|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PROIECT INGINERIA REGLARII AUTOMATE II | | | | | |
| NUME student | Inesc Erik-Marius | GRUPA: | 30132 | Nota |  |

**Design, implementation and Testing of classical control structures (P9)**

Autor: Inesc Erik-Marius

Grupa**:** 30132

**AN UNIVERSITAR: 2021-2022**

Cuprins

[1. Scopul Proiectului 3](#_Toc104122922)

[a. Obiective 3](#_Toc104122924)

[b. Specificații 3](#_Toc104122925)

[2. Determinarea modelului matematic al sistemului 4](#_Toc104122926)

[a. Analiza sistemului 4](#_Toc104122927)

[b. Achizitie semnale 4](#_Toc104122928)

[c. Identificare / Modelare analitica 7](#_Toc104122929)

[3. Proiectarea sistemului de control 8](#_Toc104122930)

[4. Implementarea sistemului de control 11](#_Toc104122931)

[5. Testare și analiza rezultate 12](#_Toc104122932)

[6. Concluzii 14](#_Toc104122933)

# Scopul Proiectului

# Proiectul are ca si scop proiectarea, implementarea si testarea unei structuri clasice de control. Astfel proiectul va fi bazat pe proiectarea si implementarea unui regulator digital pentru controlul vitezei al unui motor de curent continuu.

## Obiective

• Montare circuit

• Determinare date experimentale cu ajutorul mediului de dezvoltare Arduino

• Preluare date experimentale și afișare sub formă de grafic în Matlab

• Modelarea procesului

• Determinarea funcției de transfer pe baza graficului

• Determinarea funcției de transfer a regulatorului

• Implementare și testare regulator în Arduino

• Preluare date și afișare sub formă de grafic în Matlab

## Specificații

**Descrierea pasilor care i-am urmat in realizarea proiectului:**

-Am montat circuitul si am determinat setul de date experimentale in mediul de dezvoltare Arduino. Cu ajutorul encoderului am citit datele provenite de la motor si am determinat numarul de rotatii ale rotitei si le-am afisat in Serial Monitor.

Determinarea functiei de transfer a procesului: Datele au fost preluate de la Arduino, folosind comunicarea prin serial, iar dupa aceea au fost afisate printr-un plot folosind Serial Plot din Arduino IDE. Pe baza acestui plot am facut identificare si am aflat functia de transfer a procesului.

Determinarea functiei de transfer a regulatorlui: ne-am folosit de metoda Guillemin-Truxal pentru a determina regulatorul și am impus timpul de raspuns (0.4 secunde) si suprareglajul (5), astfel incat sa avem un regulator optim. Iar mai apoi, am discretizat functia de transfer al regulatorlui.

Testare regulator in Arduino: Pentru a testa regulatorul in Arduino am determinat relatia de recurenta prin egalarea functiei de transfer a regulatorului in 𝑧 −1 cu 𝑐 𝑒 (c = comanda; e = eroarea) Astfel se obtine relatia de recurenta c[0] si se poate implementa regulatorul

-Testarea, simularea și implementarea în Matlab și Simulink a regulatorului proiectat, folosind modelul matematic obtinut anterior

**Rezultate dorite:**

Datele obtinute de la motor prin intermediul encoderului sa urmareasca intrare de tip PWM.

Obitinerea functiei de transfer de gradul 1 al procesului

Obtinirea functiei de transfer a regulatorului.

Obtinirea relatiei de recurenta a regulatorului.

# Determinarea modelului matematic al sistemului

## Analiza sistemului

Marimile manipulate sunt reprezentate de datele primite de la encoder inainte de calculul regulatorului,.

Marimile controlate sunt reprezentate de datele pentru viteza motorului dupa calculul regulatoruluiAchizitie semnale

## Achizitie semnale

Pentru realizarea circuitului am avut nevoie de urmatoarele componente :

* Placa Arduino Uno - Placa de dezvoltare UNO R3 cu microcontroller ATmega328p
* Breadboard - se foloseste pentru cablarea circuitelor electrice ajutand la realizarea extrem de rapida a montajelor.
* Encoder – Dispozitiv electromagnetic care masoara miscarea.
* Baterii AA alcaline 1.5V x4 bucati.
* Baterie 9V
* Suport baterii 4x AA
* Motor DC 3V-6V cu reductor 1:48
* Jumpere ( fire mama-mama si fire tata-mama)
* Fir conectare baterie 9V la Arduino
* Transistor P2N2222A –NPN
* Intrerupator
* Rezistenta 150ohm
* Am achizitionat un kit-robot cu 4 motoare (am folosit doar 1, celalte 2 le-am demontat si le-am lasat sa mearga in gol).
* Un sasiu din plexiglas
* Patru roti
* Suporti motoare + suruburi si piulite necesare asamblarii.

