**Identificarea experimentală a modelului dinamic pentru instalatia Servo INTECO (reglare poziție)**

***Fișă de laborator***

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

**Mod de lucru:** În cele ce urmează, se va efectua identificarea experimentală a platformei de laborator Servo INTECO ce poate fi aproximată cu un sistem de ordinul I cu integrator. Modelul generic astfel căutat are forma:

Aceste tipuri de procese nu sunt stabile (datorită prezenței integratorului) și atunci nu se poate utiliza procedura standard de identificare experimentală pentru procesele de ord. I. Spre exemplu, datorită instabilității, nu este posibilă trasarea caracteristicii statice.

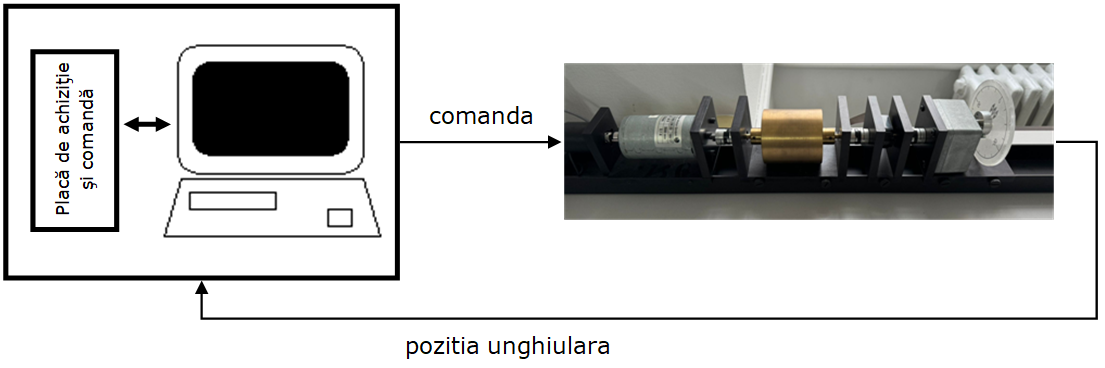
Pentru identificarea experimentală a acestor tipuri de sisteme instabile, se poate utiliza o procedură descrisă conceptual în următorii pași:

|  |
| --- |
| **Identificarea experimentală a proceselor instabile**   1. Se aplică la intrarea procesului o treaptă de comandă (de amplitudine corespunzătoare) pentru un scurt interval de timp astfel încât să se asigure că sistemul rămâne în domeniul nominal de funcționare (în cazul instalației Servo INTECO, pentru reglarea poziției unghiulare nu există capete de cursă și astfel motorul se poate roti oricât). 2. Se achiziționează și salvează evoluția ieșirii / ieșirilor. 3. Se construiește în simulare un model ce are forma cunoscută dar parametri () necunoscuți. 4. Prin încercări repetitive, se aleg valori pentru și 5. Pentru fiecare set de parametri, se simulează răspunsul la treaptă (de amplitudine egală ca în cazul experimental) al modelului și se urmărește potrivirea (suprapunerea) celor două grafice (cel real vs. cel simulat) 6. Se repetă procedura până la obținerea unui set de parametri cu potrivire cât mai bună.   **Observație:** Pentru potrivirea celor două grafice, se are în vedere faptul că ieșirea din simulare la momentul inițial are valoarea **0**, iar cea din datele experimentale poate fi diferită de **0**. Pentru aceasta, datele din simulare se translatează cu valoarea :    De asemenea, se are în vedere și timpul de rulare al experimentului astfel încât timpul de rulare a simulării să fie același. |

**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 **Servo INTECO** este o platformă didactică destinată studiului unui proces tehnologic, în care se urmăreşte controlul poziției unghiulare a unui ansamblu cinematic. Structura implementează un sistem numeric de reglare automată a mărimilor din proces.

Instalația are următoarea structură:



**Pași de lucru:**

***1. Pornirea instalației***

1.1 Se alimentează și se pornește calculatorul.

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



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



1.4 Se confirmă că ciuperca de oprire de urgență a instalației este eliberată prin rotirea acesteia în sensul indicat de săgeți, ca în imaginea următoare:



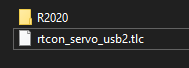
1.5 Se pornește cutia de alimentare a instalației prin apăsarea butonului “Power On” din partea frontală (dinspre perete) a acesteia și se confirmă că LED-ul roșu (Power) rămâne aprins, ca în imaginea următoare:



1.6 Se pornește Matlab si se navighează în locația:

*D:\Servo*

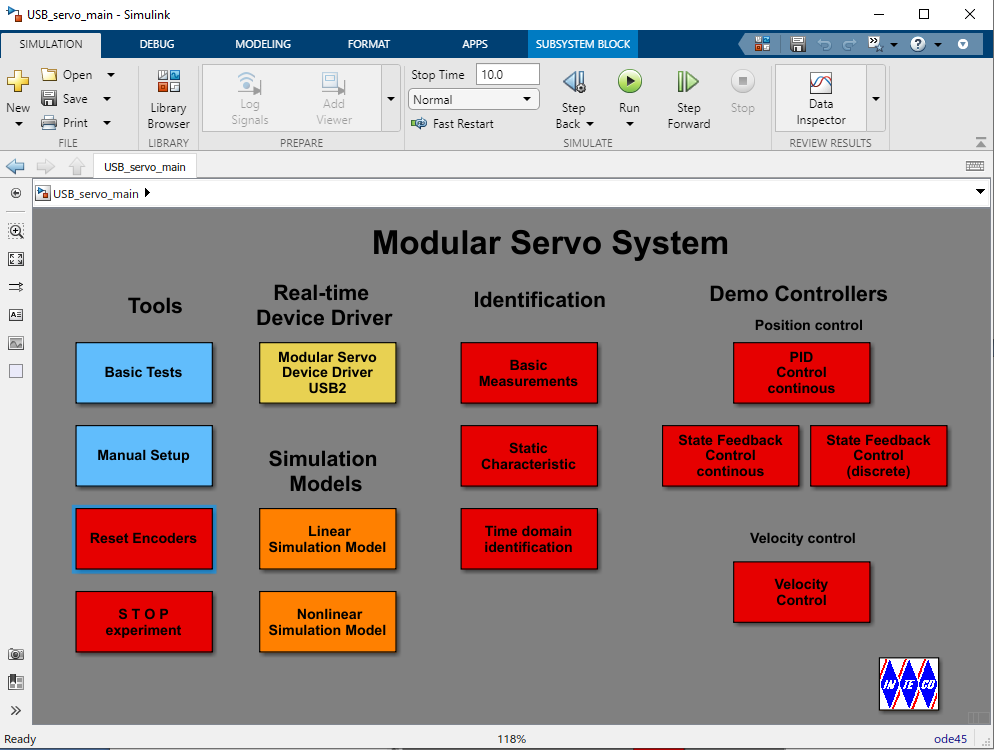
1.7 Se păstrează în acest folder doar folderul *R2020* și fișierul *rtcon\_servo\_usb2.tlc*, ca în imaginea următoare:



1.8 În linia de comandă Matlab se introduce comanda:

>> servo

și se confirmă că se deschide fereastra:



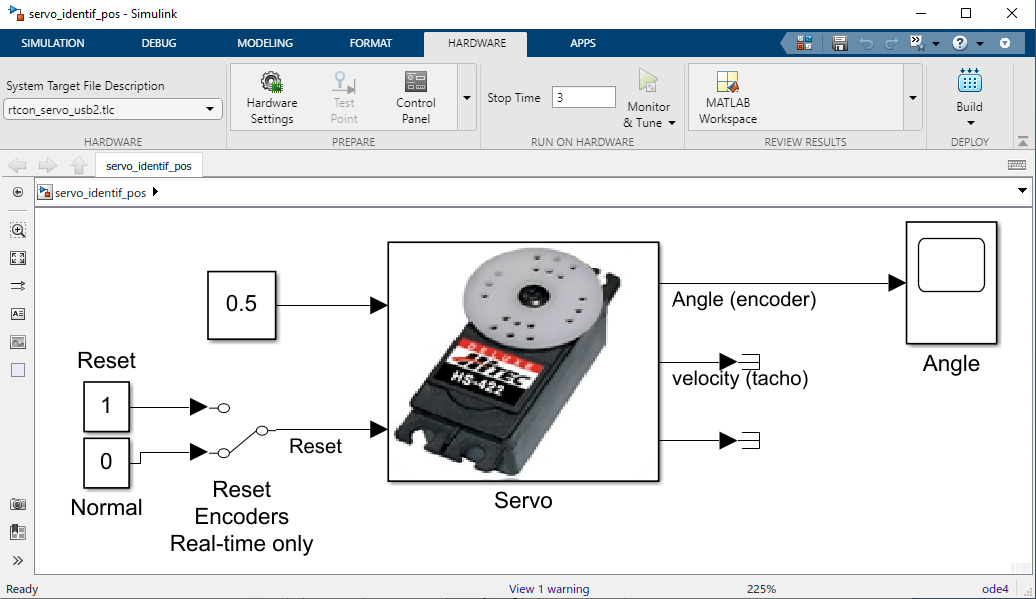
***2. Realizarea experimentului pentru achiziția datelor***

2.1 Se descarcă de pe Moodle fișierul *servo\_identif\_pos.slx*

2.2 Se mută acest fișier în locația:

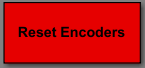
*D:\Servo*

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



**Observație:** Se poate observa că în acest caz a fost aleasă o comandă de tip treaptă de amplitudine 0.5. De asemenea, timpul de rulare a experimentului a fost ales 3 sec.

