



PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

# ROBOT CONTROL APPLICATIONS

## MBOT

Autor: Băra Bogdan Alin

Grupa: **30131**

**AN UNIVERSITAR: 2021-2022**

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

# Cuprins

1. Scopul Proiectului .....	3
a. Obiective .....	3
b. Specificații .....	3
2. Determinarea modelului matematic al sistemului .....	4
a. Analiza sistemului .....	4
b. Achiziție semnale .....	4
c. Identificare / Modelare analitică .....	4
3. Proiectarea sistemului de control.....	5
4. Implementarea sistemului de control.....	9
5. Testare și analiza rezultate .....	14
6. Concluzii .....	15

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

# 1. Scopul Proiectului

Scopul proiectului consta în controlarea robotului MBOT. Conform teoriei controlului, MBOT este un sistem instabil pentru care va fi necesara realizarea si implementarea unui regulator. Astfel robotul va identifica și va evita obstacolele aflate pe traiectoria sa.

## a. Obiective

Obiectivele acestui proiect sunt :

- identificarea functiei de transfer al sistemului de ordin I
- determinarea unui regulator folosind metoda modulului
- trecerea sistemului din continuu în discret
- identificarea în buclă închisă a sistemului
- simularea sistemului determinat in Matlab
- testarea practică

## b. Specificații

Robotul MBOT va fi programat să meargă pe direcția înainte atâta timp cât nu întâlnește nici un obstacol, iar la întâlnirea acestuia să facă o rotație la dreapta, până la ocolirea acestuia. Performanțele impuse, prin metoda folosită sunt eroarea staționară la poziție 0, suprareglajul de 4.3% și marginea de fază 63.6°. În momentul în care senzorii vorsesiza apariția unui obstacol la o distanță de 20 de cm, robotul va face o întoarcere cu un unghi de 20° și își va continua traiectoria, tot pe directia inainte.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

## 2. Determinarea modelului matematic al sistemului

### a. Analiza sistemului

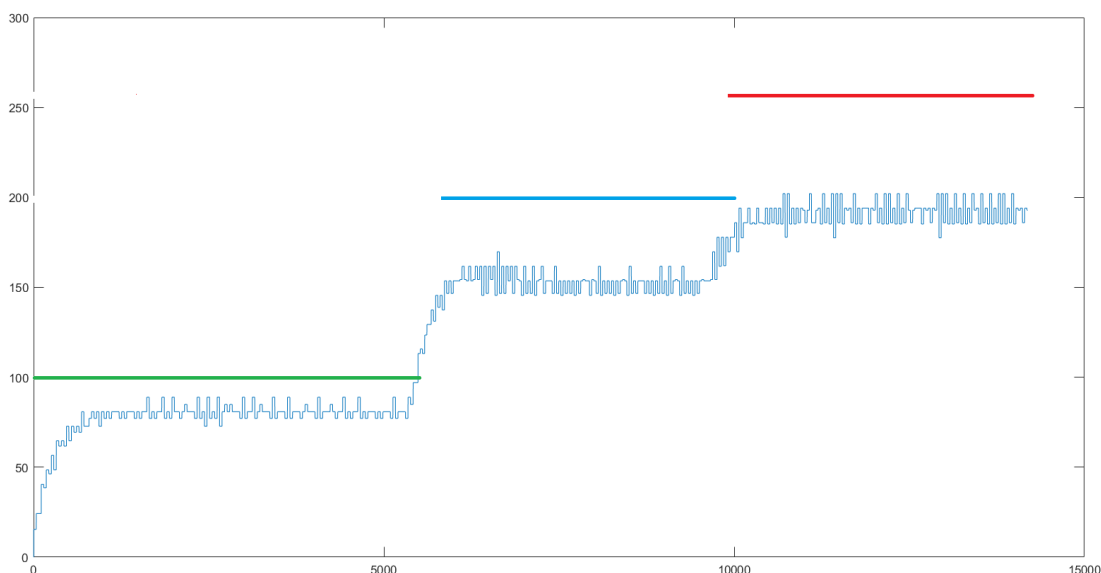
Conform teoriei controlului, MBOT este un sistem instabil de ordinul 1 la care am preluat tensiunea la intrare (marimea controlata) pentru modificarea pozitiei (marimea mainpulata).