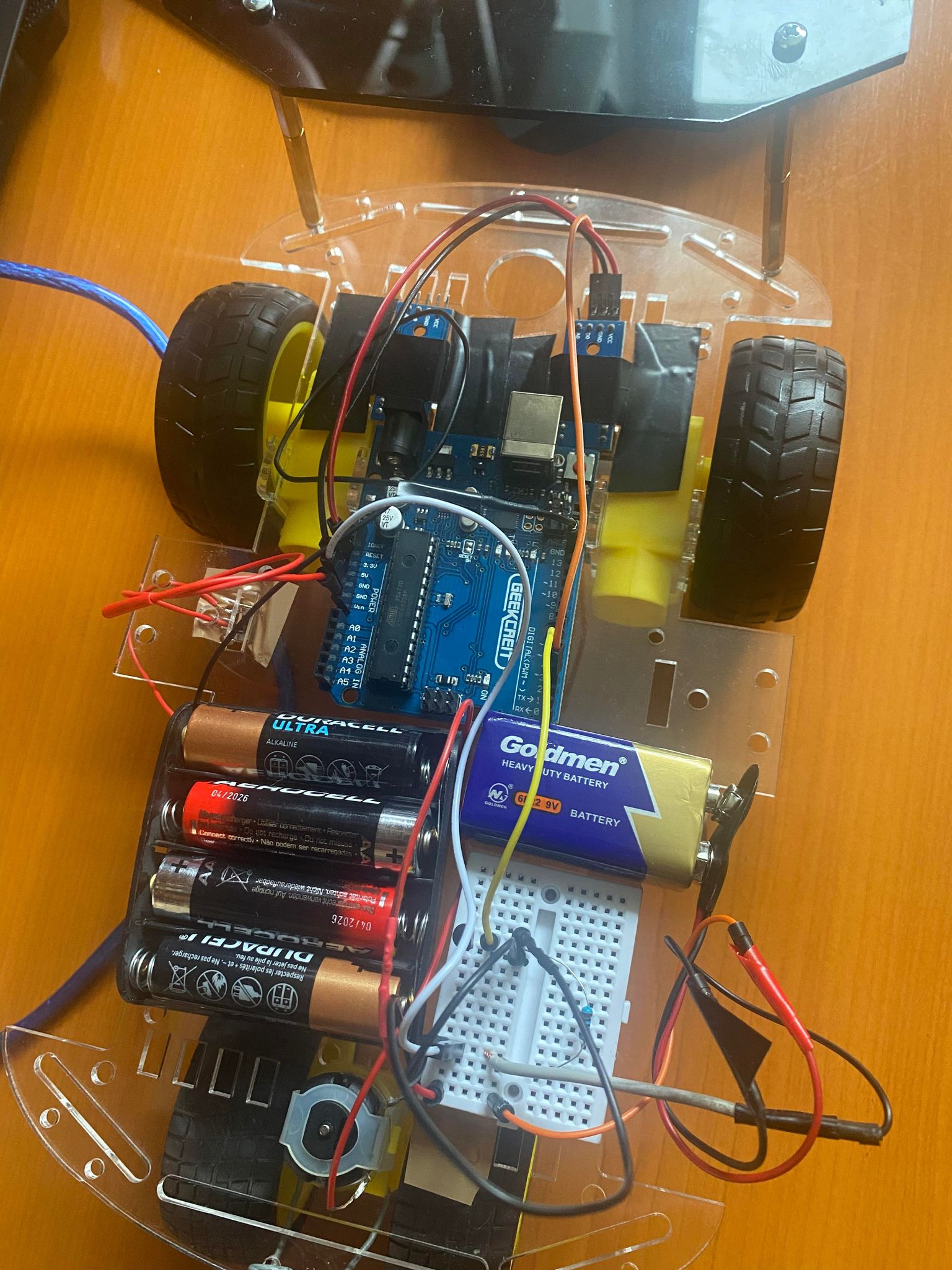
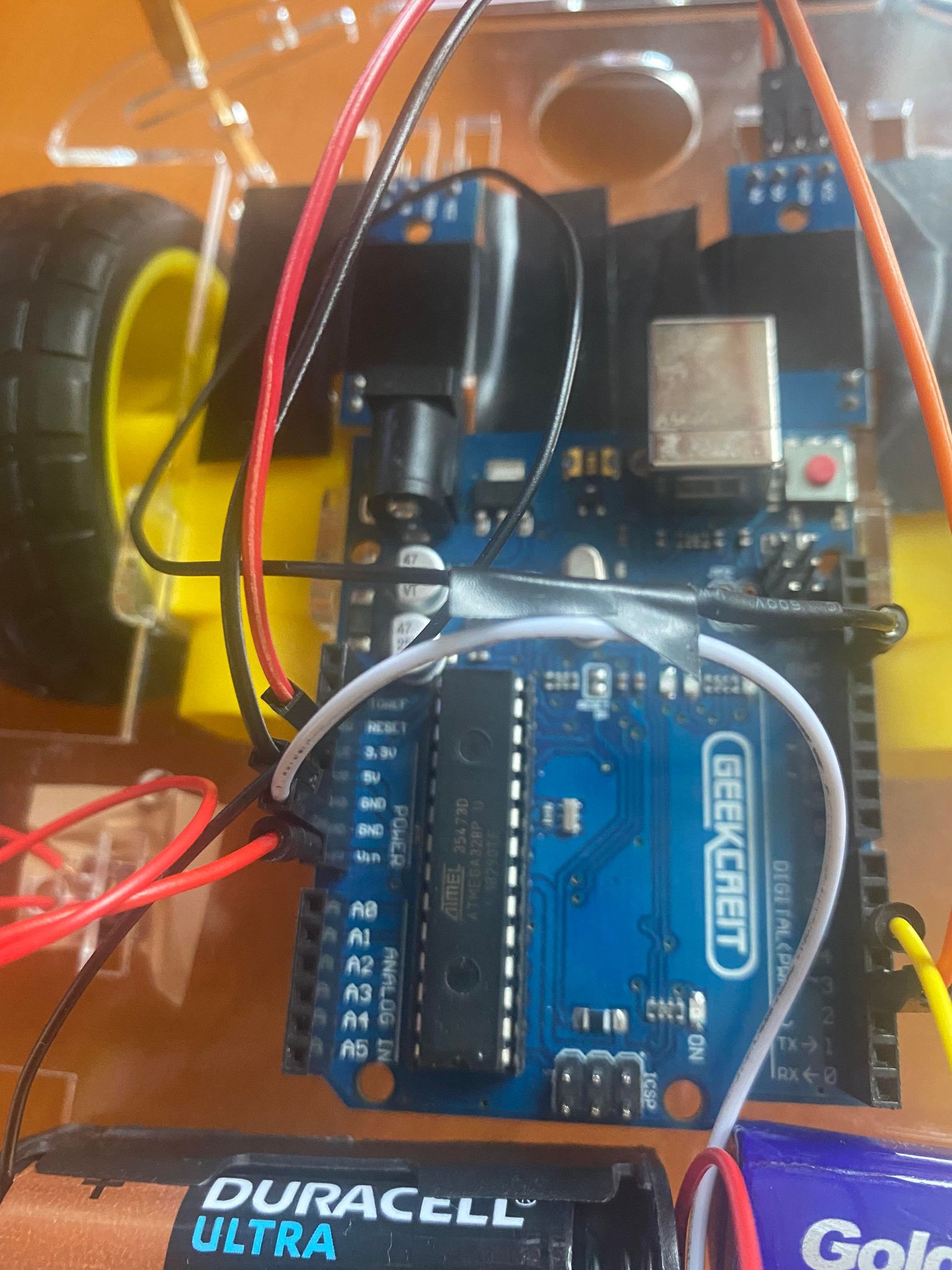
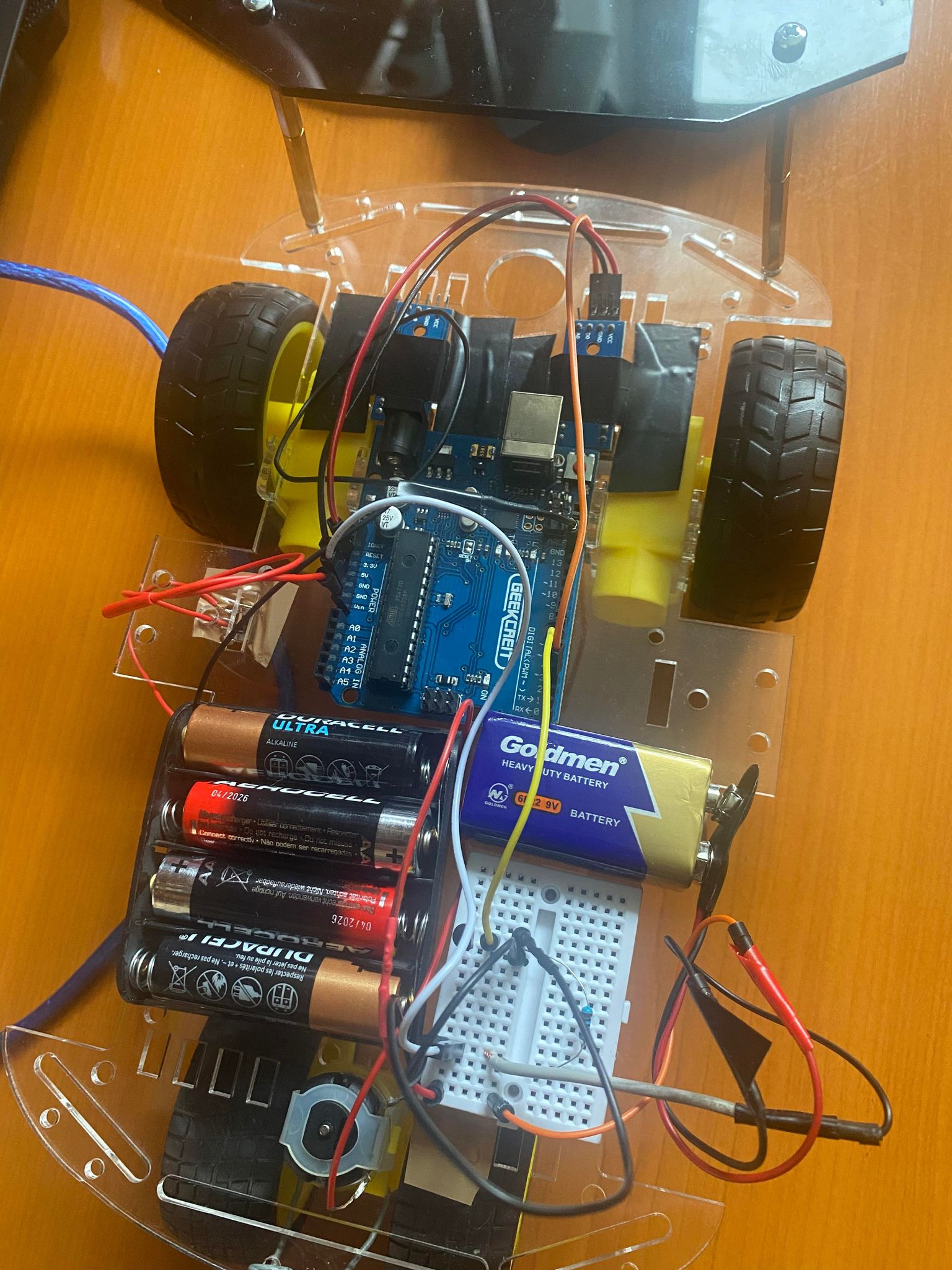
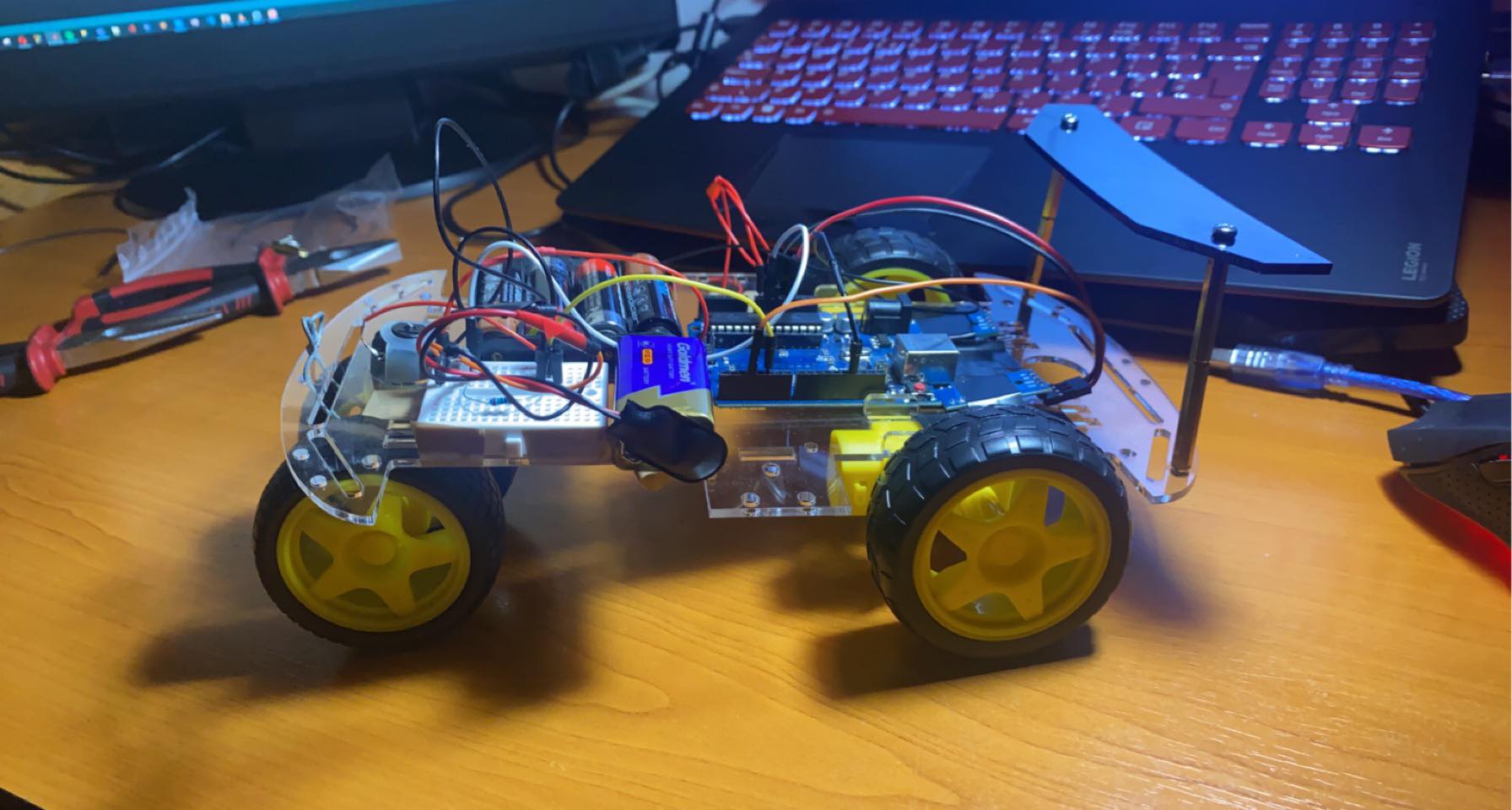
Procedura utilizata :

Vom realiza montajul pieselor si vom conecta microcontrolerul un encoder cat si un tranzistor NPN cu ajutorul unor jumpere. De obicei, pentru a controla un DC motor cu Arduino, ne folosim de punte H, care ne permite cresterea puterii de alimentare a motorului printr-un circuit separat.

In cazul de fata, nu am obtinut rezultatele dorite, deoarece probabil tensiunea de alimentare nu a fost destul de buna, sincronizarea motoarelor nu era perfecta cu puntea H, de aceea am ajuns la acest design, renuntand de la 4 motare, doar la 1, pentru a simplifica procesul.

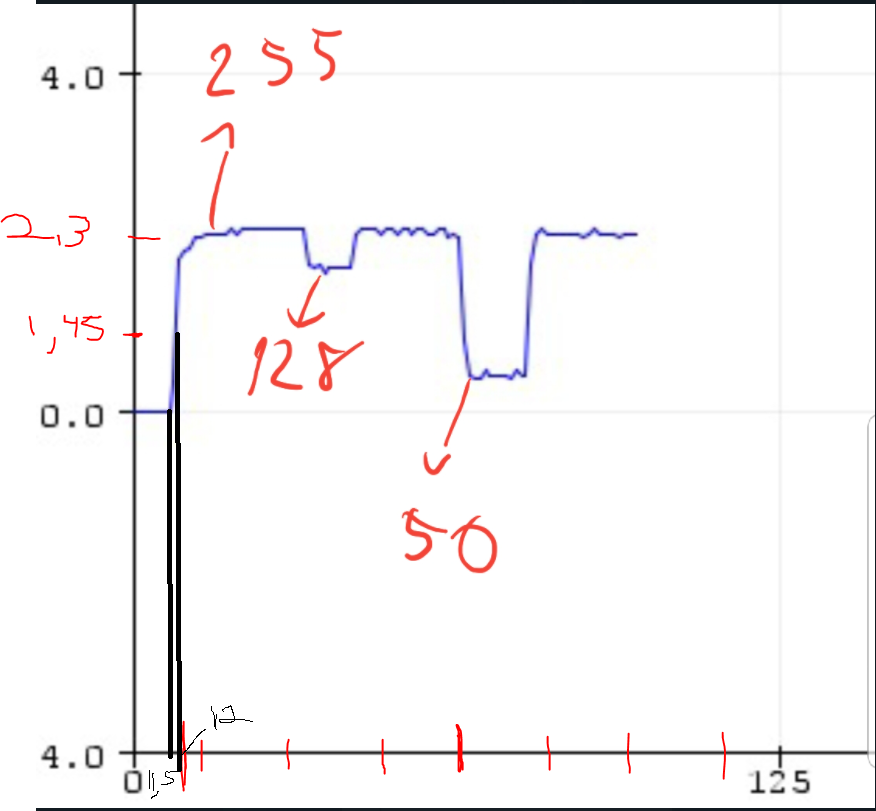
Astfel, am ajuns la folosirea unui tranzistor NPN, pentru simplificarea circuitului. Cu ajutorul tranzistorului am putut mari tensiunea cu care am alimentat motorul, in asa fel obtinand o viteza mai rdicata.

Encoderul dispune de un sezor optic care furnizeaza semnale electrice sub forma de trenuri de impulsuri, astfel miscarea masurata poate fi transmisa sub forma de valori in Arduino. Astfel datele de la motoras vor fi afisate in Serial Monitor.



## Identificare / Modelare analitica

Datele au fost preluate de la Arduino, folosind comunicarea prin serial, iar dupa aceea au fost afisate printr-un plot folosind Serial Plot din Arduino IDE.



Astfel obtinem functia de transfer de ordin 1 care are urmatoarea forma:

=

Factorul de proportionalitate : k=0.46 (-)/(-) = (2.3/5)

Constanta de timp : T=0.5 [s] (T2-T1) => (12-11.5).

# Proiectarea sistemului de control

Pentru proiectarea regulatorului folosim metoda Guillemin-Truxal.

Am impus un suprareglaj de 5% si un timp de raspuns de 0.4 secunde.

Determinăm ζ si pulsatia naturala cu formulele:

ζ =

=

Sistemul închis echivalent unuia de ordinul doi:

=

Functia de transfer in bucla inchisa este:

=

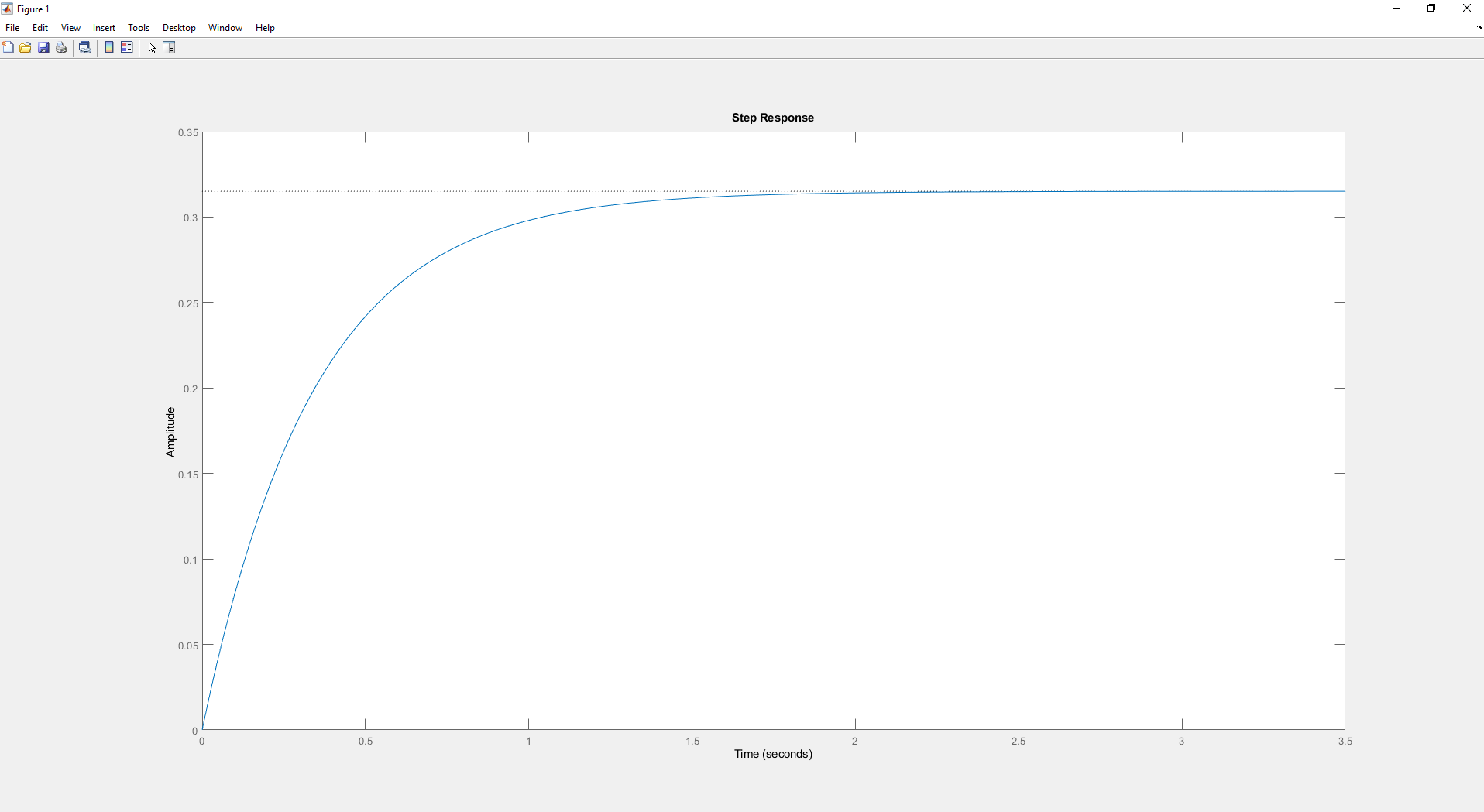
Utilizand urmatoarea formula obtinem functia de transfer a regulatorului:

=

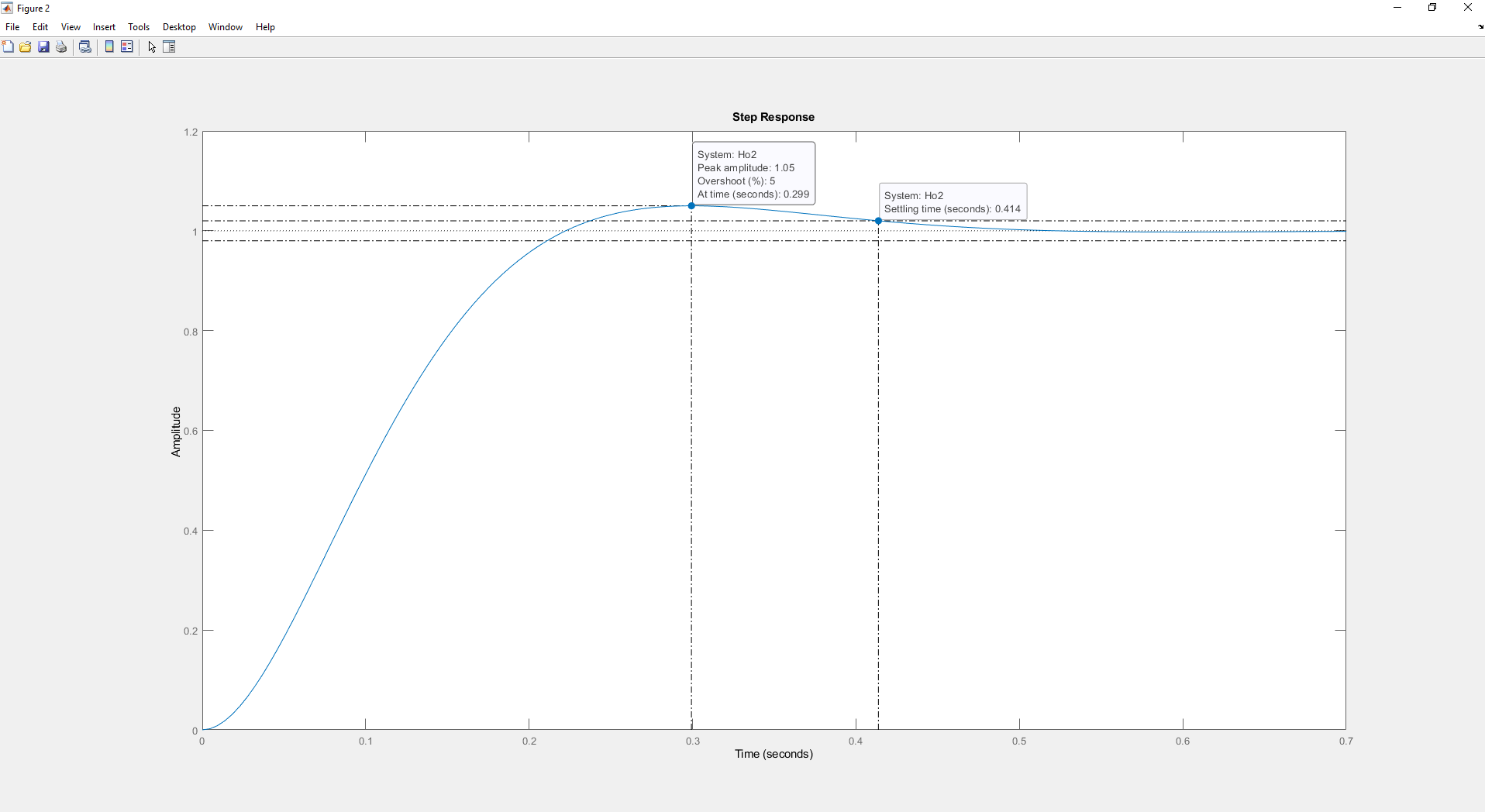
=minreal()

Functia de transfer a regulatorului este:

Raspunsul la treapta a functiei obtinute inainte de impunerea parametriilor:



Raspunsul la treapta a functiei obtinute dupa impunerea parametriilor:



Discretizarea sistemului s-a realizat utilizand metoda ZOH.

Conform Teoremei lui Shannon am ales o perioada de esantionare, Te=0.025, dupa care am folosit comanda c2d din Matlab.

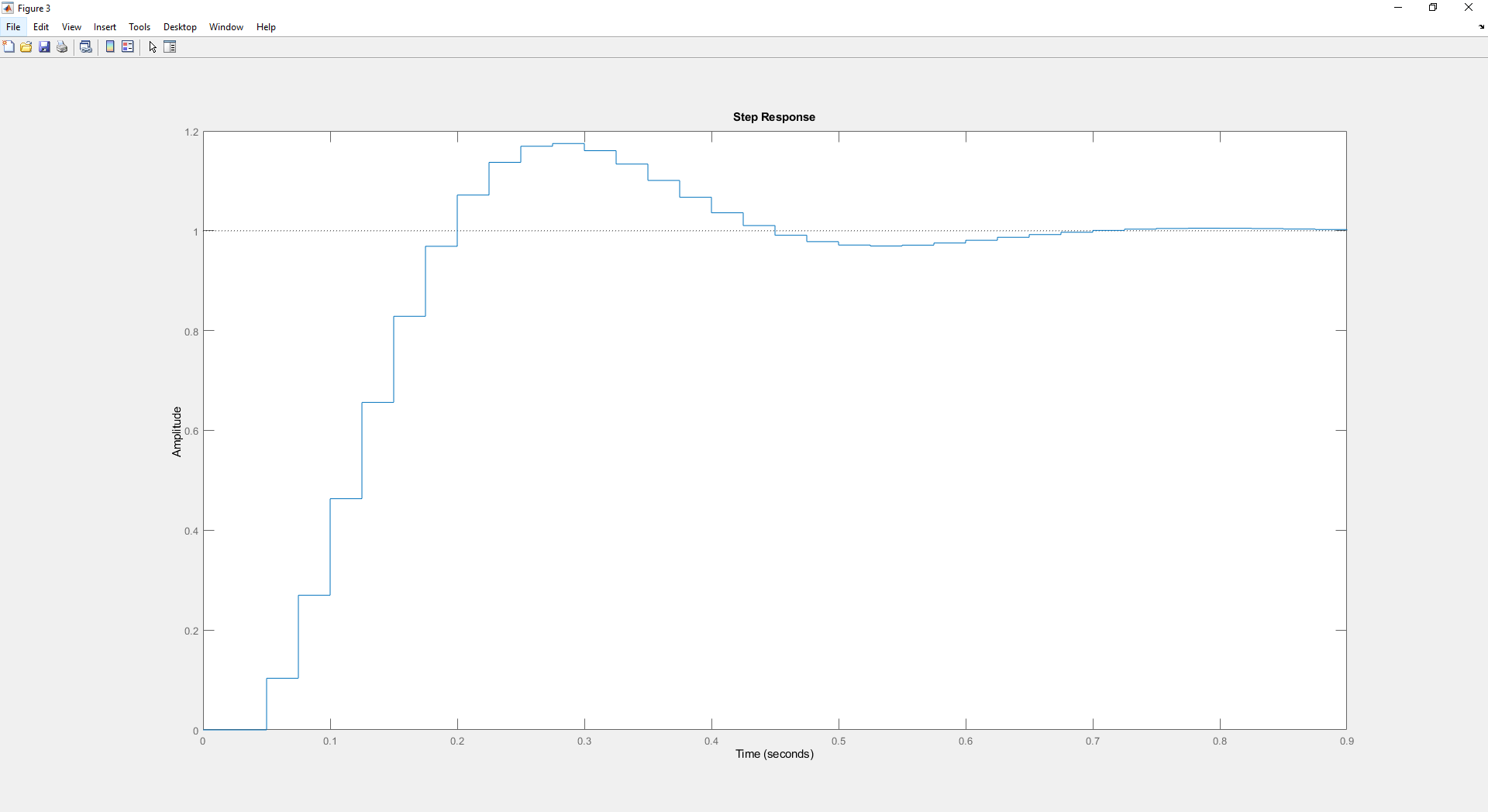
=

(z-1)= =

Aflam comanda si eroarea sistemului

c(k) = 4.612\* e(k-1)- 4.387\*5361e(k-2)+ 1.607\*c(k-1) - 0.6065\*c(k-2);

(k)=r(k)-y(k)



# Implementarea sistemului de control

Pentru implementarea sistemului de control am folosit:

Matlab pentru usurarea calculelor, aflarea functiilor de transfer a motorului si a regulatorului.

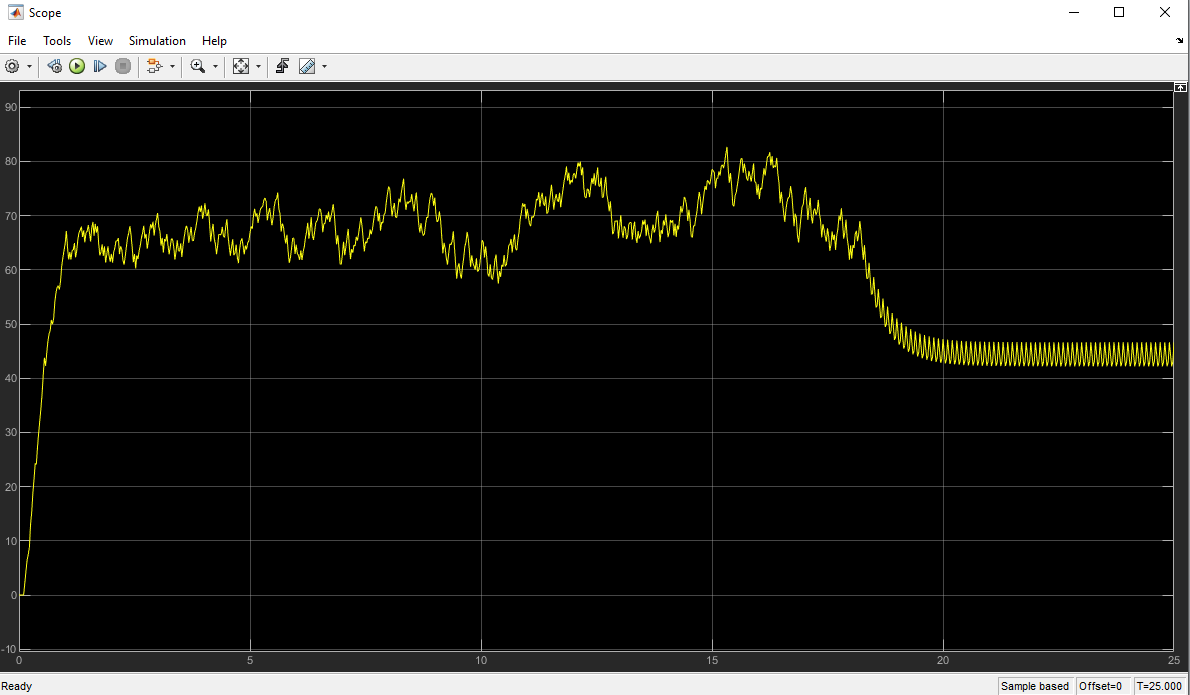
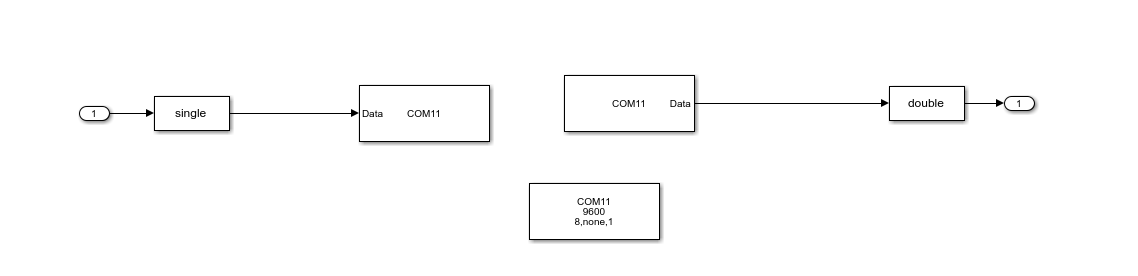
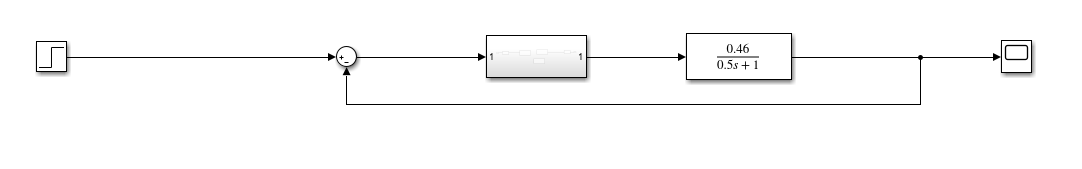
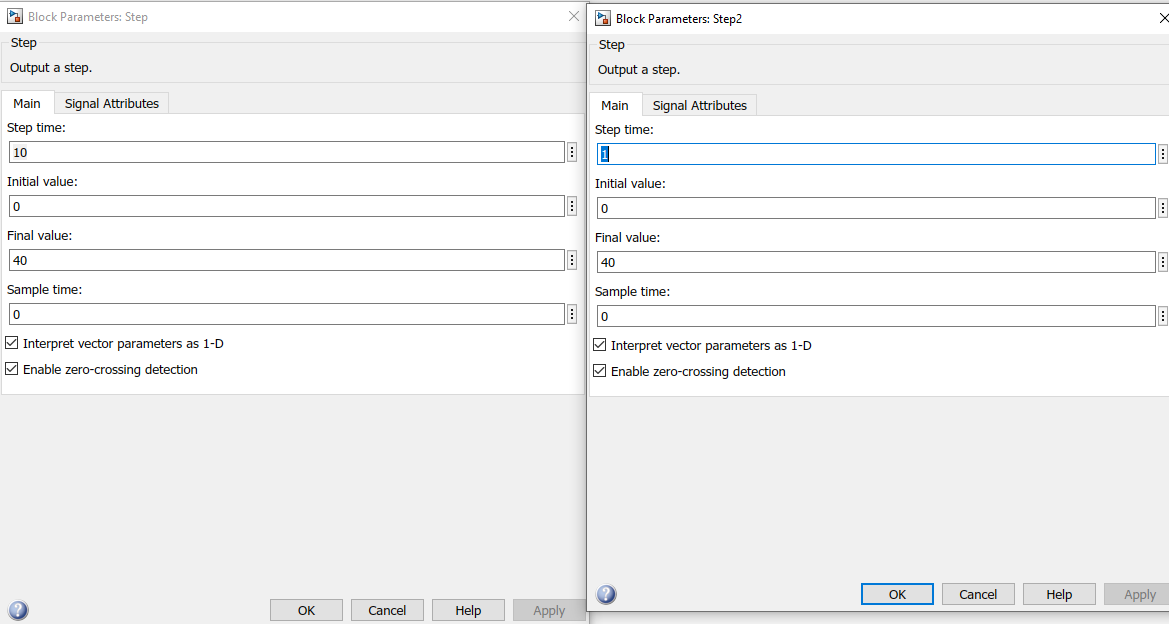
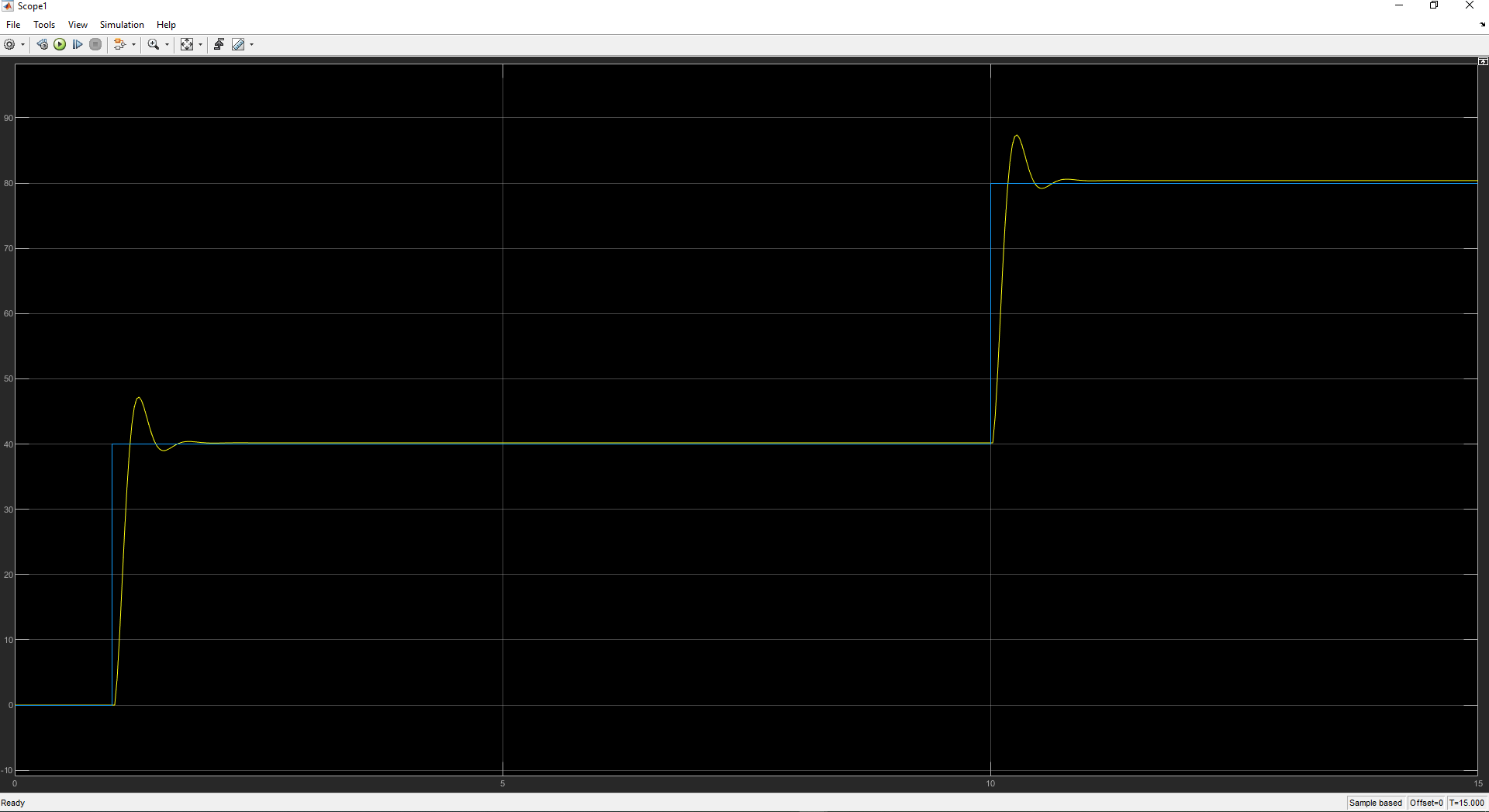
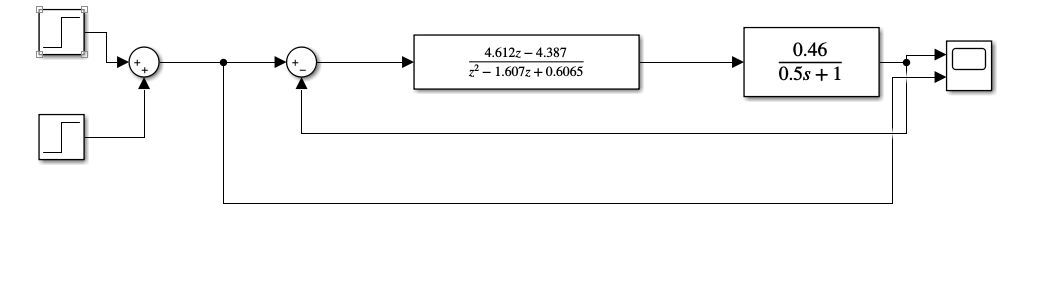
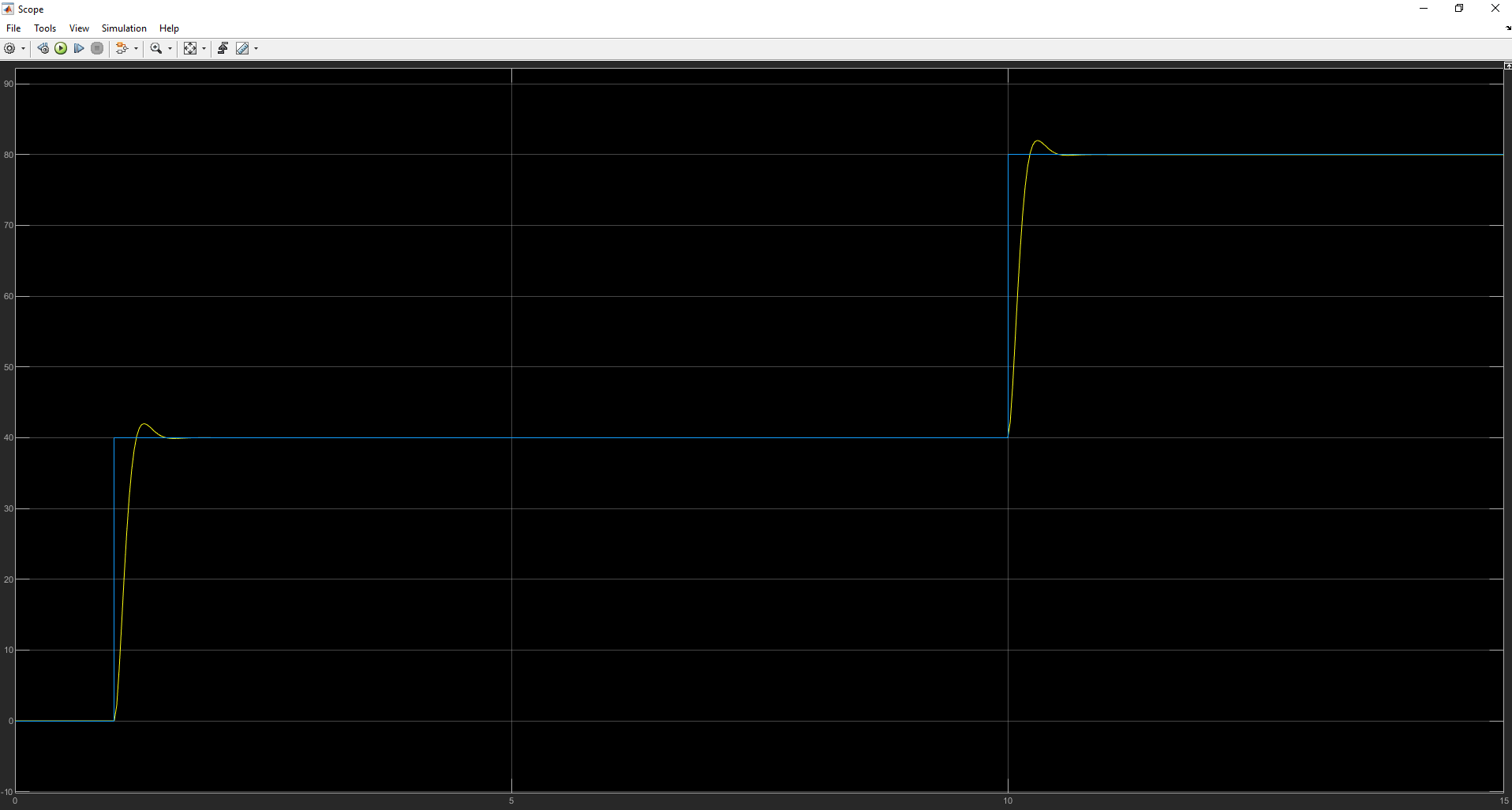
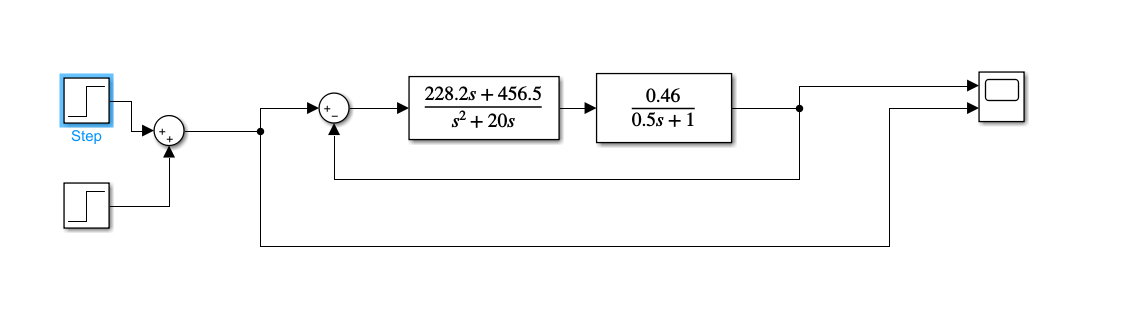
Simulink pentru testarea si verificarea rezultatelor obtinute din Matalb.

Mediul de programare Arduino IDEA pentru implementarea conceptelor teoretice obtinute din cele doua softuri de modelare.

Am folosit limbajul C/C++ pentru implementarea controlerului pe un microprocesor.

# Testare și analiza rezultate

Pentru testarea regulatorului proiectat am folosit Simulink



Am observat, ca in timpul miscarii masina isi mentine viteza la referinta pe care i-am impus-o.

# Concluzii

In teorie, regulatorul functioneaza in limitele asteptate, dar in practica rezultatele nu sunt chiar cele asteptate. Un motiv poate fi calitatea scazuta a componentelor hardware, pozitionarea componentelor pe setup, perturbatiile exterioare.

**Bibliografie**:

<https://www.onsemi.com/pdf/datasheet/p2n2222a-d.pdf> -functionare tranzistor.

[**https://create.arduino.cc/projecthub/onedeadmatch/control-dc-motor-with-npn-transistor-arduino-pwm-cdaf2e**](https://create.arduino.cc/projecthub/onedeadmatch/control-dc-motor-with-npn-transistor-arduino-pwm-cdaf2e) **-**functionare proces.

[**https://busoniu.net/teaching/sysid2021/sysid21ro\_part2\_handout.pdf**](https://busoniu.net/teaching/sysid2021/sysid21ro_part2_handout.pdf) **-**modelare proces.

[**https://www.optimusdigital.ro/ro/robotica-kit-uri-de-roboti/141-kit-robot-4-motoare.html**](https://www.optimusdigital.ro/ro/robotica-kit-uri-de-roboti/141-kit-robot-4-motoare.html) **-**achizitionare masinuta (majoritatea pieselor care nu veneau cu masinuta, le-am luat de pe Optimusdigital)