**Proiectare regulator PID pentru instalația Macara 3D**

***Fișă de lucru***

|  |  |
| --- | --- |
| **Nume Prenume** |  |
| **Grupa** |  |

**Mod de lucru:** În cele ce urmează, se va efectua proiectarea a trei regulatoare PID (pentru fiecare axa, X, Y și Z) pentru instalația de laborator Macara 3D, ce poate fi aproximată cu un sistem de ordinul I cu integrator.

Pentru acest tip de procese, se poate utiliza ca metodă de proiectare a regulatorului **Criteriul simetriei**, sau **Metoda poli-zerouri** - aplicată pentru procesul căruia i s-a neglijat integratorul (având în vedere că procesul conține deja integrator).

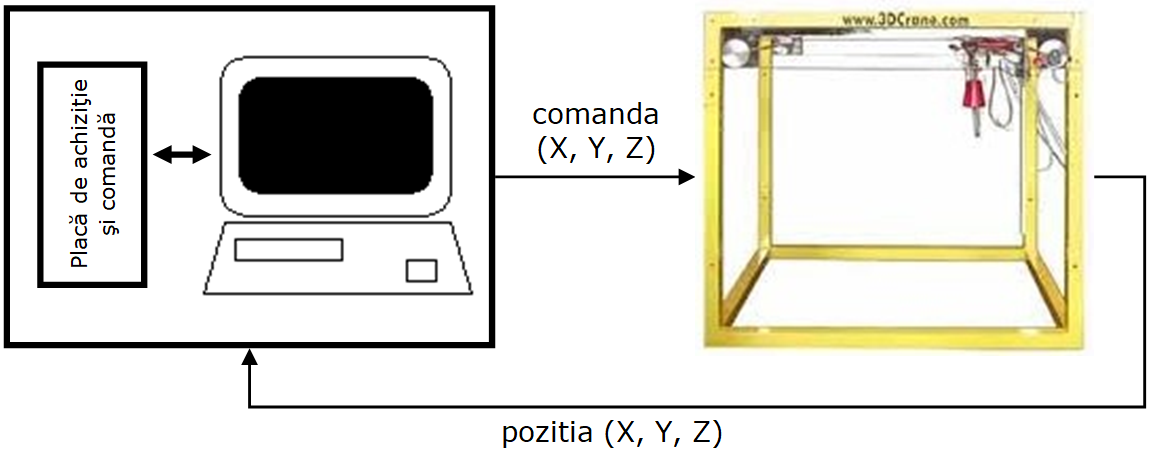
|  |
| --- |
| **Criteriul simetriei**  Pentru procesele rapide cu constantă parazită () și poli în origine de forma:  regulatorul se calculează cu formula:  ce asigură |

Pentru cele 3 axe de mișcare (X, Y și Z), se consideră un comportament descris de următoarele 3 funcții de transfer:

**Observație:** Se vor urmări cu **atenție** toți pașii prezentați în fișă, atât cei necesari pornirii și opririi instalației, precum și cei specifici lucrării didactice.

Instalaţia **Macara3D** este o platformă didactică destinată studiului unui proces tehnologic, în care se urmăreşte poziționarea in 3 dimensiuni a unui obiect prin intermediul unui fir flexibil.

Instalația are următoarea structură:



**Pași de lucru:**

***1. Proiectare analitică – criteriul simetriei***

1.1 Să se proiecteze setul de regulatoare utilizând Criteriul simetriei. Calculele se includ în spațiul următor:

|  |
| --- |
|  |

1.2 Se notează forma legilor de reglare PI obținute:

|  |  |  |
| --- | --- | --- |
| **HRx(s)** | **HRy(s)** | **HRz(s)** |
|  |  |  |

1.3 Se notează valorile parametrilor de acord pentru aceste regulatoare PI, marcându-se corespunzător unitățile de măsură:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **HRx(s)** | **HRy(s)** | **HRz(s)** |
| **KR** |  |  |  |
| **Ti** |  |  |  |

***2. Proiectare analitică – metoda poli-zerouri***

2.1 Să se proiecteze cele 3 regulatoare astfel încât răspunsul sistemului în buclă închisă să fie caracterizat de: , , .

**Observație:** Pentru proiectare, se va utiliza Metoda poli–zerouri aplicată pentru procesul căruia i s-a neglijat integratorul. Astfel, se obține un regulator PI din care se păstrează doar componenta proporțională, deoarece Integratorul se află de fapt în proces.

Calculele se includ în spațiul următor:

|  |
| --- |
|  |

2.2 Se notează forma legilor de reglare PI obținute:

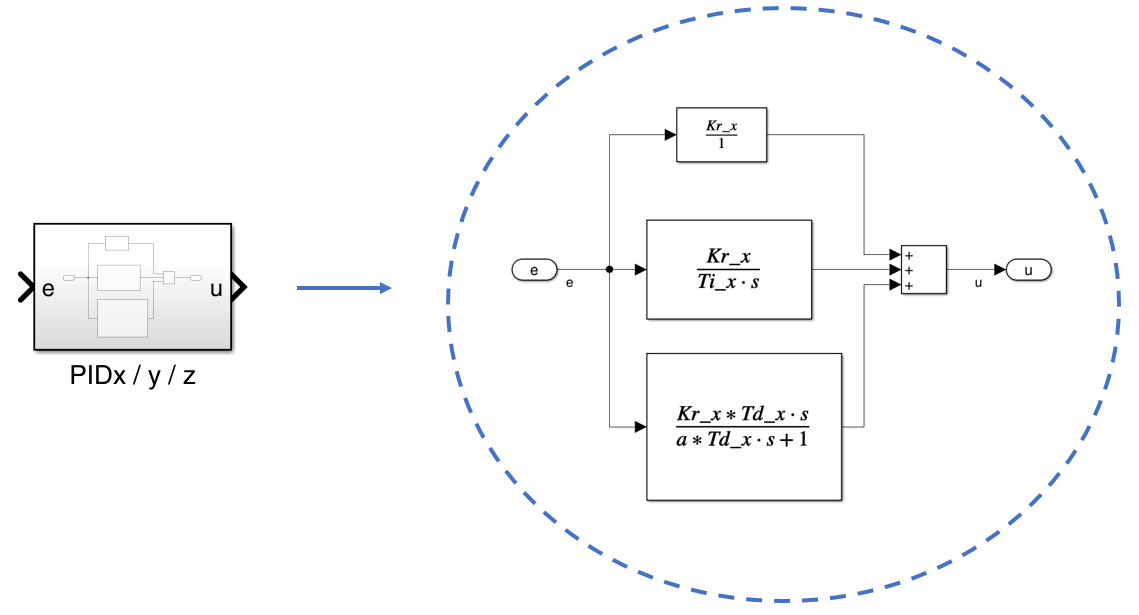
|  |  |  |
| --- | --- | --- |
| **HRx(s)** | **HRy(s)** | **HRz(s)** |
|  |  |  |

2.3 Se notează valorile parametrilor de acord pentru aceste regulatoare PI, marcându-se corespunzător unitățile de măsură:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **HRx(s)** | **HRy(s)** | **HRz(s)** |
| **KR** |  |  |  |
| **Ti** |  |  |  |

***3. Validare regulatoare în simulare***

3.1 Se implementează în Simulink structura de reglare cu un grad de libertate pentru fiecare axă în parte (X, Y și Z – 3 structuri în același fișier Simulink), unde procesul se implementează cu un bloc “Transfer fcn”, iar regulatorul se implementează ca un subsistem în forma PID paralel:



**Observație:** Parametri KR, Ti și Td ai regulatoarelor se vor introduce ca variabile în funcțiile de transfer (Transfer fcn), urmând să fie inițializați cu valori numerice în *Workspace*.