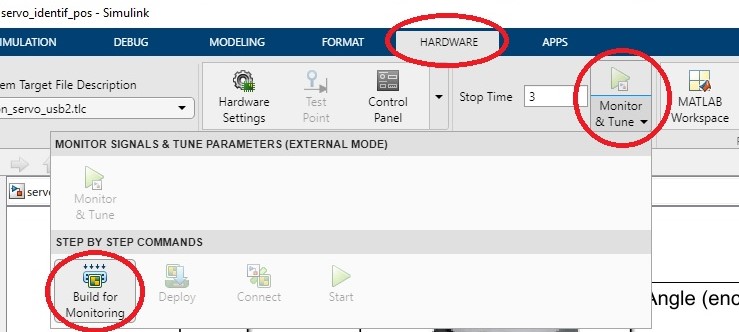
2.4 Se execută o procedură de inițializare a encoderului de poziție unghiulară prin apăsarea (dublu click) a butonului “*Reset Encoders*” din interfață, ca în imaginea următoare:

****

2.5 În schema Simulink *servo\_identif\_pos.slx* se navighează în interfață:

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

pentru a compila proiectul, ca în imaginea următoare:



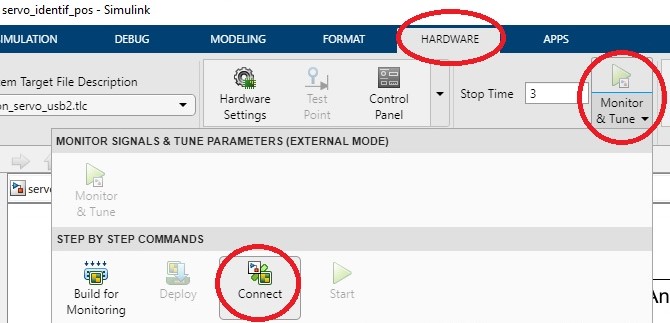
2.6 Se așteaptă finalizarea compilării până la apariția mesajului “Ready” în colțul stânga – jos a ferestrei, ca în imaginea următoare:



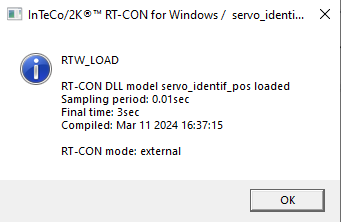
2.7 Se navighează în interfață:

*HARDWARE >> Monitor & Tune >> Connect*

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



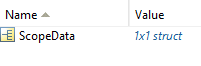
2.8 Pe ecran apare următoarea fereastră, iar pentru pornirea experimentului se apasă butonul “OK”



2.9 Servo-ul începe să se miște și se așteaptă finalizarea mișcării.

**Observație:** Ori de căte ori este necesară rularea unui nou experiment, este important ca înainte de asta să se realizeze o procedură “*Reset Encoders*”.

2.10 Se confirmă că în urma realizării experimentului în Workspace este salvată următoarea structură de date:



***3. Implementarea simulării modelului***

3.1 Se crează script-ul Matlab *script\_identif.m* în locația:

*D:\Servo*

3.2 Se închid eventualele plot-uri ramase deschise și se definește operatorul “s” pentru definirea modelelor ca funcții de transfer:

close all

s = tf('s');

3.3 Se configurează pentru funcția *step* amplitudinea treptei de comandă la valoarea *0.5*caîn cazul experimentului:

stepOpt = stepDataOptions('StepAmplitude', 0.5);

3.4 Pentru modelse definesc (prin încercări) valorile parametrilor și :

Kp = *valoarea aleasă*;

Tp = *valoarea aleasă*;

3.5 Se determină valoarea :

y0 = ScopeData.signals.values(1);

3.6 Se definește modelul dinamic cu valorile parametrilor alese mai sus:

Hp = Kp / (s\*(Tp\*s+1));

3.7 Se simulează răspunsul la treaptă (de aplitudine 0.5) pentru această funcție de transfer pentru un timp de 3 sec și se salvează datele în vectorii ysim, tsim:

[ysim, tsim] = step(Hp, 3, stepOpt);

3.8 Se plotează datele experimentale din structura *ScopeData*:

figure(1)

plot(ScopeData.time, ScopeData.signals.values)

hold on

3.9 Se plotează datele simulării având în vedere că acestea trebuie translatate cu valoarea :

plot(tsim, (ysim+y0));

title('Angular position');

legend('real data', 'sim data');

**4*. Realizarea identificării parametrilor modelului (****Kp, Tp****)***

4.1 Se rulează de mai multe ori script-ul și se aleg valori pentru parametri pana la obținerea unei suprapuneri cât mai precise.

4.2 Se completează graficul obținut în spațiul următor (marcând corespunzător etichetele pe axe și unitățile de măsură):

|  |
| --- |
|  |

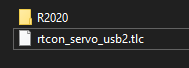
4.3 Se completează funcția de transfer în spațiul următor:

|  |
| --- |
|  |

***5. Oprirea instalației***

5.1 Se închide Matlab.

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



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



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



5.5 **Nu** se oprește calculatorul

***6. Concluzii și observații***

Se notează concluzii și observații: