

Proiect IV

Bota Andrei

1. Introducere și Obiective

1.1. Contextul Proiectului

În ingineria sistemelor de conducere, controlerul PID (Proportional-Integral-Derivativ) reprezintă soluția standard pentru reglarea proceselor industriale datorită eficienței și flexibilității sale. Această lucrare prezintă proiectarea și implementarea unui instrument virtual (VI) în mediul LabVIEW pentru simularea și analiza unui sistem de reglare automată în buclă închisă.

1.2. Scopul Lucrării

Scopul principal al acestui proiect este realizarea unui sistem capabil să mențină o variabilă de proces (PV) la o valoare dorită, numită Setpoint (SP), prin ajustarea continuă a comenzii furnizate de algoritmul PID către o plantă simulată. Aplicația pune accent pe analiza în timp real a performanțelor, oferind utilizatorului date cantitative despre calitatea reglării.

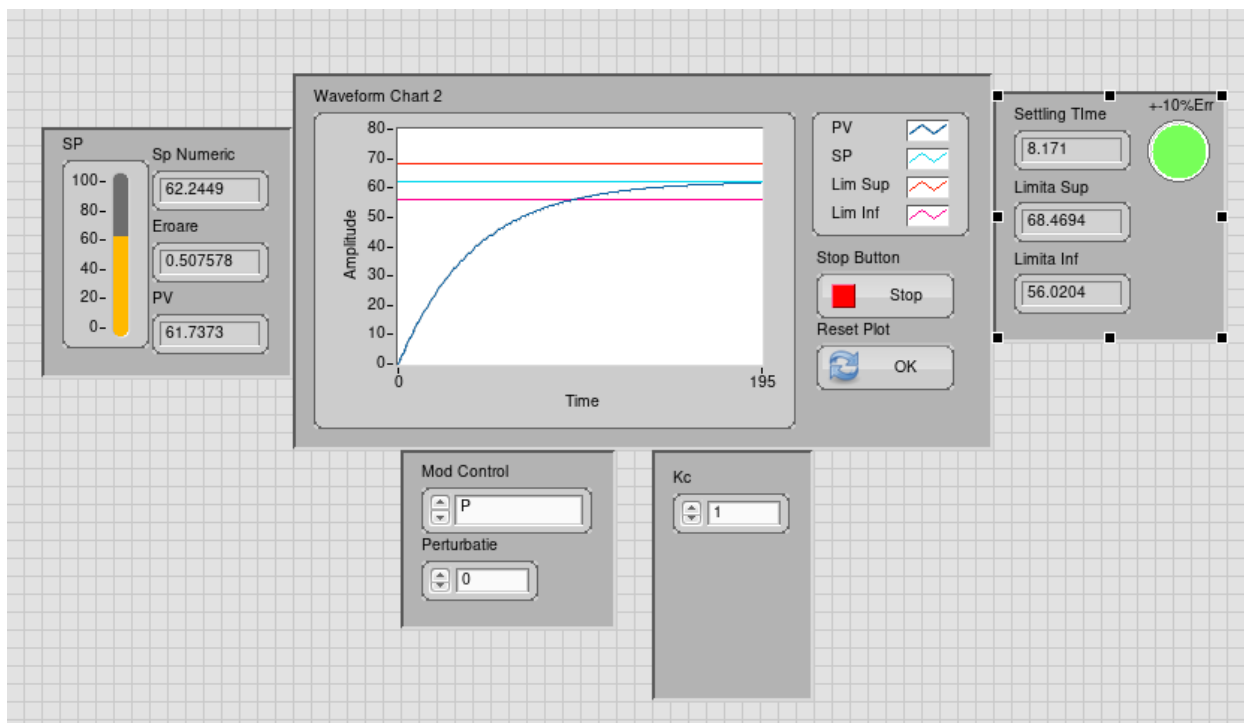


Figura 1. Front Panel

2. Descrierea Interfeței (Front Panel)

Interfața grafică a fost concepută pentru a fi intuitivă, grupând elementele în funcție de rolul lor în procesul de reglare. Aceasta permite monitorizarea vizuală a evoluției sistemului și intervenția rapidă asupra parametrilor de control.

