + 1)} = \frac{\sqrt{2}}{(T_1 s + 1)(T_2 s + 1)} \quad (1.7)$$

considerându-se o valoare fixă de $T_1 = 5$ sec și, succesiv, 3 valori distincte pentru T_2 și anume $T_2 = 1$ sec., $T_2 = 5$ sec. și $T_2 = 25$ sec.

În cazurile în care $T_1 \neq T_2$ funcția de transfer discretizată se obține astfel

$$\begin{aligned} H_p^d(z) &= (1 - z^{-1}) \mathcal{Z} \left[\frac{H_p(s)}{s} \right] = (1 - z^{-1}) \mathcal{Z} \left[\frac{k_p}{s(T_1 s + 1)(T_2 s + 1)} \right] = \\ &= k_p (1 - z^{-1}) \mathcal{Z} \left(\frac{1}{s} + \frac{T_1}{T_2 - T_1} \frac{1}{s + 1/T_1} + \frac{T_2}{T_1 - T_2} \frac{1}{s + 1/T_2} \right) = \\ &= k_p (1 - z^{-1}) \left(\frac{1}{1 - z^{-1}} + \frac{T_1}{T_2 - T_1} \frac{1}{1 - z^{-1} e^{-h/T_1}} + \frac{T_2}{T_1 - T_2} \frac{1}{1 - z^{-1} e^{-h/T_2}} \right) = \\ &= k_p \left(1 + \frac{T_1}{T_2 - T_1} \frac{z - 1}{z - e^{-h/T_1}} + \frac{T_2}{T_1 - T_2} \frac{z - 1}{z - e^{-h/T_2}} \right). \end{aligned} \quad (1.8)$$

Notând $\alpha_1 = e^{-h/T_1}$, $\alpha_2 = e^{-h/T_2}$ și $\beta_1 = \frac{T_1}{T_2 - T_1}$, $\beta_2 = \frac{T_2}{T_1 - T_2}$ putem scrie

$$H_p^d(z) = k_p \left(1 + \beta_1 \frac{z - 1}{z - \alpha_1} + \beta_2 \frac{z - 1}{z - \alpha_2} \right) = \frac{b_1 z^2 + b_2 z + b_3}{z^2 - a_1 z - a_2} \quad (1.9)$$

unde

$$a_1 = \alpha_1 + \alpha_2, \quad a_2 = -\alpha_1 \alpha_2$$

și

$$b_1 = k_p, \quad b_2 = k_p(\beta_1 + \beta_2 - a_1), \quad b_3 = -k_p(\beta_1 + \beta_2 + a_2)$$

Relațiile de calcul vor fi, deci,

$$y_p(k) = a_1 * y_p(k-1) + a_2 * y_p(k-2) + b_1 * u(k) + b_2 * u(k-1) + b_3 u(k-2), \quad (1.10)$$

$$y_p(0) = y_p(1) = 0, \quad k = 2, 3, \dots$$

În situația în care $T_1 = T_2 = T$ relațiile de calcul se modifică în modul următor

$$\begin{aligned} H_p^d(z) &= (1 - z^{-1}) \mathcal{Z} \left[\frac{H_p(s)}{s} \right] = (1 - z^{-1}) \mathcal{Z} \left[\frac{k_p}{s(Ts + 1)^2} \right] = \\ &= k_p(1 - z^{-1}) \mathcal{Z} \left(\frac{1}{s} - \frac{1}{s + 1/T} - \frac{1}{T} \frac{1}{(s + 1/T)^2} \right) = \\ &= k_p(1 - z^{-1}) \left(\frac{1}{1 - z^{-1}} - \frac{1}{1 - z^{-1}e^{-h/T}} - \frac{1}{T} \frac{he^{-h/T}}{(1 - z^{-1}e^{-h/T})^2} \right) = \\ &= k_p \left(1 - \frac{z-1}{z - e^{-h/T}} - \frac{1}{T} \frac{z-1}{(z - e^{-h/T})^2} \right). \end{aligned} \quad (1.11)$$

Notând $\alpha = e^{-h/T}$, și $\beta = 1/T$, putem scrie

$$H_p^d(z) = k_p \left(1 - \frac{z-1}{z-\alpha} - \beta \frac{h(z-1)z}{(z-\alpha)^2} \right) = \frac{b_1 z^2 + b_2 z + b_3}{z^2 - a_1 z - a_2} \quad (1.12)$$

unde

$$a_1 = 2\alpha, \quad a_2 = -\alpha^2$$

și

$$b_1 = k_p(2 - \beta h), \quad b_2 = k_p(-3\alpha - 1 + \beta h), \quad b_3 = k_p(\alpha^2 + \alpha).$$

Componenta ieșirii datorată comenzii va fi calculată, de asemenea, cu relația 1.10 (dar cu noile expresii ale coeficienților), iar ieșirea completă, în ambele situații, cu relația:

$$y(k) = y_p(k) + y_v(k), \quad k = 0, 1, \dots \quad (1.13)$$

3. Proces de ordinul 1 cu timp mort.

Modelul liniar de ordinul 1 cu timp mort

$$H_p(s) = \frac{k_p * e^{-s\tau}}{T_p s + 1} \quad (1.14)$$

este o aproximare larg acceptată pentru descrierea dinamicii proceselor industriale stabile. Chiar dacă dinamica procesului nu conține efectiv un timp mort, modelul de mai sus este util în practică, timpul mort aproximând efectul constantelor de timp "mici", i.e. (mult) mai mici decât constanta de timp "principală" T_p . Într-un astfel de caz, într-o primă aproximație, valoarea timpului mort este dată de suma constantelor de timp "mici".

Lăsând în sarcina studenților stabilirea modelului discretizat (se va considera că timpul mort este un multiplu al perioadei de eșantionare) și a relațiilor de calcul al răspunsului, propunem

desfașurarea experimentelor de reglare manuală pentru următoarele valori ale parametrilor funcției de transfer:

$$k_p = \sqrt{2}, \quad T_p = 5 \text{ sec.}$$

și, succesiv, trei valori ale timpului mort:

$$\tau_1 = 0.5 \text{ sec.}, \quad \tau_2 = 1 \text{ sec.}, \quad \tau_3 = 2 \text{ sec.}$$

Anticipând desfășurarea experimentelor de simulare, amintim studenților că teoria și parac-tica reglării subliniază faptul că prezența timpului mort face dificilă obținerea unor performanțe superioare ale reglării. De fapt, ceea ce contează, în modelul sintetic considerat mai sus, este nu atât valoarea timpului mort cât valoarea raportului $\frac{\tau}{T_p}$. Cu cât valoarea acestui raport este mai mare cu atât reglarea este mai dificilă. De exemplu, pentru $\frac{\tau}{T_p} > 0.5$ obținerea unei reglări satisfăcătoare cu mijloace clasice (cum sunt controllerele PID) este practic imposibilă.