### b. Achizitie semnale

-Pentru achizitia semnalelor m-am conectat la MBOT și am dat diferite pe rând 3 turații diferite motorului, 100, 200 și 255. După preluarea acestor valori, le-am introdus în matlab, am folosit comanda plot pentru a le afișa.

### c. Identificare / Modelare analitica

Pentru identificarea sistemului am utilizat formulele  $k = \frac{y_{ss} - y_0}{u_{ss} - u_0}$ ,  $T = T_{63,4\%} - T_{in}$  de unde a rezultat functia de transfer  $Hf(s) = \frac{0.7276}{0.306*s+1}$



PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

### 3. Proiectarea sistemului de control

Pentru că valorile din setul de performanțe să fie cele propuse va fi necesară proiectarea unui controler folosind o metodă adecvată dintre cele studiate anterior. Am utilizat metoda modulului.

#### Pași obținere regulator PID prin metoda modulului

1. Pornind de la aproximarea funcției de transfer (9.2) vom impune funcția de transfer în buclă directă de forma:

$$H_d^*(s) = \frac{1}{2 \times T_z \times s \times (T_z \times s + 1)} \quad (10.4)$$

2. Obținem astfel regulatorul din ecuația:

$$H_c(s) = \frac{H_d^*(s)}{H_f(s)} \quad (10.5)$$

3. Regulatorul proiectat prin metoda modulului asigură pentru structura de reglare următoarele performanțe:

$$\left\{ \begin{array}{l} \varepsilon_{stp} = 0 \\ \varepsilon_{stv} = 2 \times T_z \\ c_v = \frac{1}{\varepsilon_{stv}} = \frac{1}{2 \times T_z} \\ \omega_n = \frac{1}{\sqrt{2} \times T_z} \\ \zeta = \frac{1}{\sqrt{2}} \\ t_s = 8 \times T_z \\ \sigma = 4.3\% \\ PM = 63.6^\circ \end{array} \right. \quad (10.6)$$

4. Verificăm dacă performanțele obținute respectă setul inițial de performanțe impus.

22

$$y_0 = 80,85 \quad y_{99} = 153,61$$

$$u_0 = 100 \quad u_{99} = 200$$

$$K = \frac{y_{99} - y_0}{u_{99} - u_0} = \frac{153,61 - 80,85}{200 - 100} = \frac{72,76}{100} = 0,7276$$

$$T_{65\%} = 5,614 - 5,308 = 0,306$$

$$T_{63\%} = 5,614$$

$$T_{im} = 5,308$$

$$T = T_{63\%} - T_{im} = 0,306$$

$$T_e = 0,001$$

$$H_f(s) = \frac{0,7276}{0,306s + 1}$$

$$T_z = 0,306$$

$$H_d = \frac{1}{2T_z s (T_z s + 1)} = \frac{1}{0,612 s (0,306 s + 1)}$$

$$H_R = \frac{H_d}{H_f} = \frac{\frac{1}{0,612 s (0,306 s + 1)}}{\frac{0,7276}{0,306 s + 1}} = \frac{0,306 s + 1}{0,4452912 s (0,306 s + 1)} =$$

$$= \frac{0,306 s + 1}{0,1362591072 s^2 + 0,4452912 s}$$

$$H_{Rz} = \frac{0,002256 z - 0,002238}{z^2 - 1,9972709967} = \frac{0,002246 z^{-1} - 0,002238 z^{-2}}{1 - 1,997 z^{-1} + 0,9967 z^{-2}}$$

$$= \frac{C(z^{-1})}{E(z^{-1})}$$

PRO  
NU

$$0 = C_k - 1,997 C_{k-1} + 0,9967 C_{k-2} + 0,002246 E_{k-1} - 0,002238 E_{k-2}$$

$$C_k = 1,997 C_{k-1} - 0,9967 C_{k-2} - 0,002246 E_{k-1} + 0,002238 E_{k-2}$$

Regulatorul in discret:

$$Hr_z = \frac{0,002246 * z^{-1} - 0,002238 * z^{-2}}{1 - 1,997 * z^{-1} + 0,9967 * z^{-2}}$$

Performantele:

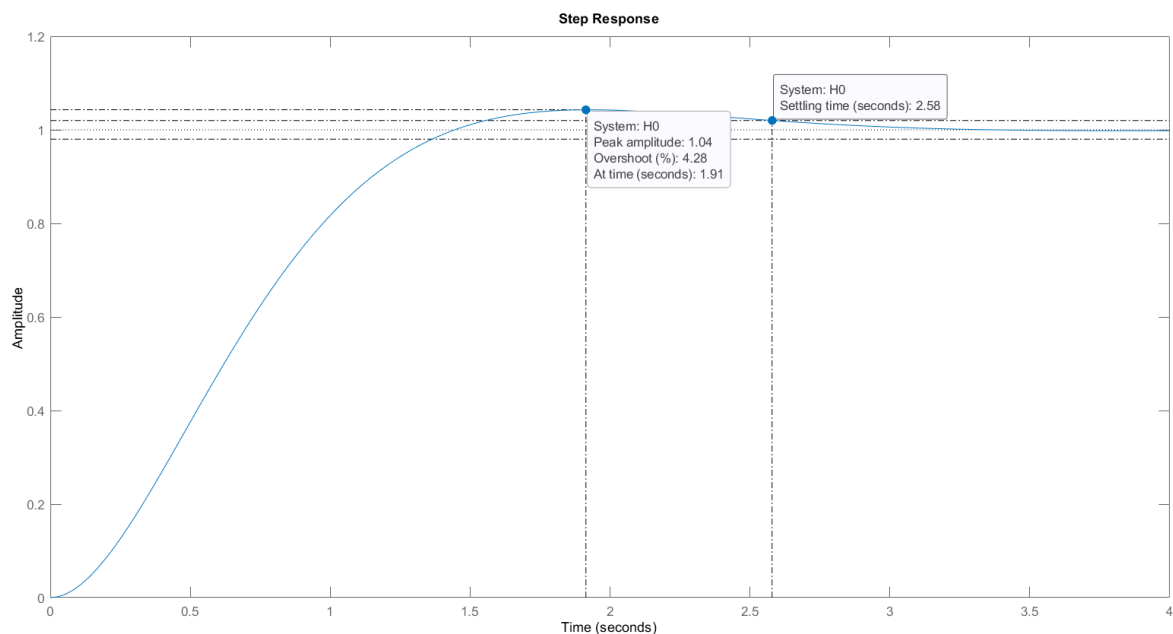
$\epsilon_{stp}=0$  (eroarea stationara la pozitie)

$tr=2.58$  (timpul de raspuns)

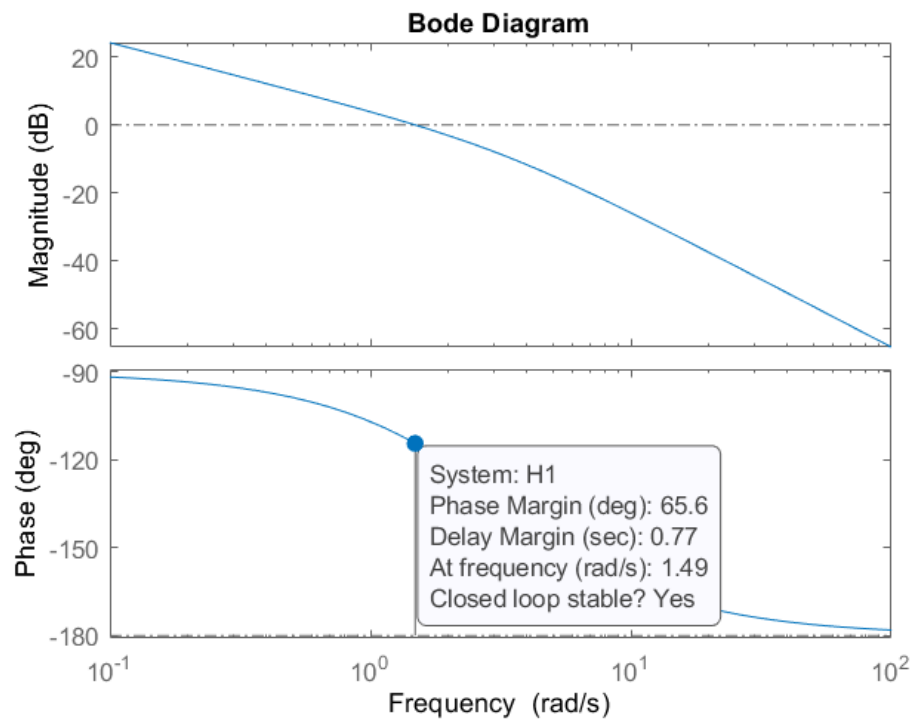
$\sigma=4.28\%$  (suprareglajul)

$PM=65.6^\circ$

Am testat in Matlab aceste performante utilizand functia step.



PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	



Relatia de recurenta este urmatoarea :

$$c(k) = 1.997 * c(k-1) - 0.9967 * c(k-2) - 0.002246 * \varepsilon(k-1) + 0.002238 * \varepsilon(k-2)$$



PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

## 4. Implementarea sistemului de control

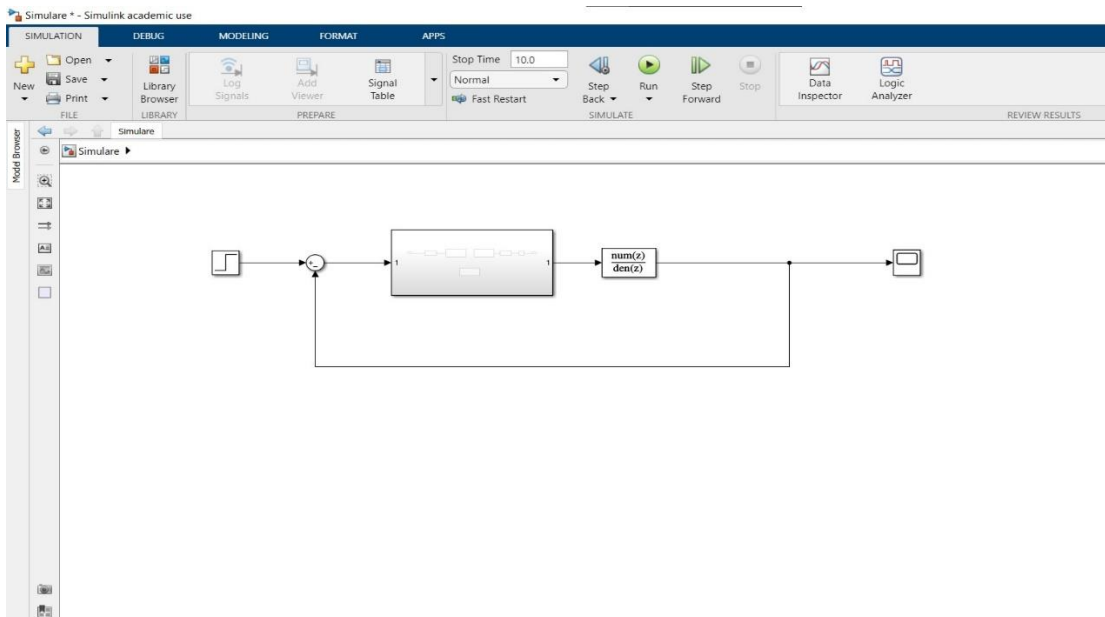
Descrieți detaliile tehnice ale implementării aplicației: mediul de implementare, modul de prezentare, modul de utilizare al aplicației, etc.

### Matlab

```
Hf=tf(0.7276,[0.306, 1]);

t=0:0.001:30;
u=t;
Hr=tf([0.306 1],[0.1362591072 0.445912 0]);
H1=series(Hf,Hr);
H0=feedback(H1,1);
Te=0.001;
Hfz=c2d(Hf,Te,'zoh');
Hrz=c2d(Hr,Te,'zoh');
minreal(Hrz);
figure;
step(H0);
figure;
bode(H1);
figure;
y=100*step(H0,t);
plot(t,y);
```

### Simulink



PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

### **Arduino(IDE)**

```
#include "MeAuriga.h"
```

```
#include <SoftwareSerial.h>
```

```
MeUltrasonicSensor ultraSensor(PORT_10); // Ultrasonic module
```

```
int distanceLimit=60; // Maximum distance we want to ping for (in centimeters).
```

```
int maxPwm=255;
```

```
int halfPwm=200;
```

```
int turnPwm=150;
```

```
int d=500;
```

```
int frontDistance;
```

```
//Motor Left
```

```
const int pwmMotor1 = 11;
```

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

```

const int inMotor1_1 = 47;
const int inMotor1_2 = 46;

//Motor Right
const int pwmMotor2 = 10;
const int inMotor2_1 = 49;
const int inMotor2_2 = 48;

void setup()
{
  Serial.begin(9600);

  pinMode(pwmMotor1,OUTPUT); //We have to set PWM pin as output
  pinMode(inMotor1_1,OUTPUT); //Logic pins are also set as output
  pinMode(inMotor1_2,OUTPUT);
  pinMode(pwmMotor2,OUTPUT); //We have to set PWM pin as output
  pinMode(inMotor2_1,OUTPUT); //Logic pins are also set as output
  pinMode(inMotor2_2,OUTPUT);

}

void loop()
{
  GetFrontDistance(); //Get the ultrasonic distance in cm

  if(frontDistance>=distanceLimit){
    FullSpeedMode(); //The robot is in full speed mode
  }

  else {
    ReduceSpeed(); //Reduce the speed of the robot
  }
}

```

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

```

    Stop();    //Stop the robot
  do {
    TurnRight();
    delay(100); // Wait for sensor to stabilize
    GetFrontDistance(); //Get the ultrasonic distance in cm
  } while(frontDistance<distanceLimit);
}
}

```

```

int GetFrontDistance(){
  frontDistance=ultraSensor.distanceCm();
  Serial.print("Front distance (in cm) is: "); //For debugging
  Serial.println(frontDistance);
  return frontDistance;
}

```

```

void FullSpeedMode(){
  digitalWrite(inMotor1_1, HIGH);
  digitalWrite(inMotor1_2, LOW);
  analogWrite(pwmMotor1,maxPwm);
  digitalWrite(inMotor2_1, LOW);
  digitalWrite(inMotor2_2, HIGH);
  analogWrite(pwmMotor2,maxPwm);
  Serial.println("Full speed mode"); //For debugging
}

```

```

void ReduceSpeed(){
  digitalWrite(inMotor1_1, HIGH);
  digitalWrite(inMotor1_2, LOW);
  analogWrite(pwmMotor1,halfPwm); //Set speed via PWM
  digitalWrite(inMotor2_1, LOW);
}

```

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

```

digitalWrite(inMotor2_2, HIGH);
analogWrite(pwmMotor2, halfPwm); //Set speed via PWM
Serial.println("Reduce the speed"); //For debugging
delay(d);
}

```

```

void Stop(){
  analogWrite(pwmMotor1, 0);
  analogWrite(pwmMotor2, 0);
  Serial.println("Stop");
  delay(d);
}

```

```

void TurnRight(){
  digitalWrite(inMotor1_1, HIGH);
  digitalWrite(inMotor1_2, LOW);
  analogWrite(pwmMotor1, turnPwm); //Set speed via PWM
  digitalWrite(inMotor2_1, HIGH);
  digitalWrite(inMotor2_2, LOW);
  analogWrite(pwmMotor2, turnPwm); //Set speed via PWM
  Serial.println("Turn the robot right"); //For debugging
}

```

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

## 5. Testare și analiza rezultate

Descrieți metodologia de testare a aplicației și rezultatele

Includeți experimentele pe care le-ați realizat, analiza rezultatelor pe care le-ați obținut.

După indentificarea funcției de transfer, proiectarea regulatorului și discretizarea acestora, am realizat o simulare în simulink, pentru a verifica funcționarea și performanțele sistemului proiectat. Acestea erau în parametrii, după care am conectat laptopul personal prin cablu la MBOT Bird, m-am asigurat că sunt legat la COM-ul bun, am verificat codul din Arduino și am uploadat datele în MBOT.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Băra Bogdan Alin	GRUPA:	30131	Nota	

## 6. Concluzii

Regulatorul calculat a ajutat la îndeplinirea setului de performanțe impus.