2.1. Zona de Comandă și Configurare (Stânga și Jos)

Această zonă permite utilizatorului să interacționeze direct cu simularea:

- **Slider Setpoint (SP):** Elementul principal de control prin care se stabilește valoarea dorită a variabilei de proces. Mișcarea acestuia declanșează automat cronometrarea timpului de stabilizare.
- **Mod Control (Enum):** Permite selectarea tipului de algoritm utilizat (P, PI sau PID), oferind flexibilitate în demonstrarea rolului fiecărei componente (Proportională, Integrală, Derivativă).
- **Parametri PID (Kc, Ti, Kd):** Câmpuri numerice pentru acordarea fină a controlerului, permițând observarea directă a modului în care acești coeficienți influențează stabilitatea și viteza de răspuns.
- **Perturbație:** Un control numeric utilizat pentru a introduce șocuri externe în sistem, testând capacitatea PID-ului de a menține referința în condiții adverse.

2.2. Zona de Vizualizare (Central)

Elementul central este Waveform Chart 2, care oferă o perspectivă istorică asupra comportamentului sistemului. Pentru o analiză riguroasă, acesta afișează simultan patru fire de date (trasee):

1. PV (Linia Albastră): Reprezintă variabila de proces curentă; urmărește modul în care sistemul reacționează la comenzi.
2. SP (Linia Roșie): Reprezintă referința (ținta) pe care PV trebuie să o atingă.
3. Limita Sup (Linia Cyan): Marchează pragul de $+10\%$ din valoarea Setpoint-ului.
4. Limita Inf (Linia Magenta): Marchează pragul de -10% din valoarea Setpoint-ului.

Acest „tunel” format de limitele superioară și inferioară permite validarea vizuală instantanee a încadrării sistemului în standardele de performanță impuse.

2.3. Indicatorii de Performanță și Stare (Dreapta)

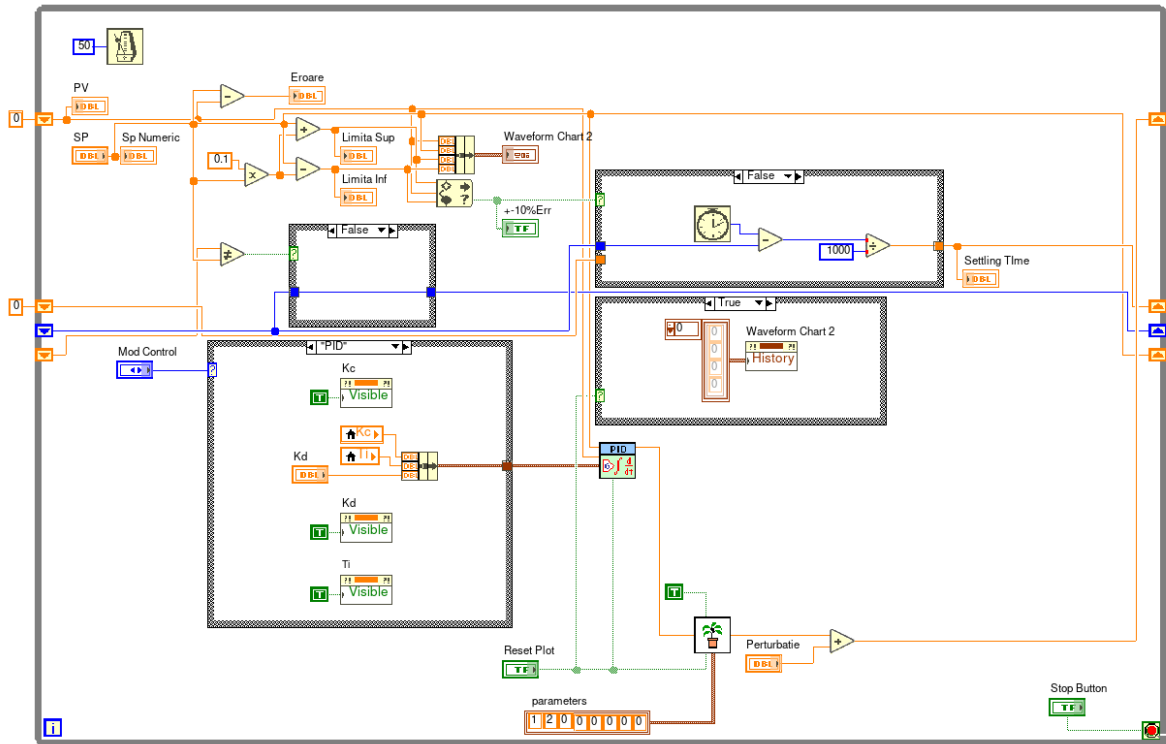
Sistemul calculează și afișează automat eficiența reglării prin:

- Settling Time (s): Un cronometru de precizie care măsoară secunde scurse de la ultima schimbare a Setpoint-ului până în momentul în care PV intră în banda de 10% . Valoarea este „înghețată” pe ecran imediat ce condiția de stabilitate este îndeplinită.
- LED-ul de Stare ($\pm 10\% \text{Err}$): Un indicator vizual de tip Boolean (verde) care se aprinde doar atunci când variabila de proces se află în interiorul benzii de toleranță. Acesta confirmă utilizatorului că sistemul este într-o stare stabilă.
- Buton Reset Plot: O funcție esențială care golește graficul și reinițializează memoria algoritmilor, permițând efectuarea unor teste curate, fără influența datelor anterioare.

3. Implementarea Tehnică (Block Diagram)

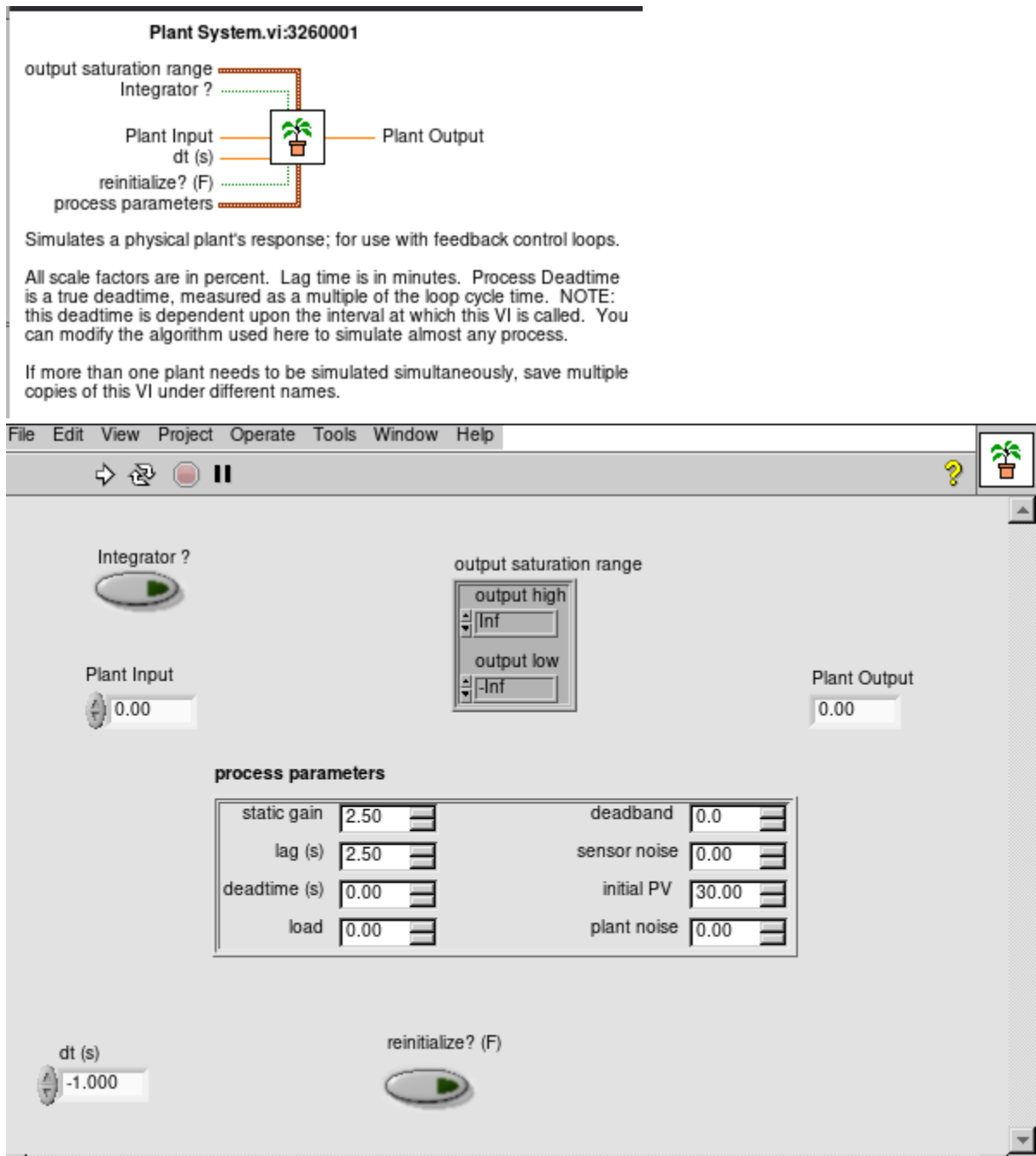
- Secțiunea Block Diagram reprezintă nucleul logic al aplicației, unde fluxul de date este gestionat printr-o arhitectură de tip *While Loop* care rulează continuu pentru a asigura

simularea în timp real a procesului de reglare.

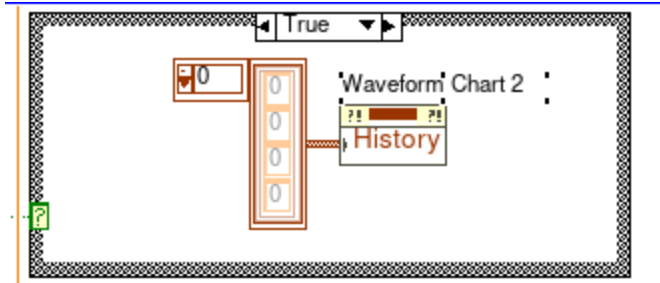


• 3.1. Structura Principală și Controlul PID

- Logica de control este construită în jurul funcției PID.vi, care primește ca intrări valoarea dorită (Setpoint) și valoarea curentă (PV).
- Calculul Comenzii: Pe baza parametrilor Kc, Ti și Kd, algoritmul calculează semnalul de comandă trimis către „Planta” simulată (pictograma cu floarea), care modelează comportamentul fizic al sistemului.

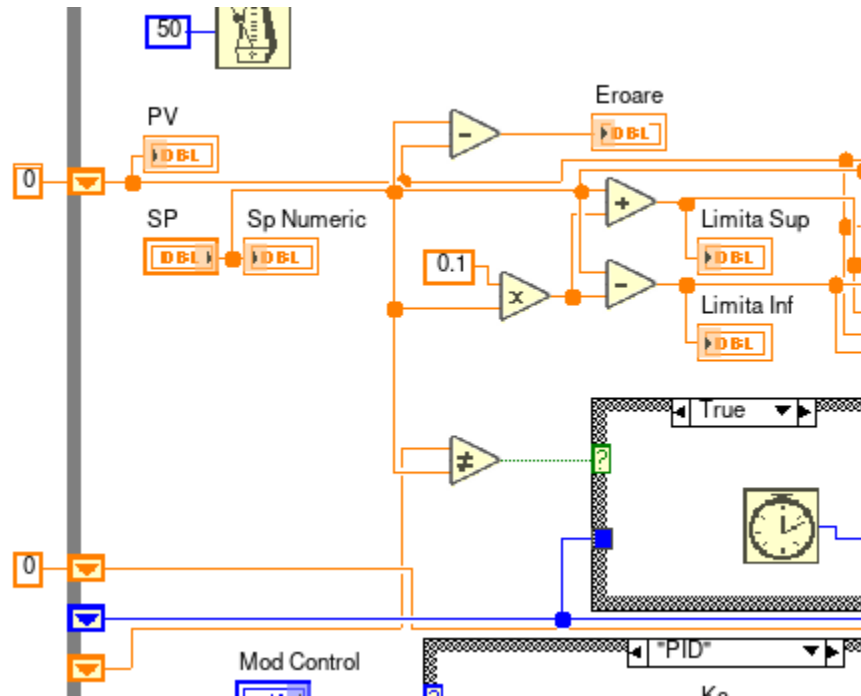


- Bucla de Feedback: Valoarea de ieșire a plantei este retransmisă la intrarea PID-ului prin intermediul unui Shift Register, închizând astfel bucla de reglare.
- 3.2. Logica de Resetare Sincronizată
- O componentă critică a implementării este sistemul de resetare globală, activat de butonul Reset Plot. Acesta execută simultan patru acțiuni esențiale:
- History Data: Golește istoricul graficului *Waveform Chart 2* folosind un *Property Node* alimentat cu un tablou (array) gol.



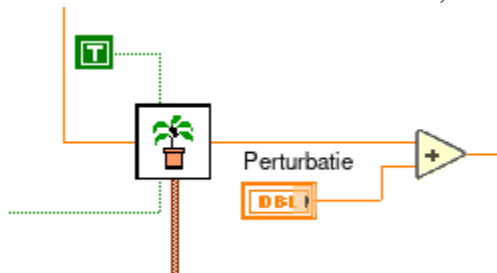
- Reinitialize PID: Trimite un semnal pe intrarea reinitialize? a PID-ului pentru a șterge eroarea acumulată în componenta integrală.
 - Reset Plant: Aduce starea internă a simulării la valorile inițiale.
 - Value Reset: Folosește o funcție Select pentru a forța valoarea din Shift Register la 0, asigurând o repornire curată a sistemului.
- ### 3.3. Calculul Limitelor și al Settling Time
- Pentru evaluarea automată a performanței, au fost implementate următoarele blocuri logice:
 - Determinarea Limitelor: Se calculează dinamic pragurile SP $\pm 10\%$ folosind funcții matematice de bază. Aceste valori sunt grupate într-un Bundle alături de SP și PV pentru a fi afișate simultan pe grafic.
 - Cronometrarea Settling Time: * Folosind un Shift Register și funcția Not Equal?, sistemul detectează orice modificare a Setpoint-ului și salvează momentul de start prin funcția Tick Count (ms).
 - O structură de tip Case comandată de LED-ul $\pm 10\% \text{Err}$ gestionează afișarea: cât timp sistemul este instabil, timpul curge; imediat ce PV intră în banda de $\pm 10\%$, cronometrul „îngheață” ultima valoare prin retransmiterea acesteia în Shift

Register fără modificări.



• 3.4. Injectarea Perturbației

- Pentru testarea robusteții, între ieșirea plantei și feedback-ul către sistem a fost inserată o funcție de adunare (Add). Aceasta permite utilizatorului să adauge manual un offset variabilei de proces, simulând un factor perturbator extern. Această implementare demonstrează capacitatea controlerului PID de a compensa erorile neprevăzute și de a stabiliza sistemul conform referinței.

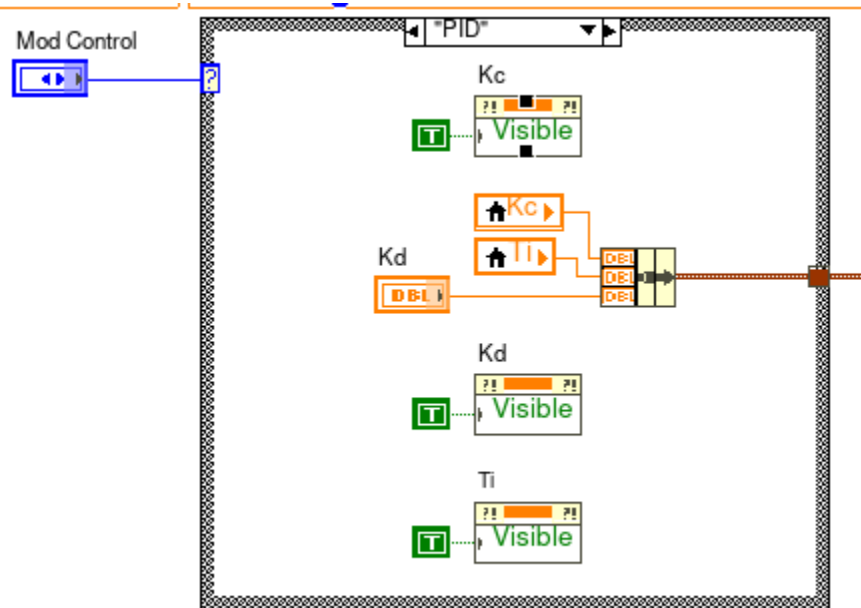


3.5. Gestionarea Dinamică a Interfeței (Vizibilitatea Parametrilor)

Pentru a menține interfața curată și pentru a evita confuziile utilizatorului, am implementat o logică de afișare dinamică a controalelor pentru parametrii PID (K_c , T_i , K_d).

- Implementare prin Case Structure: Am utilizat o structură de tip Case, controlată direct de selectorul "Mod Control" (Enum).
- Utilizarea Property Nodes: În interiorul fiecărui caz (P, PI, PID), am inserat Property Nodes setate pe atributul Visible pentru fiecare control numeric în parte.
- Logica Funcțională:
 - Dacă se selectează modul P, doar controlul pentru K_c devine vizibil, restul fiind ascunse.
 - În modul PI, sunt afișate controalele pentru K_c , T_i .
 - În modul PID, toate cele trei controale (K_c , T_i , K_d) sunt făcute vizibile pentru a permite acordarea completă a sistemului.

Avantaj: Această abordare previne introducerea de valori pentru parametri care nu sunt utilizați de algoritmul curent, reducând riscul de erori de operare și oferind un aspect profesional aplicației.



5. Rezultate Experimentale și Analiză Comparativă

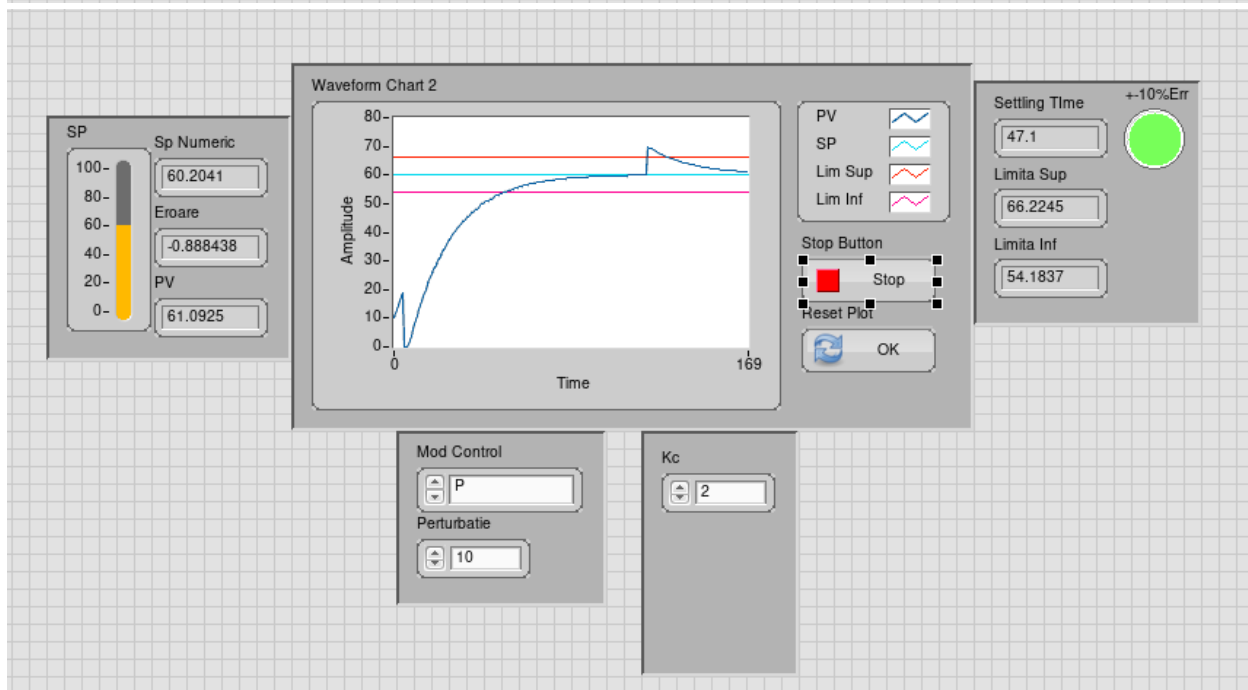
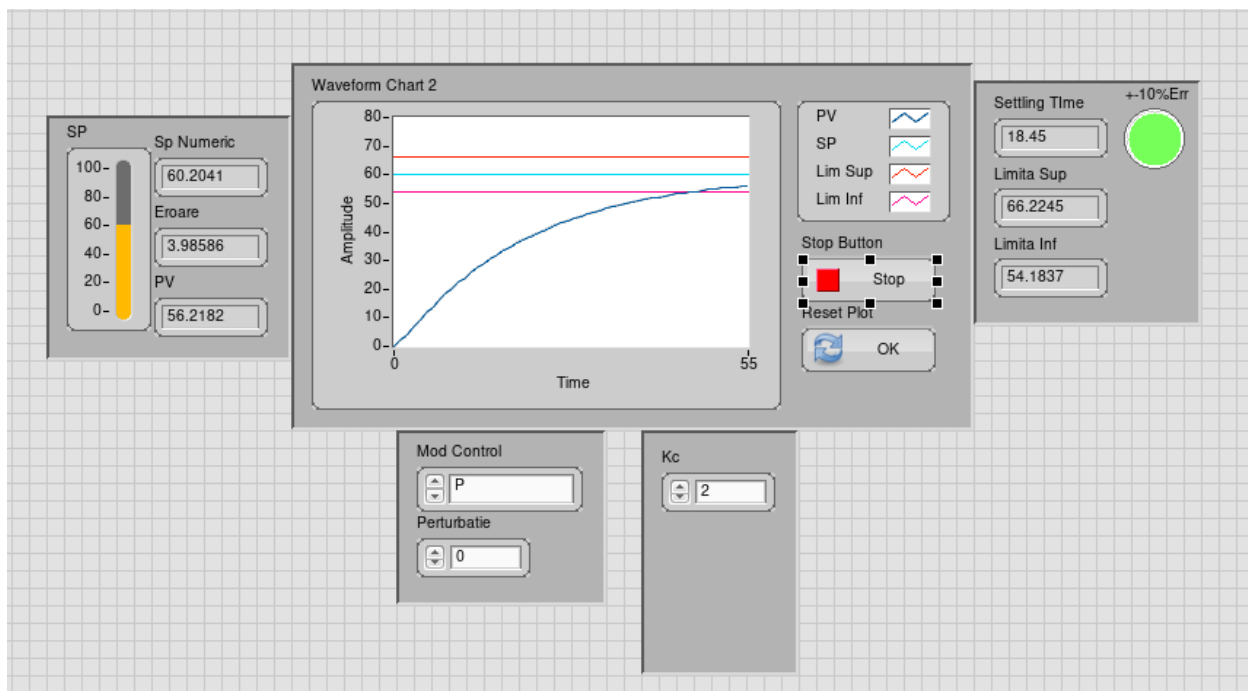
În această secțiune sunt prezentate rezultatele simulărilor efectuate pentru a valida funcționalitatea sistemului și pentru a evidenția impactul fiecărei componente a algoritmului PID asupra performanței procesului de reglare.

Testele au fost efectuate pornind de la o stare inițială de echilibru ($PV=0$), aplicând o variație treaptă a Setpoint-ului la valoarea **SP = 60**.

5.1. Scenariul 1: Controlul Proporțional (Modul P)

Configurație test:

- Mod Control: **P**
- Parametri: $K_c = 2$ (Doar amplificare)

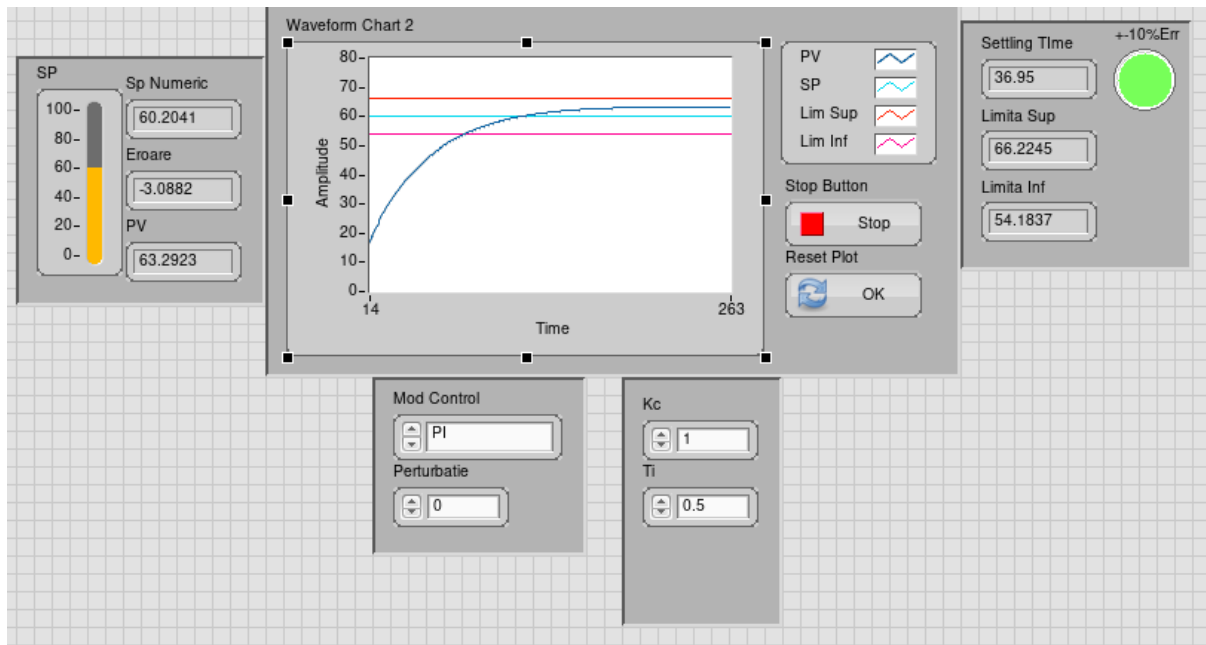


Perturbatie adaugata de 10

5.2. Scenariul 2: Controlul Proportional-Integral (Modul PI)

Configurație test:

- Mod Control: **PI**
- Parametri: $K_c = 2$, $T_i = 0.5$ (Se adaugă acțiunea integrală)



5.2. Scenariul 3: Controlul Proportional-Integral Derivativ (Modul PID)

Configurație test:

- Mod Control: **PID**
- Parametri: $K_c = 2$, $T_i = 0.5$, $K_d = 0.01$ (Se adaugă acțiunea integrală)