1.3.1 Reglare automată. Sisteme convenționale de reglare

Prin structură convențională de reglare se înțelege un sistem de reglare monovariabil (i.e. procesul, controllerul și sistemul de reglare sunt obiecte SISO) cu o lege de reglare tipizată (bipozițională, tripozițională, PID, i.e. curent disponibilă pe piață sub formă de echipamente sau programe de calculator) (vezi fig. 1.2).

Având în vedere că studiul aprofundat al sistemelor convenționale de reglare face obiectul unui alt curs, aici propunem studenților studiul lor prin simulare ca o etapă utilă în aprofundarea cunoștințelor de programare în mediul LabVIEW. De aceea, ne vom mărgini la legi de reglare PID (cu variantele ei mai simple P și PI) și exclusiv la tehnici experimentale de acordare ale acestora.

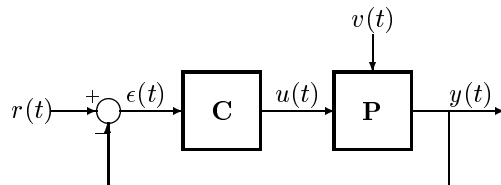


Figure 1.2: Structura unui sistem de reglare convențional.

Vom prezenta modelele matematice numai a legilor de reglare de tip PI și PID (cele mai des utilizate în practica reglărilor automate industriale) și algoritmi PI și PID obținuți prin discretizarea acestora.

Algoritm PI. Funcția de transfer a unui controller analogic PI este

$$H_c(s) = k_c \left(1 + \frac{1}{T_i s} \right), \quad (1.15)$$

cu parametrii de acordare factorul de amplificare k_c și (constantă de) timpul de integrare T_i .

Pentru a obține relațiile de implementare în LabVIEW calculăm discretizatului lui $H_c(s)$.

$$H_c^d(z) = (1 - z^{-1}) \mathcal{Z} \left[\frac{H_c(s)}{s} \right] = (1 - z^{-1}) \mathcal{Z} \left[k_c \left(\frac{1}{s} + \frac{1}{T_i s^2} \right) \right] = \quad (1.16)$$

$$= k_c \left(1 + \frac{h}{T_i} \frac{1}{z-1} \right) = \frac{c_1 z + c_2}{z-1},$$

unde

$$c_1 = k_c, \quad c_2 = k_c \left(\frac{h}{T_i} - 1 \right). \quad (1.17)$$

Relațiile de implementare ale algoritmului PI vor fi

$$\begin{cases} e(k) = r(k) - y(k) \\ u(k) = u(k-1) + c_1 e(k) + c_2 e(k-1) \end{cases} \quad (1.18)$$

Programul LabVIEW trebuie să prevadă posibilitatea de a modifica on-line parametrii de acordare k_c și T_i și după fiecare modificare a acestora trebuie imediat recalculați parametrii de calcul c_1 și c_2 .

Algoritmul PID. Există mai multe versiuni de legi de reglare PID. Prezentăm două dintre cele mai uzuale versiuni.

În prima versiune componenta derivativă acționează asupra erorii de reglare. Funcția de transfer a unui astfel de controller analogic PID este

$$H_c(s) = k_c \left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \alpha T_d s} \right), \quad (1.19)$$

cu parametrii de acordare: factorul de amplificare k_c , (constanta de) timpul de integrare T_i și (constanta de) timpul de derivare T_d . Constanta de timp "parazită" $T\gamma = \alpha T_d$ (fără ea componenta derivativă nu este posibilă) se consideră a fi o fracțiune din T_d (valorile uzuale pentru α sunt 0.1 : 0.2) în acest fel ținându-se sub control amplitudinea impulsurilor la variații bruște (treaptă) ale erorii.

Algoritmul PID în prima versiune se obține calculând discretizata lui $H_c(s)$:

$$\begin{aligned} H_c^d(z) &= (1 - z^{-1}) \mathcal{Z} \left[\frac{H_c(s)}{s} \right] = (1 - z^{-1}) \mathcal{Z} \left[k_c \left(\frac{1}{s} + \frac{1}{T_i s^2} + \frac{T_d}{1 + \alpha T_d s} \right) \right] = \\ &= k_c \left(1 + \frac{h}{T_i} \cdot \frac{1}{z-1} + \frac{1}{\alpha} \cdot \frac{z-1}{z - e^{-h/\alpha T_d}} \right) = \frac{c_1 z^2 + c_2 z + c_3}{z^2 - d_1 z - d_2}, \end{aligned} \quad (1.20)$$

unde, notând $\beta = e^{-h/\alpha T_d}$, avem

$$c_1 = k_c \left(1 + \frac{1}{\alpha} \right), \quad c_2 = k_c \left(\frac{h}{T_i} - \frac{2}{\alpha} - \beta - 1 \right), \quad c_3 = k_c \left(\beta - \beta \cdot \frac{h}{T_i} + \frac{1}{\alpha} \right)$$

și

$$d_1 = \beta + 1, \quad d_2 = -\beta.$$

În consecință, relațiile de implementare ale primei versiuni a algoritmului PID vor fi următoarele:

$$\begin{cases} e(k) = r(k) - y(k) \\ u(k) = d_1 * u(k-1) + d_2 * u(k-2) + c_1 * e(k) + c_2 * e(k-1) + c_3 * e(k-2) \end{cases} \quad (1.21)$$

În a doua versiune componenta derivativă acționează numai asupra ieșirii (nu și asupra referinței ca în prima versiune), vezi figura ??, i.e. controllerul se prezintă ca unul de tip PI în raport cu variațiile referinței și ca unul de tip PID în raport cu variațiile mărimii reglate. Această

a doua versiune asigură un compromis mai bun dintre performanțele la variațiile referinței și cele la variațiile perturbațiilor. În această a doua versiune avem,

$$U(s) = H_{c1}(s)(R(s) - H_{c2}(s)Y(s)) \quad (1.22)$$

unde $H_{c1}(s)$ este funcția de transfer PI de mai sus, iar

$$H_{c2}(s) = \frac{1 + T_d s}{1 + \alpha T_d s}, \quad (1.23)$$

cu aceiași parametrii de acordare ca la prima versiune.

Deși nu este riguros corect (de ce?) calculul discretizatei din cea de a doua versiune se poate face cu o bună aproximație (dacă h este suficient de mic) folosind discretizatul legii de reglare PI (vezi 1.16) și calculând suplimentar discretizatul lui $H_{c2}(s)$. Avem

$$\begin{aligned} H_{c2}^d(z) &= (1 - z^{-1})\mathcal{Z} \left[\frac{H_{c2}(s)}{s} \right] = (1 - z^{-1})\mathcal{Z} \left[\frac{1 + T_d s}{s(1 + \alpha T_d s)} \right] = \\ &= \left(1 + \frac{1 - \alpha}{\alpha} \frac{z}{z - e^{-h/\alpha T_d}} \right) = \frac{d_1 z + d_2}{z - f} \end{aligned} \quad (1.24)$$

unde

$$d_1 = \frac{1}{\alpha}, \quad d_2 = e^{-h/\alpha T_d}, \quad f = e^{-h/\alpha T_d}.$$

Relațiile de implementare ale algoritmului PID în cea de a doua versiune vor fi

$$\begin{cases} u_2(k) = f * u_2(k-1) + d_1 y(k) + d_2 y(k-1) & \%PD \\ e(k) = r(k) - u_2(k) \\ u(k) = u(k-1) + c_1 e(k) + c_2 e(k-1) & \%PI \end{cases} \quad (1.25)$$

Ca și în cazul algoritmului PI, programul LabVIEW pentru implementarea algoritmului PID trebuie să prevadă posibilitatea de a modifica on-line parametrii de acordare k_c , T_i și T_d (parametrul α nu se modifică în timpul operării) și după fiecare modificare a oricăruia dintre ei, trebuie imediat recalculați parametrii de relațiile de implementare. De asemenea se impun următoarele observații:

- algoritmul PID fiind de ordinul 2, trebuie inițializate două mărimi (care joacă rolul stărilor controllerului): $u(0)$ și $u(1)$ în prima versiune, respectiv $u(0)$ și $u_2(0)$ în versiunea a doua;
- în a doua versiune nu se calculează de fapt binecunoscuta eroare de reglare $e = r - y$; totuși, având în vedere faptul că $H_{c2}(0) = 1$ (respectiv $H_{c2}^d(1) = 1$) rezultă că în regim staționar $u_2 = y$, i.e. în regim staționar anularea diferenței $r - u_2$ are ca efect anularea erorii de reglare.

1.4 Sarcini de lucru

- Se va elabora un program Lab-View pentru simularea în timp real a unui proces de ordinul 1 (cu datele numerice din text) cu posibilitatea de conducere manuală de la tastatură sau mouse (i.e. de modificare a comenzii în timpul "funcționării" procesului) și de modificare (la momente aleatoare) a perturbației.
- Folosind programul de la punctul precedent se va face un antrenament de reglare manuală a procesului. Se vor evalua aproximativ performanțele obținute (suprareglaj și timp de răspuns la variații treaptă ale referinței, ecartul de variație al erorii de reglare la variații treaptă ale perturbației etc.).

2. Idem pentru un proces de ordinul 2 (cu datele numerice din text).
3. Dacă se dispune de timpul necesar se vor repeta cerințele de la punctul 1 pentru procesul de ordinul 1 cu timp mort (cu datele numerice din text).
4.
 - Se va elabora un program Lab-View pentru simularea în timp real a sistemului de reglare convențională cu controller PI și PID pentru fiecare dintre procesele de la punctele precedente.
 - Folosind acordările experimentale ale controllerului se vor compara performanțele de reglare obținute manual și automat.

Bibliografie

- [1] **I. Dumitrache**, *Ingineria Reglării Automate*, Curs.
- [2] **Jora B., Popeea C., Barbulea S.** *Metode de Calcul Numeric în Automatică*, Ed. Enciclopedică, București 1996.
- [3] **S. Călin, I. Dumitrache**, *Regulatoare Automate*, EDP, 1987.

Aplicația 1

Aplicația se va elabora în trei pași;

- a) Răspunsul indicial al procesului de ordinul întâi (Fig. 2);
- b) Reglarea manuală a procesului pentru atingerea unei referințe prestabilite (Fig. 3);
- c) Reglarea manuală a procesului în prezența perturbațiilor aleatoare (Fig. 4).

Fereastra-panou va conține elemente de control al parametrilor K, T și al comenzii (u). Pentru o manipulare ușoară a valorilor comenzii se poate introduce un control de tip Slide, la care se adaugă posibilitatea de vizualizare digitală.

Se vor introduce două grafice pentru vizualizarea ieșirii procesului și a comenzii.

Fereastra-diagramă (Fig.1) va conține modelul de ordinul I implementat într-o buclă While, având ca și condiție de stop apăsarea butonului de STOP din fereastra-panou.

Pentru ca simularea să se apropie cât mai mult de realitate se va introduce o temporizare care să se apropie de execuția în timp real. Se urmăresc următorii pași:

1. Se introduce un element de așteptare în bucla While (**Functions** → **Time&Dialog** → **Wait**);
2. În panou se setează formatul Time&Date pentru scala X prin selectarea opțiunii **X Scale** → **Formatting** → **Format: Time&Date** din meniul graficului;
3. Se introduce în diagramă câte un nod de proprietăți pentru fiecare din cele două grafice;
4. Se alege din lista de proprietăți **X Scale** → **Offset and Multiplier** → **Offset** și se alege operația de scriere pentru acest element (**Change to Write**);
5. Se introduce funcția care citește ora și data de la calculator (**Functions** → **Time&Dialog** → **Get Date/Time in Seconds**) și se conectează la nodul de proprietăți

Diagrama de mai jos a fost construită treptat urmărind cele trei obiective, forma atașată fiind cea mai completă (punctul c). Perturbațiile sunt de tip treaptă unitară și apar la momente aleatoare de timp. O modalitate de implementare a lor este cuprinsă în partea de jos a diagramei.

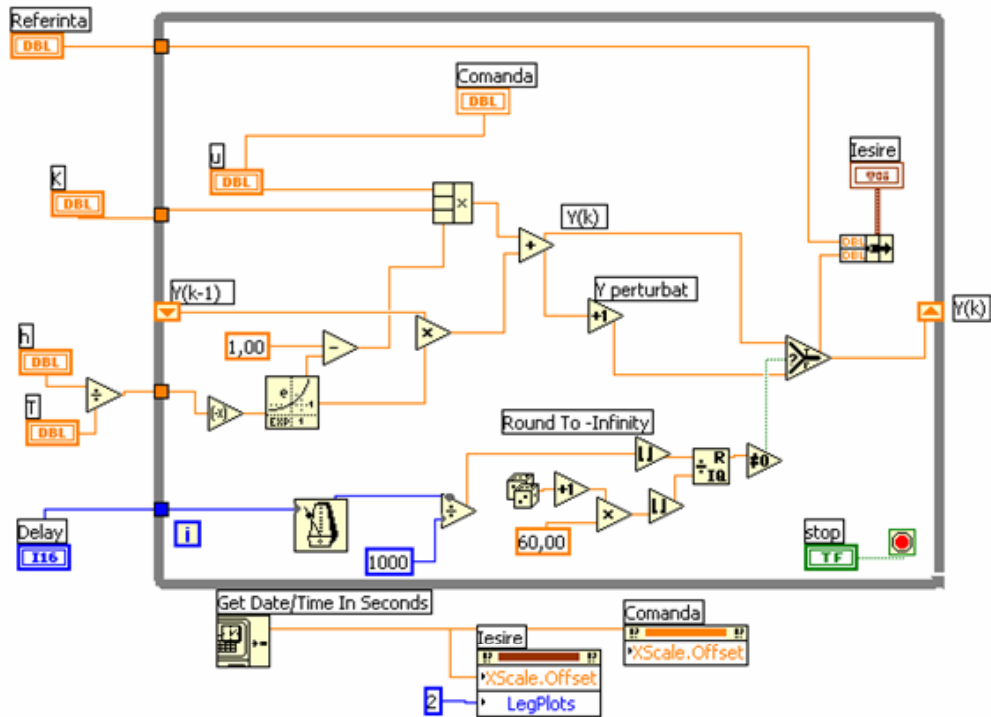


Fig. 1. Aplicația 1: Diagrama

Rezultate :

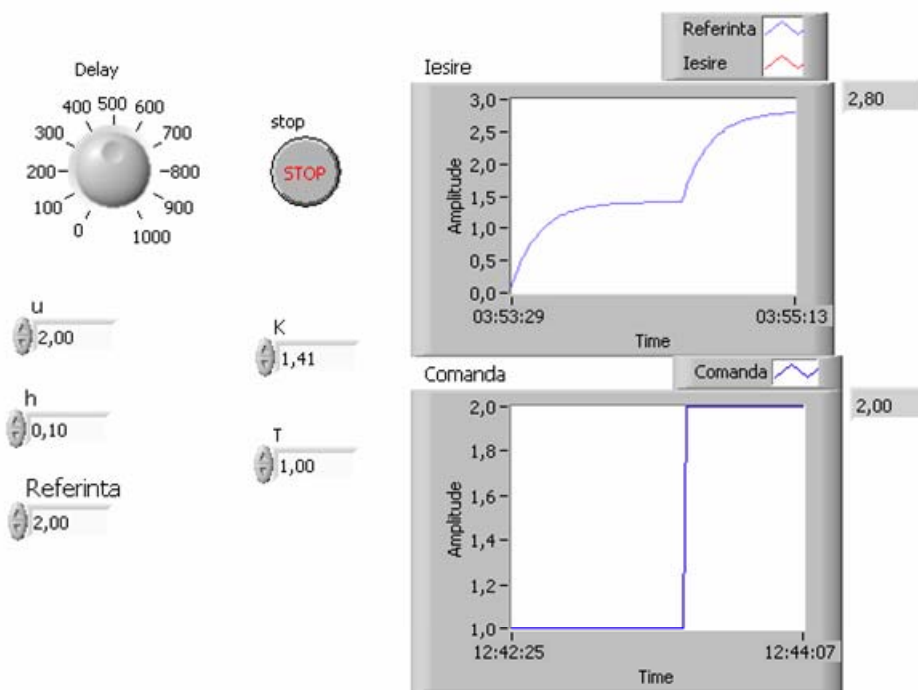


Fig. 2. Aplicația 1: Răspunsul indicial al procesului de ordinul 1

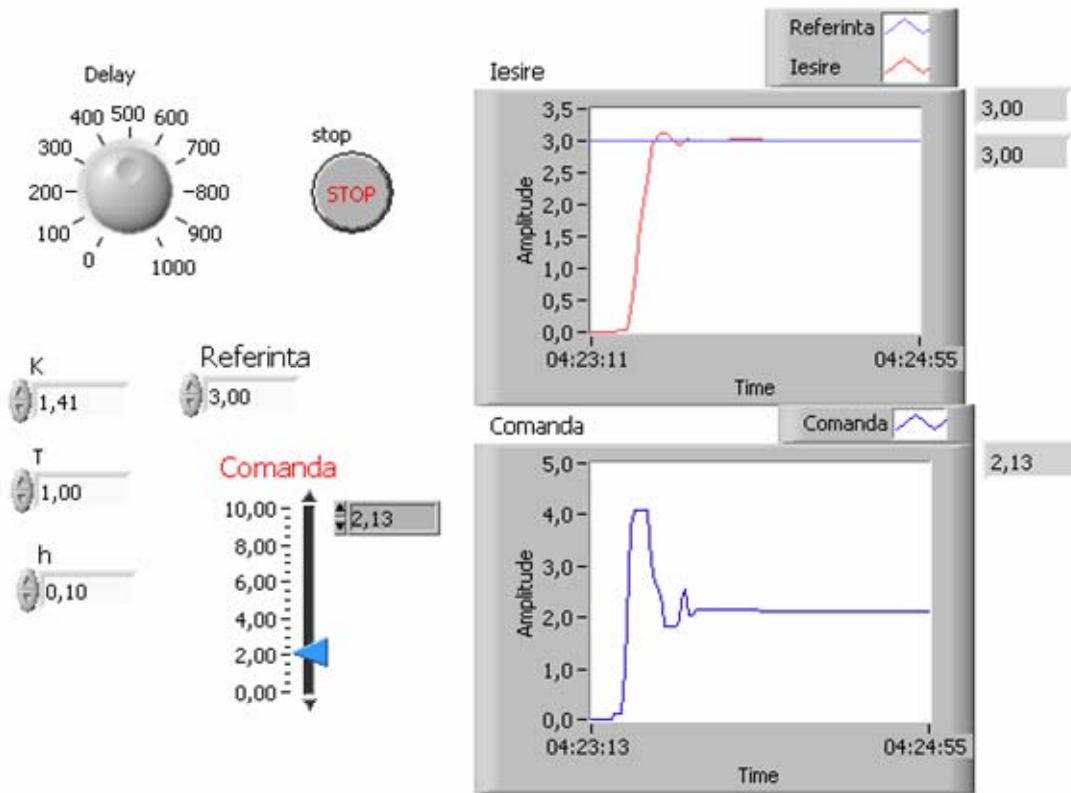


Fig. 3. Aplicația 1: Reglarea manuală a procesului.

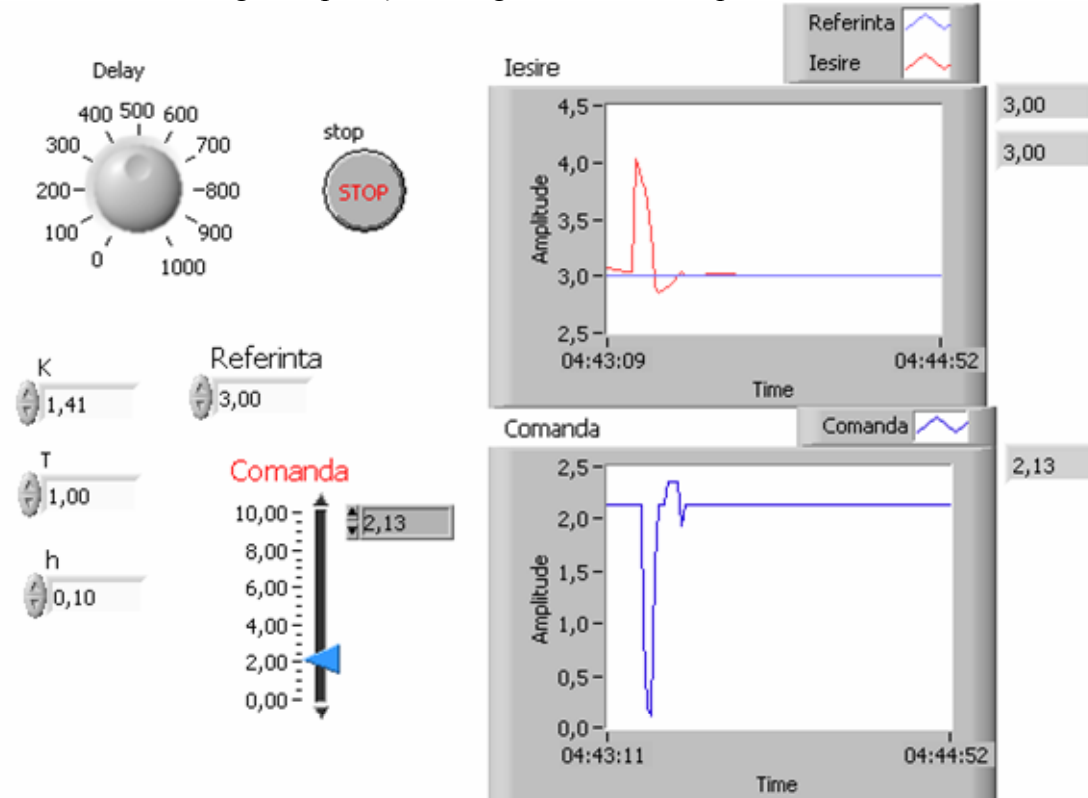


Fig. 4. Aplicația 1: Reglare manuală în prezența perturbațiilor.

Aplicația 2

Se îndeplinesc aceleași cerințe ca la aplicația anterioară, de data aceasta pentru un proces de ordinul II.

$$y(k) = b_1 u(k-1) + b_0 u(k-2) - a_1 y(k-1) - a_0 y(k-2),$$

unde:

$$b_1 = K \left(1 - e^{-\frac{h}{T}} - \frac{h}{T} \cdot e^{-\frac{h}{T}} \right)$$

$$b_0 = K \left(e^{-2\frac{h}{T}} - e^{-\frac{h}{T}} + \frac{h}{T} \cdot e^{-\frac{h}{T}} \right)$$

$$a_1 = -2 \cdot e^{-\frac{h}{T}}$$

$$a_0 = -e^{-2\frac{h}{T}}$$

Coeficienții a_1 , a_0 , b_1 , b_0 vor fi calculații în aplicații independente, care vor fi apelate în diagrama principală. În acest mod diagrama va fi mult mai puțin încărcată și mai ușor de urmărit.

SubVi-uri (Fig. 5)

Pentru a transforma o aplicație în SubVI se urmăresc următorii pași:

1. Se realizează diagrama aplicației în mod normal și se salvează;
2. În fereastra – panou se execută click-dreapta pe icoana din partea din dreapta-sus a ferestrei și se selectează **Show Connector**. Icoana se va transforma și vor apărea atâtea terminale câte elemente există pe panou;
3. Se asociază variabilele de intrare ale subaplicației prin executare click pe elementul de tip control și apoi pe un terminal liber; (nu toate elementele trebuie conectate, ci numai cele care prezintă interes);
4. Se asociază variabila de ieșire în același mod, dar de data aceasta se va selecta un element de tip indicator.
5. Se salvează aplicația.

Pentru a introduce un SubVI într-o altă aplicație se selectează **Functions** → **Select a VI** și se alege aplicația necesară.

În cazul acestei aplicații se vor crea patru subaplicații corespunzătoare celor patru parametri, având ca elemente de intrare h (pasul de discretizare), T (constanta de timp) și, K , iar ca element de ieșire parametrul de calculat.

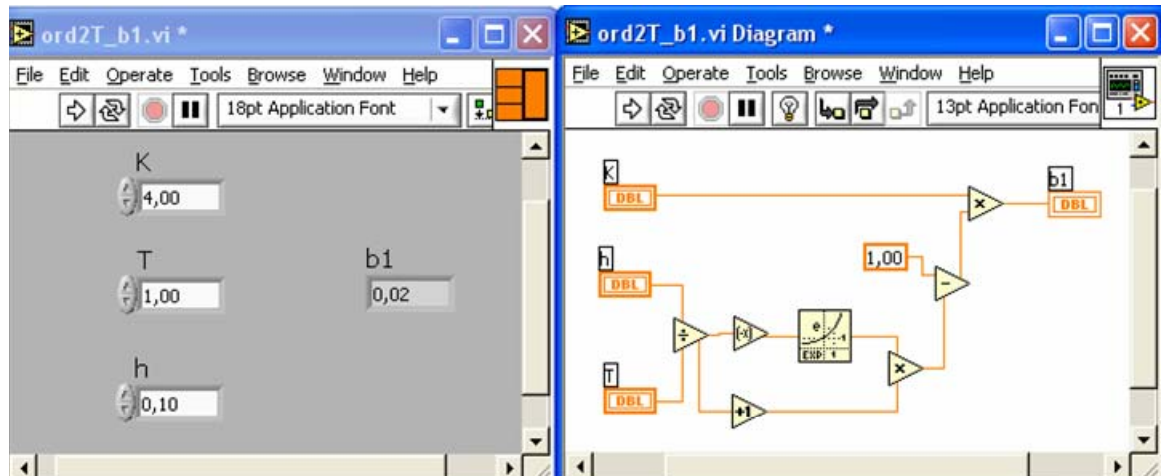


Fig. 5. Aplicația 2: Exemplu de SubVI cu calculul coeficienților.

Rezultate experimentale:

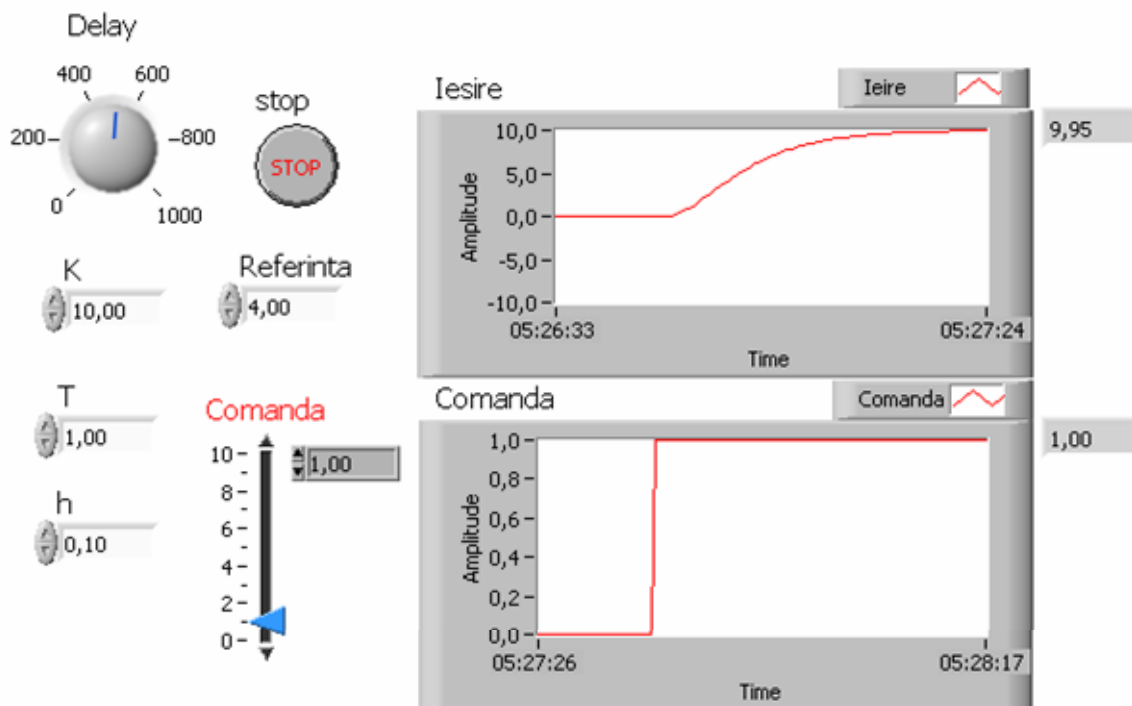


Fig. 6. Aplicația 2: Răspunsul indicial al unui proces de ordinul II.