**Observație:** Se alege constanta de filtrare .

Schema Simulink implementată se include în spațiul următor:

|  |
| --- |
|  |

3.2 Pentru testarea regulatoarelor PI proiectate prin **Criteriul simetriei**, se implementează un script Matlab pentru configurarea parametrilor de acord, in care se setează:

* Kr = *valoarea obținută analitic*
* Ti = *valoarea obținută analitic*
* Td=0 (se dezactivează)

Conținutul script-ului se include în spațiul următor:

|  |
| --- |
|  |

3.3 Se rulează simularea iar răspunsurile celor 3 axe se includ în spațiul următor:

|  |
| --- |
|  |

3.4 Se estimează timpul tranzitoriu, suprareglajul și eroarea staționară și se introduc în tabelul următor (marcând corespunzător unitățile de măsură):

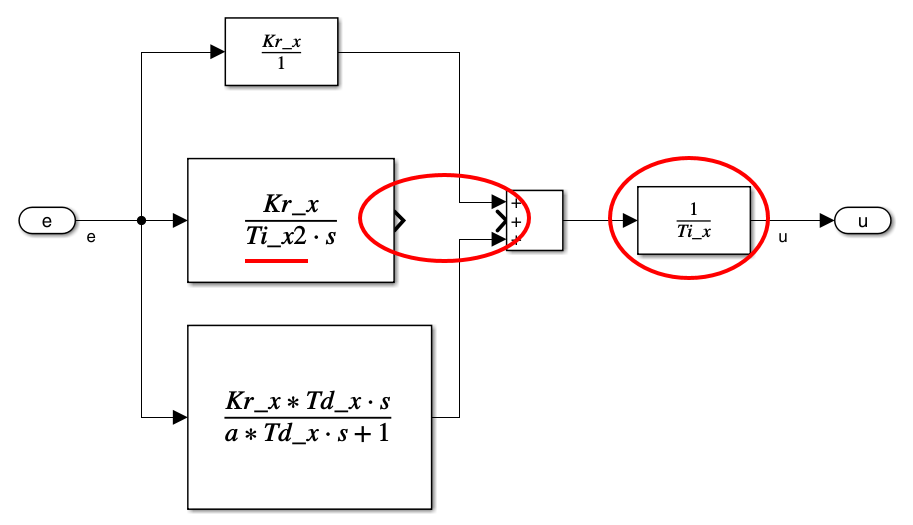
|  |  |  |  |
| --- | --- | --- | --- |
|  | **tt** | **σ** | **εst** |
| **X** |  |  |  |
| **Y** |  |  |  |
| **Z** |  |  |  |

3.5 Se interpretează rezultatele, observațiile se includ în spațiul următor:

|  |
| --- |
|  |

3.6 Pentru testarea regulatoarelor PI proiectate prin **Metoda poli-zerouri**, se va implementa un **nou fișier Simulink** similar (cel de la Criteriul simetriei se păstrează pentru mai târziu). Acesta va implementa practic doar un regulator P (componenta integrativă se află deja în proces), dar se adaugă un factor suplimentar de amplificare pentru a compensa lipsa factorului :

Regulatoarele din cele 3 subsisteme se modifică astfel:



se implementează un **nou script Matlab** pentru configurarea parametrilor de acord, in care se setează:

* Kr = *valoarea obținută analitic*
* Ti = *valoarea obținută analitic*
* Ti2 = *orice valoare* (s-a deconectat componenta integrativă si s-a redenumit Ti Ti2 pentru că va fi folosită mai târziu)
* Td=0 (se dezactivează)

Conținutul script-ului se include în spațiul următor:

|  |
| --- |
|  |

3.7 Se rulează simularea iar răspunsurile celor 3 axe se includ în spațiul următor:

|  |
| --- |
|  |

3.8 Se estimează timpul tranzitoriu, suprareglajul și eroarea staționară și se introduc în tabelul următor (marcând corespunzător unitățile de măsură):

|  |  |  |  |
| --- | --- | --- | --- |
|  | **tt** | **σ** | **εst** |
| **X** |  |  |  |
| **Y** |  |  |  |
| **Z** |  |  |  |

3.9 Se interpretează rezultatele, observațiile se includ în spațiul următor:

|  |
| --- |
|  |

***4. Pornirea instalației***

4.1 Se alimentează și se pornește calculatorul.

4.2 Se alimentează cutia de alimentare a macaralei prin comutarea butonului I/O pe poziția “I” din partea din spate a acesteia, ca în imaginea următoare:



4.3 Se pornește placa de achiziție de date prin comutarea butonului On/Off pe poziția “On” din partea frontală a acesteia, ca în imaginea următoare:



4.4 Se confirmă că ciuperca de oprire de urgență a macaralei este eliberată prin rotirea inelului negru în sensul indicat de săgeată, ca în imaginea următoare:



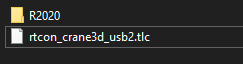
5.5 Se pornește cutia de alimentare a macaralei prin apăsarea butonului “Power On” din partea frontală a acesteia și se confirmă că LED-ul roșu (Power) rămâne aprins, ca în imaginea următoare:



4.6 Se pornește Matlab si se navighează în locația:

*D:\Crane*

4.7 Se păstrează în acest folder doar folderul *R2020* și fișierul *rtcon\_crane3d\_usb2.tlc*, ca în imaginea următoare:



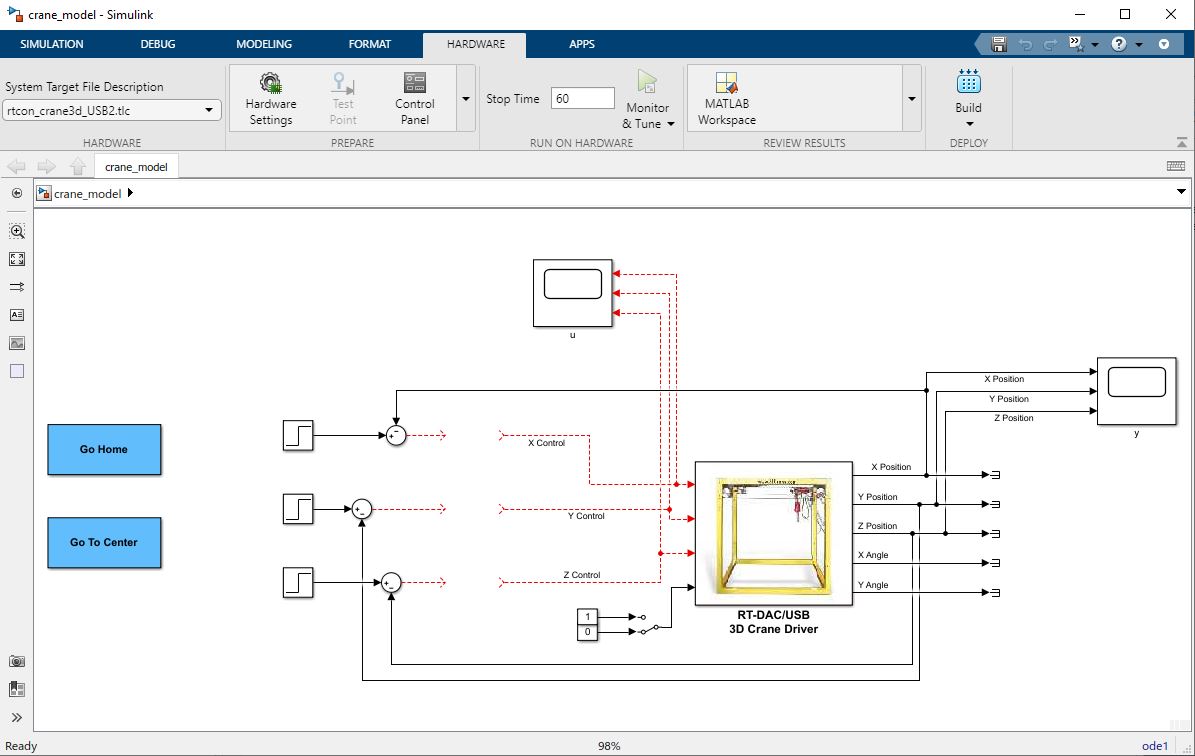
***5. Testare regulatoare pe instalație – criteriul simetriei***

5.1 Se descarcă de pe Moodle fișierul *crane\_model.slx*

5.2 Se mută acest fișier în locația:

*D:\Crane*

5.3 Se deschide fișierul în Simulink, ca în imaginea următoare:



**Observație:** Se poate observa că în acest caz au fost alese pentru cele 3 referințe (X, Y, Z), trepte de amplitudine 0.6. De asemenea, timpul de rulare a experimentului a fost ales 60 sec.

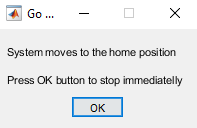
5.4 Se copiază în cele 3 spații goale cele 3 subsisteme cu regulatoarele PID opținute prin **criteriul simetriei** (cele implementate în pasul de simulare)

5.5 Se rulează scriptul corespondent pentru configurarea acestora (cel implementat în pasul de simulare)

5.6 Se execută o procedură de inițializare a encoderelor de poziție prin apăsarea (dublu click) butonului “Go Home” din interfață, ca în imaginea următoare:



5.7 Macaraua începe să se miște, iar pe ecran apare fereastra din imaginea următoare:



5.8 Se așteaptă finalizarea mișcării (și a dispariției ferestrei de mai sus) și **NU** se apasă butonul “OK” și nici nu se închide decât în caz de urgență (observați probleme cu macaraua în timp ce se mișcă).

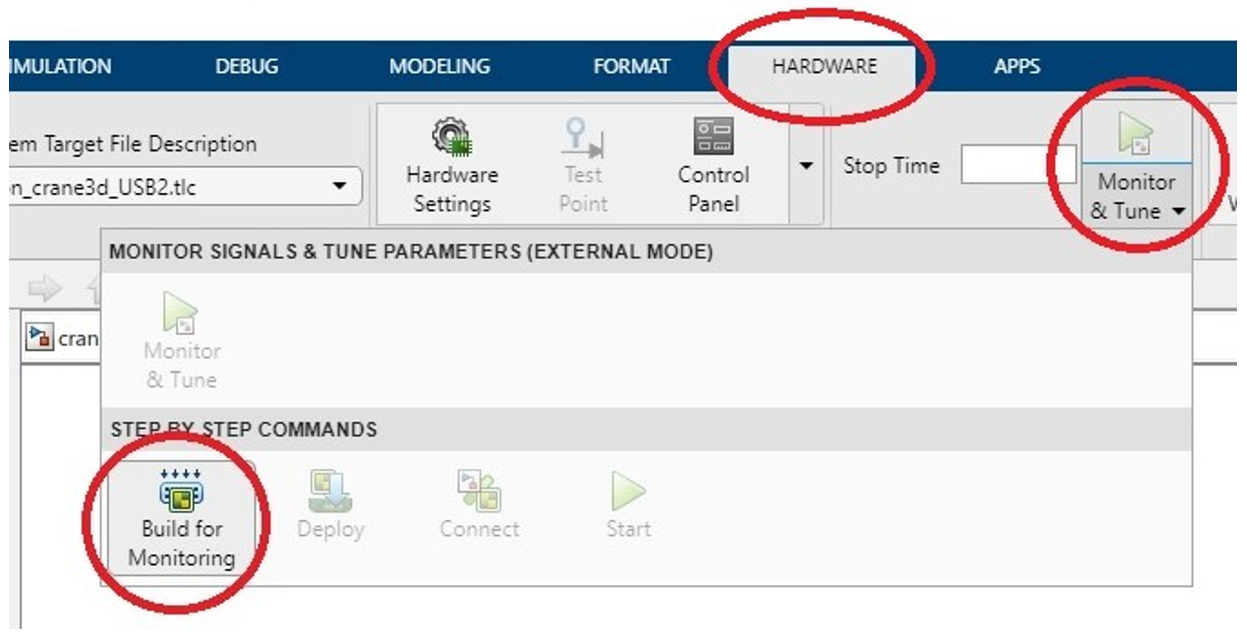
5.9 Se execută o procedură de mișcare “Go To Center” din interfață, ca în imaginea următoare (cu aceleași observații în timpul mișcării ca mai sus):



5.10 În schema Simulink se navighează în interfață:

*HARDWARE >> Monitor & Tune >> Build for Monitoring*

pentru a compila proiectul, ca în imaginea următoare:



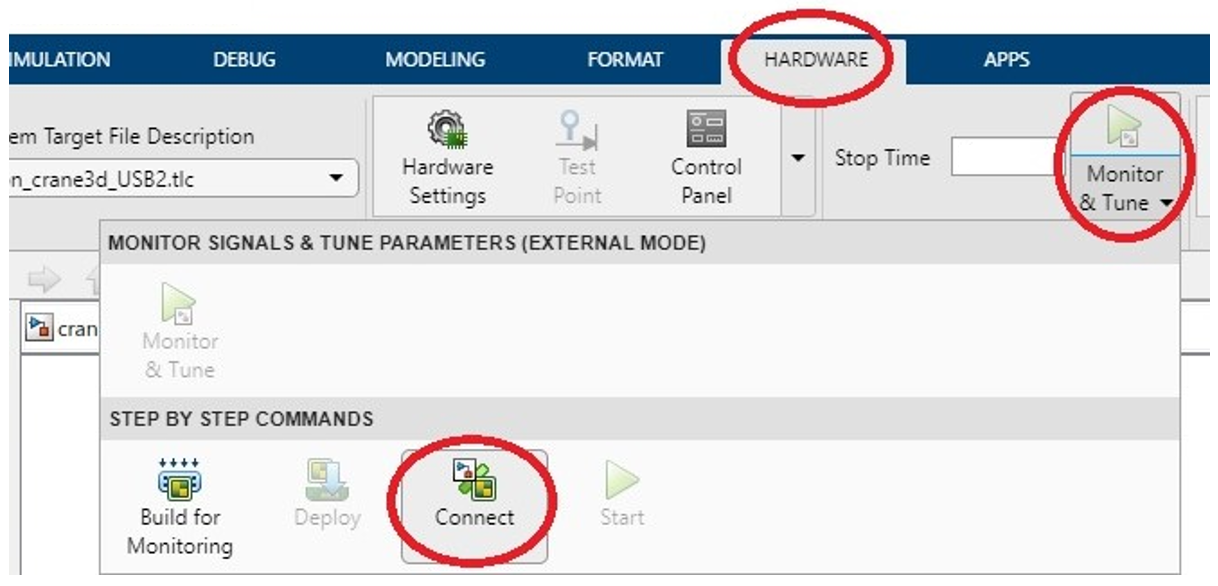
5.11 Se așteaptă finalizarea compilării până la apariția mesajului “Ready” în colțul stânga – jos a ferestrei, ca în imaginea următoare:



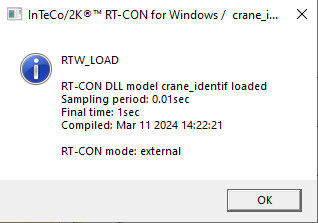
5.12 Se navighează în interfață:

*HARDWARE >> Monitor & Tune >> Connect*

pentru a porni experimentul în timp real pe instalație, ca în imaginea următoare:



5.13 Pe ecran apare următoarea fereastră, iar pentru pornirea experimentului se apasă butonul “OK”



5.14 Macaraua începe să se miște și se așteaptă finalizarea mișcării.

|  |
| --- |
| **Observație:** Ori de căte ori este necesară rularea unui nou experiment, se execută secvențial următorii pași:  - Se execută o procedură “Go To Center”  - Se compilează modelul (daca au fost realizate modificări în acesta)  - Se rulează modelul |

5.15 Răspunsurile celor 3 axe și comenzile de pe „Scope” se includ în spațiul următor:

|  |
| --- |
|  |

5.16 Se estimează timpul tranzitoriu, suprareglajul și eroarea staționară (marcând corespunzător unitățile de măsură):

|  |  |  |  |
| --- | --- | --- | --- |
|  | **tt** | **σ** | **εst** |
| **X** |  |  |  |
| **Y** |  |  |  |
| **Z** |  |  |  |

5.17 Se interpretează rezultatele, observațiile se includ în spațiul următor:

|  |
| --- |
|  |

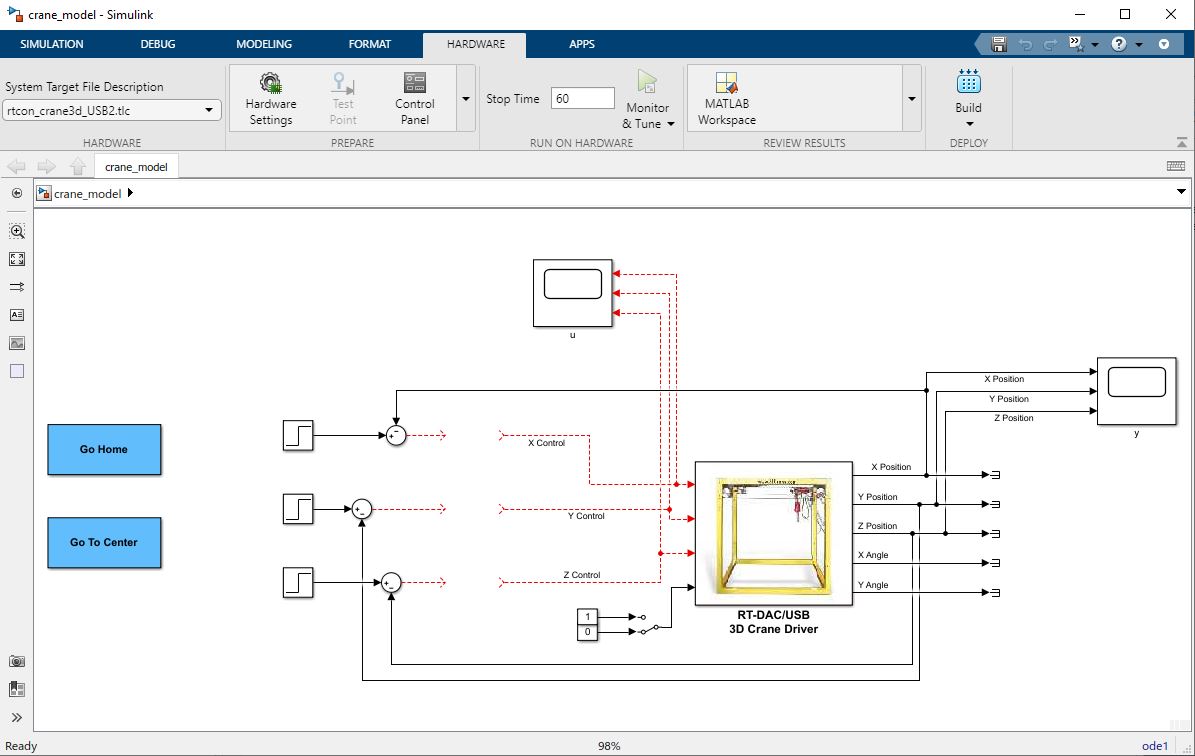
***6. Testare regulatoare pe instalație – metoda poli-zerouri***

6.1 Se pornește de la un fișier *crane\_model.slx* ***nou***

6.2 Se mută acest fișier în locația:

*D:\Crane*

6.3 Se deschide fișierul în Simulink, ca în imaginea următoare:



**Observație:** Se poate observa că în acest caz au fost alese pentru cele 3 referințe (X, Y, Z), trepte de amplitudine 0.6.

6.4 Se configureaza timpul de rulare a experimentului la 15 sec.

6.5 Se copiază în cele 3 spații goale cele 3 subsisteme cu regulatoarele PID opținute prin **metoda poli-zerouri** (cele implementate în pasul de simulare)

6.6 Se rulează scriptul corespondent pentru configurarea acestora (cel implementat în pasul de simulare)

6.7 Se rulează noul experiment (similar ca la pasul 5):

* Go Home
* Go To Center
* Compilare model
* Rulare model

6.8 Răspunsurile celor 3 axe și comenzile de pe „Scope” se includ în spațiul următor:

|  |
| --- |
|  |

6.9 Se estimează timpul tranzitoriu, suprareglajul și eroarea staționară (marcând corespunzător unitățile de măsură):

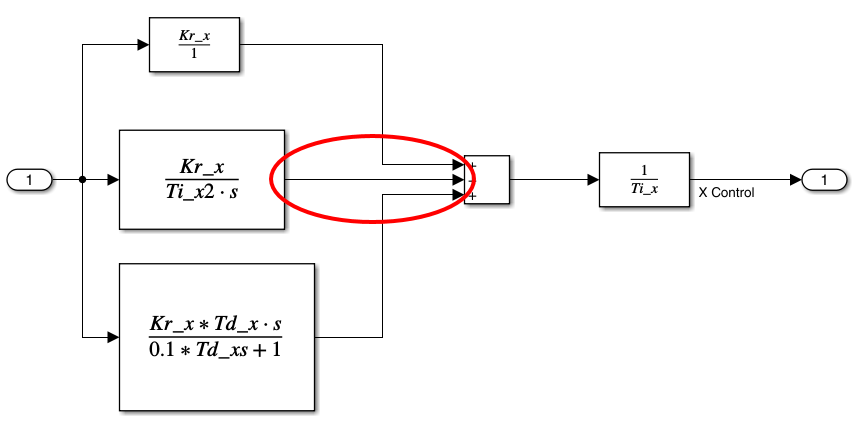
|  |  |  |  |
| --- | --- | --- | --- |
|  | **tt** | **σ** | **εst** |
| **X** |  |  |  |
| **Y** |  |  |  |
| **Z** |  |  |  |

***7. Acordare experimentală regulatoare***

**Observație:** analizând răspunsul celor 3 axe, se poate observa că se obțin erori staționare nenule. Acest fapt se datorează unei neliniarități inerente instalației denumită **zonă de insensibilitate (*pentru comenzi mici motoarele nu se rotesc*)**: se poate observa pe graficul comezii că după ce sistemul s-a stabilizat, deși eroarea este diferită de zero și în consecință comenzile sunt ușor pozitive, ieșirea nu mai evoluează pentru a aduce eroarea către zero.

**Observație:** pentru a rezolva această problemă, se introduce suplimentar un integrator, care va “acumula” această eroare nenulă pentru a determina creșterea comenzii și deci compensarea acesteia.

7.1 Se conectează componenta integrativă a regulatoarelor, ca în imaginea următoare:



7.2 Se setează prin încercări (în script-ul de configurare a PID-ului) mai multe valori (recomandare: începând cu **1** crescător) pentru constanta de timp de integrare suplimentară (Ti2) și se testează pe instalație analizându-se rezultatele.

|  |
| --- |
| **Observație:** Ori de căte ori este necesară rularea unui nou experiment, se execută secvențial următorii pași:  - Se execută o procedură “Go To Center”  - Se compilează modelul (daca au fost realizate modificări în acesta)  - Se rulează modelul |

Se completează observațiile (despre stabilitate, performanțe) în tabelul următor:

|  |  |
| --- | --- |
| **Valoare Ti2** | **Observații** |
| 1 |  |
| … |  |
| … |  |

7.3 Se dorește în continuare analiza efectelor introducerii unei **componente derivative**, în încercarea de a îmbunătăți suplimentar performanțele. Se setează prin încercări (în script-ul de configurare a PID-ului) mai multe valori (recomandare: începând cu **0.1** crescător și nu până la valori mari) pentru constanta de timp de derivare (Td) și se testează pe instalație analizându-se rezultatele.

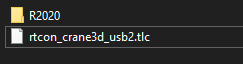
Se completează observațiile (despre stabilitate, performanțe) în tabelul următor:

|  |  |
| --- | --- |
| **Valoare Td** | **Observații** |
| 0.1 |  |
| … |  |
| … |  |

***8. Oprirea instalației***

8.1 Se închide Matlab.

8.2 Se șterg din locația *D:\Crane* toate folderele și fișierele cu excepția folderului *R2020* și a fișierului *rtcon\_crane3d\_usb2.tlc*, ca în imaginea următoare:



8.3 Se oprește placa de achiziție de date prin comutarea butonului On/Off pe poziția “Off” din partea frontală a acesteia, ca în imaginea următoare:



8.4 Se oprește alimentarea cutiei de alimentare a macaralei prin comutarea butonului I/O pe poziția “O” din partea din spate a acesteia, ca în imaginea următoare:



8.5 **Nu** se oprește calculatorul