**Proiectare regulator PID pentru instalația Servo INTECO**

***Fișă de lucru***

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

**Mod de lucru:** În cele ce urmează, se va efectua proiectarea regulatorului PID pentru instalația de laborator Servo INTECO, ce poate fi aproximată cu un sistem de ordinul I cu integrator.

Pentru acest proces, vom utiliza ca metodă de proiectare a regulatorului **Metoda poli-zerouri** - aplicată pentru procesul căruia i s-a neglijat integratorul (având în vedere că procesul conține deja integrator).

Pentru poziția unghiulară se consideră un comportament descris de următoarea funcție 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 **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:

O imagine care conține aparat casnic, aparat de bucătărie, Electrocasnic, captură de ecran

Descriere generată automat

**Pași de lucru:**

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

1.1 Să se proiecteze regulatorul, 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:

|  |
| --- |
|  |

1.2 Se notează forma legii de reglare PI obținute:

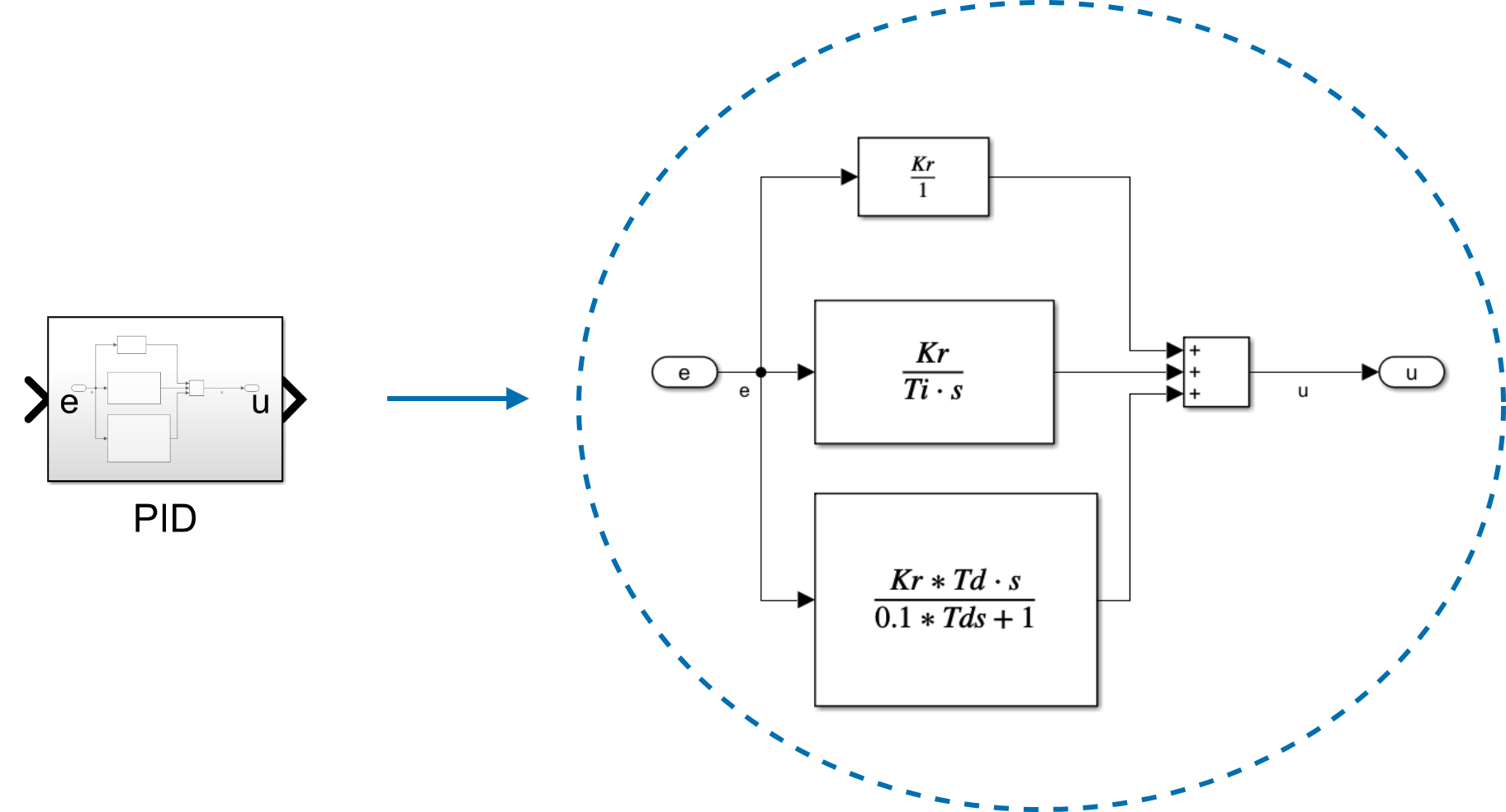
|  |
| --- |
| **HR(s)** |
|  |

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

|  |  |
| --- | --- |
|  | **HR(s)** |
| **KR** |  |
| **Ti** |  |

***2. Validare regulatoare în simulare***

2.1 Se implementează în Simulink structura de reglare cu un grad de libertate, unde procesul se implementează cu un bloc “Transfer fcn”, iar regulatorul se implementează ca un subsistem în forma PID paralel:

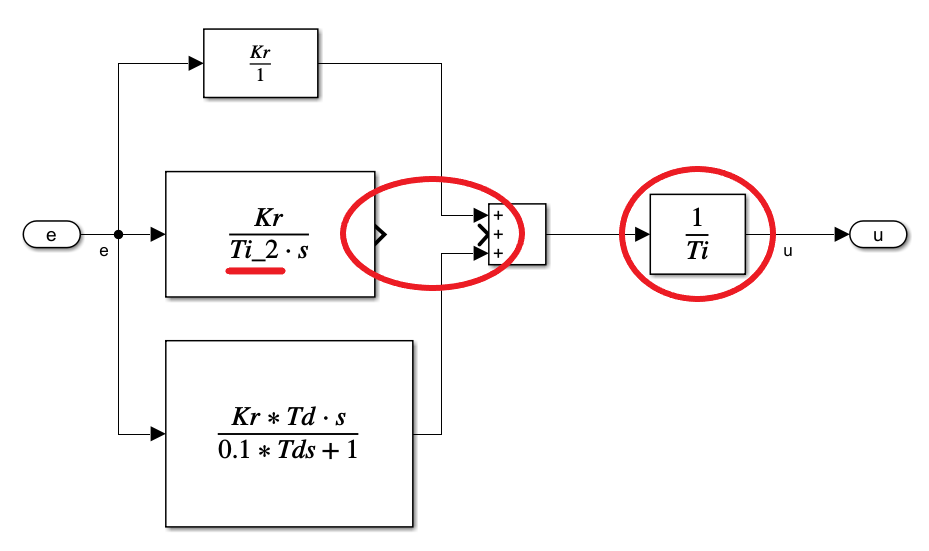


**Observație:** Parametrii KR, Ti și Td ai regulatoarelor se vor introduce ca variabile în funcțiile de transfer (Transfer fcn), urmând să fie inițializați cu valori numerice în *Workspace*.

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

2.2 Se modifică schema bloc astfel încât va exista practic doar un regulator P (componenta integrativă se află deja în proces), dar se va adăuga un factor suplimentar de amplificare pentru a compensa lipsa factorului :

Regulatorul se modifică astfel:



se implementează un **nou script Matlab** pentru configurarea parametrilor de acord, în 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:

|  |
| --- |
|  |

2.3 Se rulează simularea iar răspunsul se include în spațiul următor:

|  |
| --- |
|  |

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

|  |  |  |
| --- | --- | --- |
| **tt** | **σ** | **εst** |
|  |  |  |

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

|  |
| --- |
|  |

***3. Pornirea instalației***

3.1 Se alimentează și se pornește calculatorul.

3.2 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:



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



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



3.5 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:



3.6 Se pornește Matlab si se navighează în locația:

*D:\Servo*

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

O imagine care conține text, Font, captură de ecran, număr

Descriere generată automat

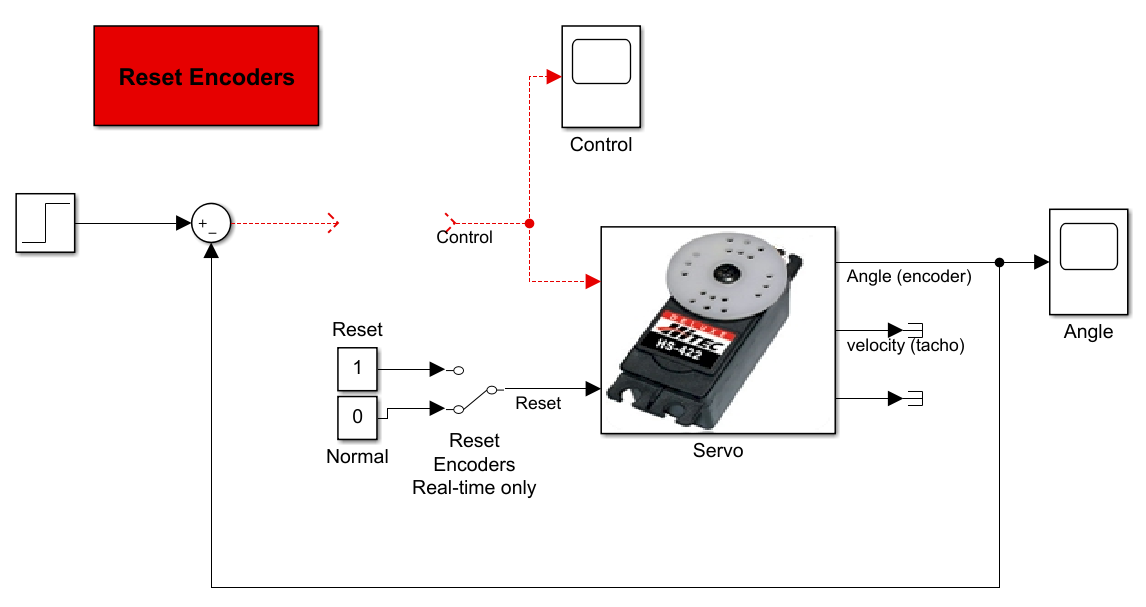
***4. Testare regulator pe instalație***

4.1 Se descarcă de pe Moodle fișierul *servo\_model.slx*

4.2 Se mută acest fișier în locația:

*D:\Servo*

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



**Observație:** Se poate observa că în acest caz au fost alese pentru referință o treaptă de amplitudine 100. De asemenea, timpul de rulare a experimentului a fost ales 60 sec.

4.4 Se copiază în spațiul gol subsistemul cu regulatorul PID obținut

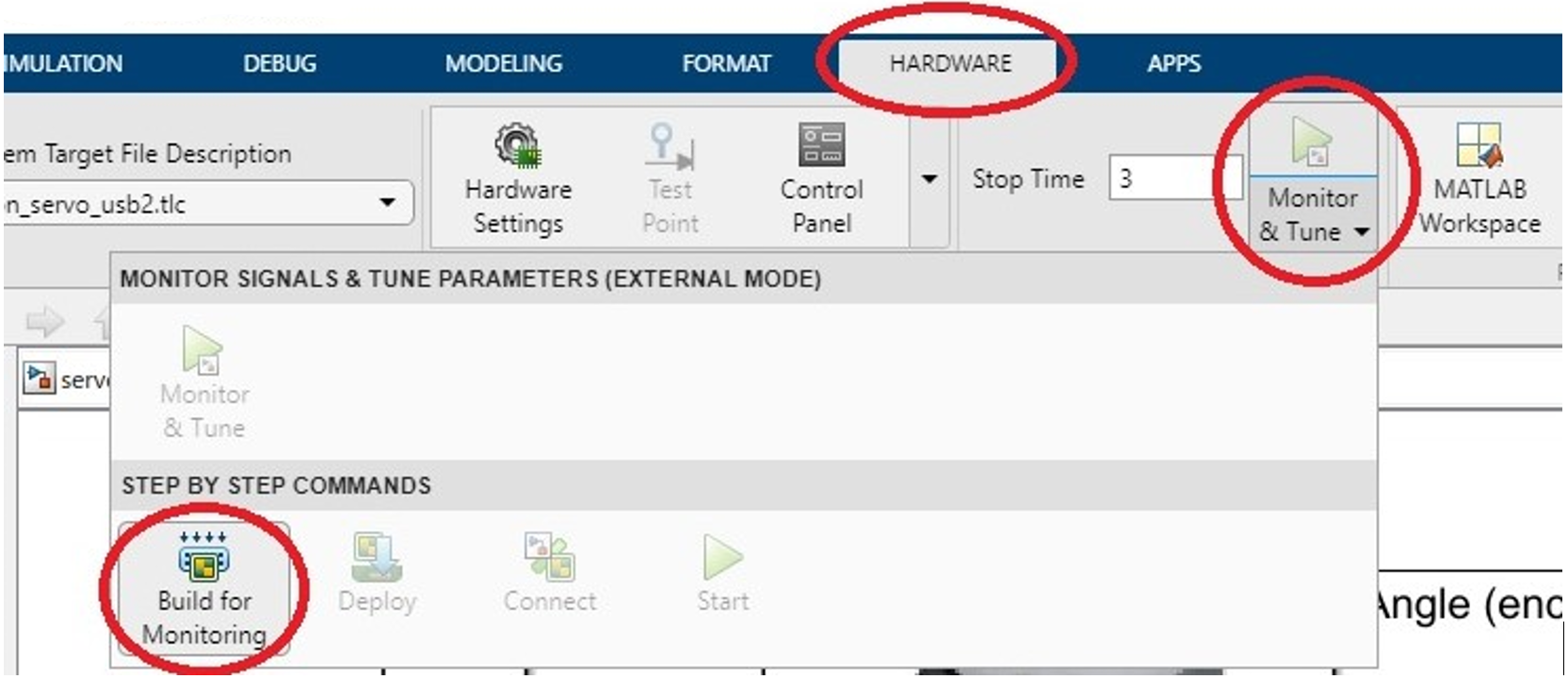
4.5 Se rulează scriptul corespondent pentru configurarea acestora (cel implementat în pasul de simulare)

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

4.7 În schema Simulink *servo\_model.slx* se navighează în interfață:

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

pentru a compila proiectul, ca în imaginea următoare:



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



4.9 Se navighează în interfață:

*HARDWARE >> Monitor & Tune >> Connect*

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

O imagine care conține text, captură de ecran, software, Pictogramă computer

Descriere generată automat

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

O imagine care conține text, captură de ecran, Font, software

Descriere generată automat

4.11 Servo-ul 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 “*Reset Encoders*”  - Se compilează modelul (daca au fost realizate modificări în acesta)  - Se rulează modelul |

4.12 Răspunsul sistemului și comanda de pe „Scope” se includ în spațiul următor:

|  |
| --- |
|  |

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

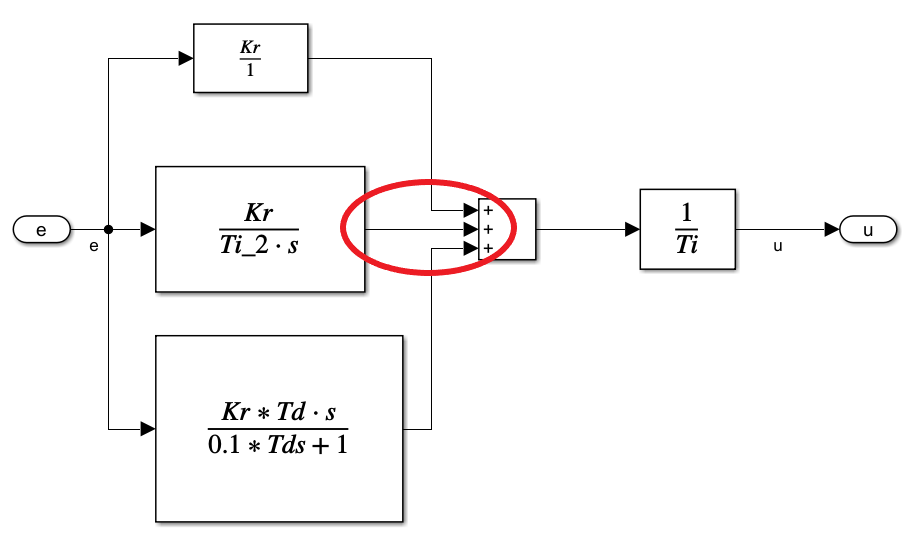
|  |  |  |
| --- | --- | --- |
| **tt** | **σ** | **εst** |
|  |  |  |

***5. Acordare experimentală regulatoare***

**Observație:** analizând răspunsul sistemului, 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.

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



5.2 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 “Reset Encoders”  - Se compilează modelul (daca au fost realizate modificari în acesta)  - Se rulează modelul |

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

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

5.3 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 |  |
| … |  |
| … |  |

***6. Oprirea instalației***

6.1 Se închide Matlab.

6.2 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:

O imagine care conține text, Font, captură de ecran, număr

Descriere generată automat

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



6.4 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:



6.5 **Nu** se oprește calculatorul