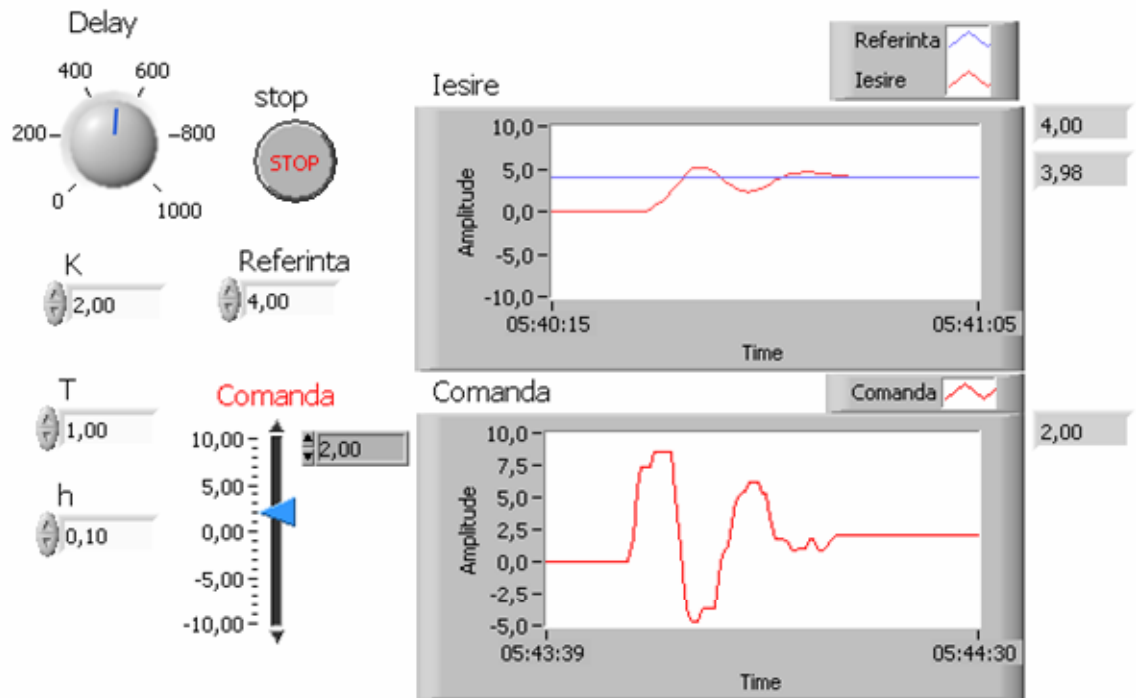


Fig. 7. Aplicația 2: Reglarea manuală a procesului de ordinul 2.

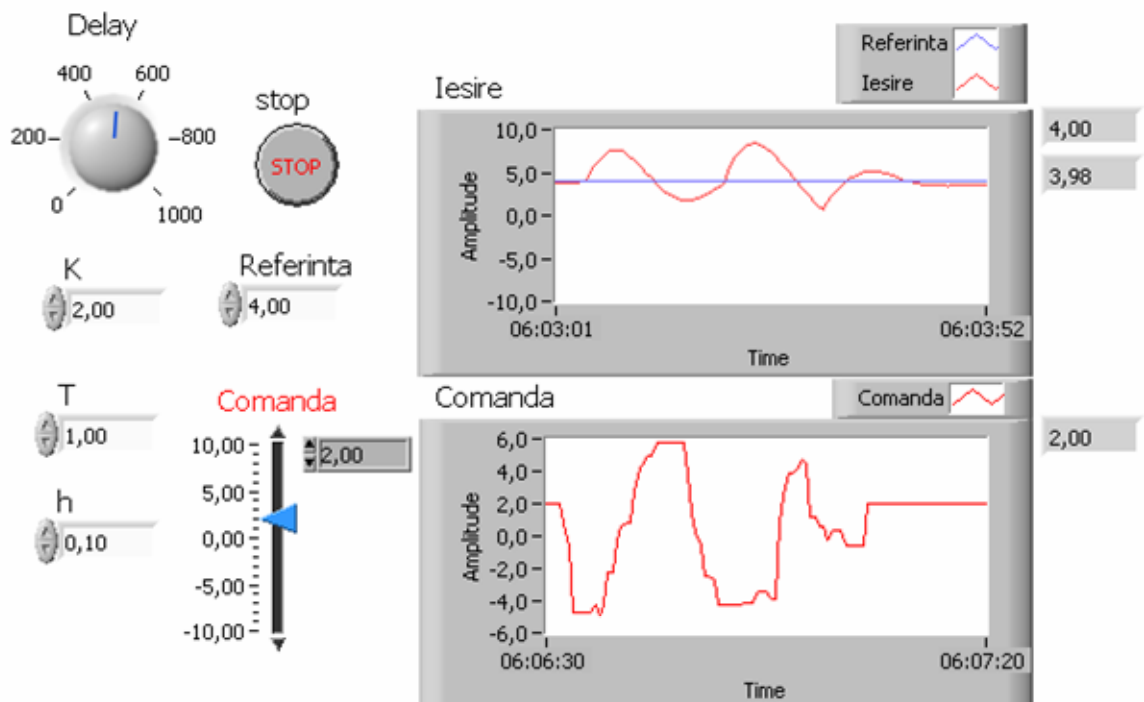


Fig. 8. Aplicația 2: Reglarea manuală în prezența perturbațiilor.

Aplicația 3

Se va implementa un regulator de tip PI. După discretizare modelul regulatorului va fi de forma:

$$u(k+1) = u(k) + K_R e(k+1) + K_R \left(\frac{h}{T_i} - 1 \right) e(k),$$

unde u este comanda dată de regulator, iar e este eroarea, mărime de intrare în regulator.

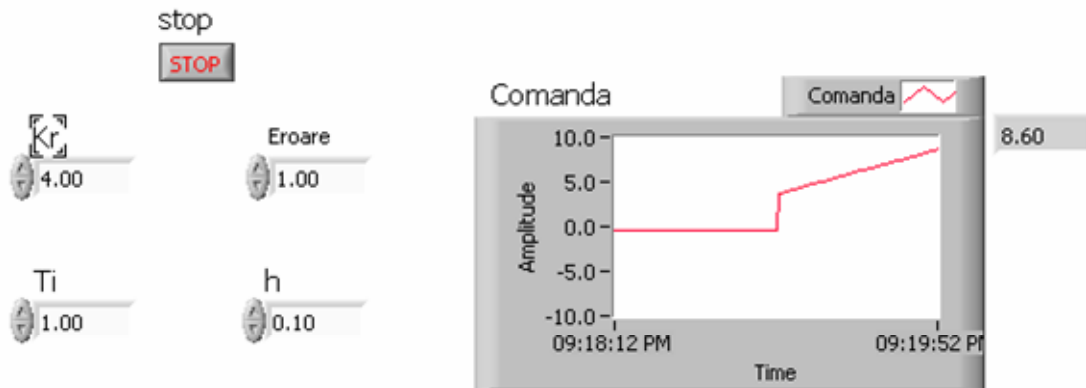


Fig. 9. Aplicația 3: Regulator PI

Aplicația 4.

Se va crea o buclă de reglare convențională cu controller de tip PI și proces de ordinul I. În acest scop vor fi utilizate aplicațiile create în aplicațiile 1 și 3. Acestea vor fi transformate în subaplicații și apelate în diagrama principală.

SubVI-ul Proces (Fig. 10) va avea ca mărimi de intrare:

- comanda de la regulator (u_k);
- ieșirea procesului la pasul anterior (y_{k-1}),

și ca mărime de ieșire y_k , ieșirea procesului la pasul curent.

1. Se elimină din diagrama bucla While, elementele de temporizare și nodurile de proprietăți ale graficelor;
2. Se elimină din fereastra – panou graficul comenzii;
3. Se introduce în fereastra panou un element de control având semnificația ieșirii procesului la pasul anterior (y_{k-1}). Acest element va înlocui registrul de șiftare al buclei While;
4. Se înlocuiește graficul ieșirii procesului cu un indicator de tip numeric;
5. Se execută click-dreapta pe icoana din partea dreapta-sus a ecranului pentru afișarea terminalelor aplicației (**Show Connector**);
6. Se conectează mărimile de intrare și cea de ieșire; celelalte terminale pot fi eliminate;

7. Parametrii K , T și h vor primi valorile stabilite în text; acești parametri nu vor mai fi controlați din diagrama principală
8. Se salvează aceste valori ca fiind implicite la rularea aplicației prin selectarea **Operate** → **Make Current Values Default** din bara de meniu din partea de sus a ferestrei;
9. Se salvează aplicația.

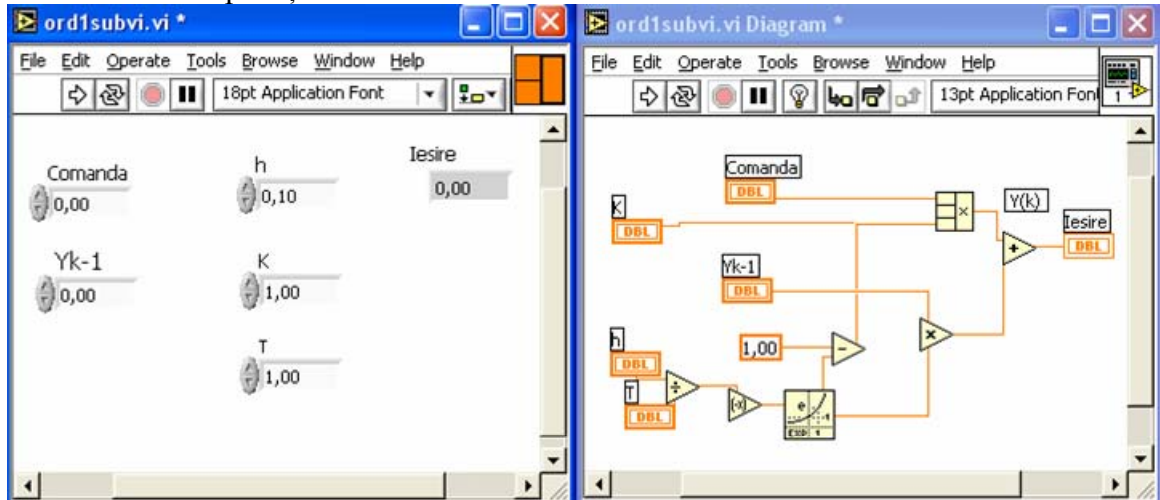


Fig. 10. Aplicația 4. SubVI Proces.

SubVI-ul PI se va realiza similar cu cel al procesului și va avea ca mărimi de intrare:

- parametri regulatorului (K_r , T_i);
- eroarea staționară la pasul curent (e_k);
- eroarea staționară la pasul anterior (e_{k-1});
- comanda la pasul anterior (U_{k-1}),

iar ca mărime de ieșire comanda la pasul curent (U_k).

Fereastra-panou (Fig. 11) a aplicației principale va conține elemente de control al parametrilor regulatorului și al referinței și grafice pentru urmărirea evoluției ieșirii procesului și a comenzii.

Fereastra-diagramă va conține o buclă While având ca și condiție de stop apăsarea butonului STOP din panou. Se vor include cele două subaplicații, la care vor conecta parametrii necesari.

Aplicația are ca și obiective urmărirea referinței, și rejecția perturbațiilor de tip treaptă, care vor fi implementate ca și la aplicațiile precedente.

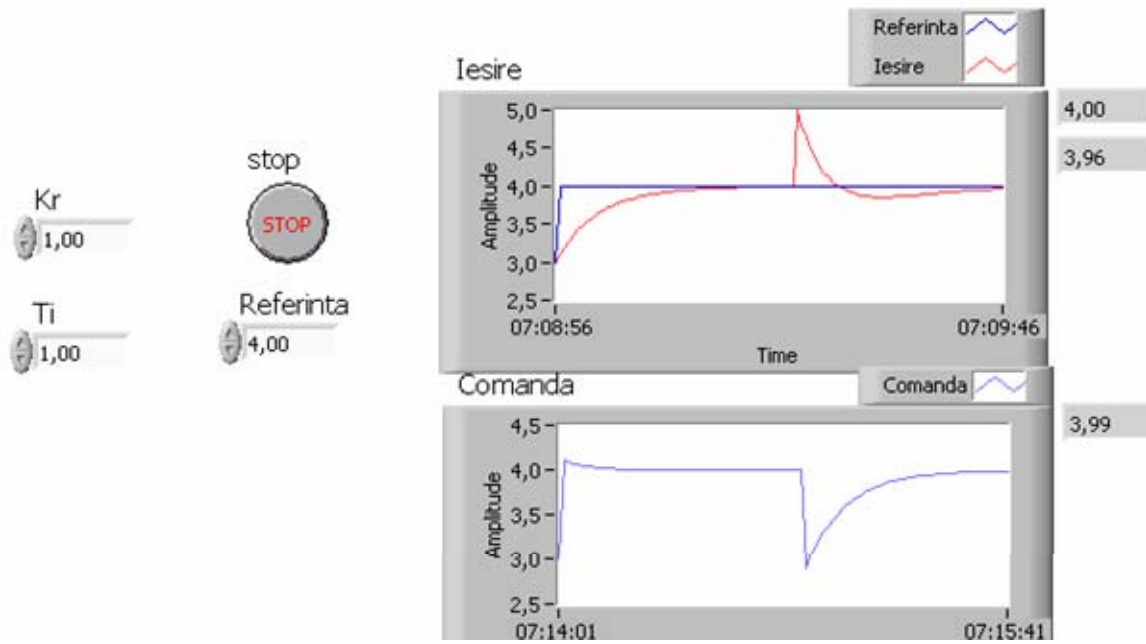


Fig. 11. Aplicația 4. Regalare convențională. Urmărirea referinței și rejecția perturbațiilor.

Tema:

Să se simuleze un regulator de tip PID și să se construiască o buclă de reglare cu PID-ul simulat și cu procesul de ordinul 1.