

Модель і метод управління колісним роботом вздовж смуги на базі нечіткого контролера на базі платформи Arduino

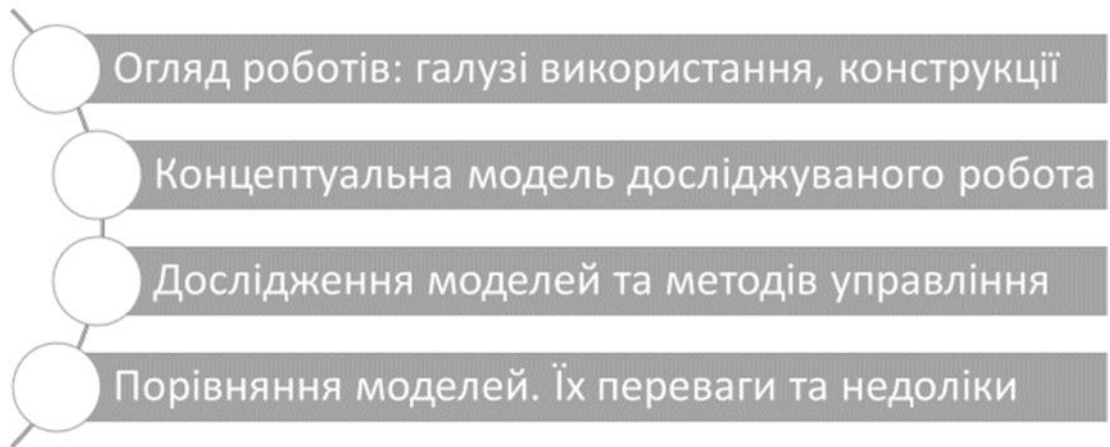
Студент Дриль М. М. СПм-18-3 | науковий керівник д.т.н., професор Каргін А.А.

2020

Мета роботи

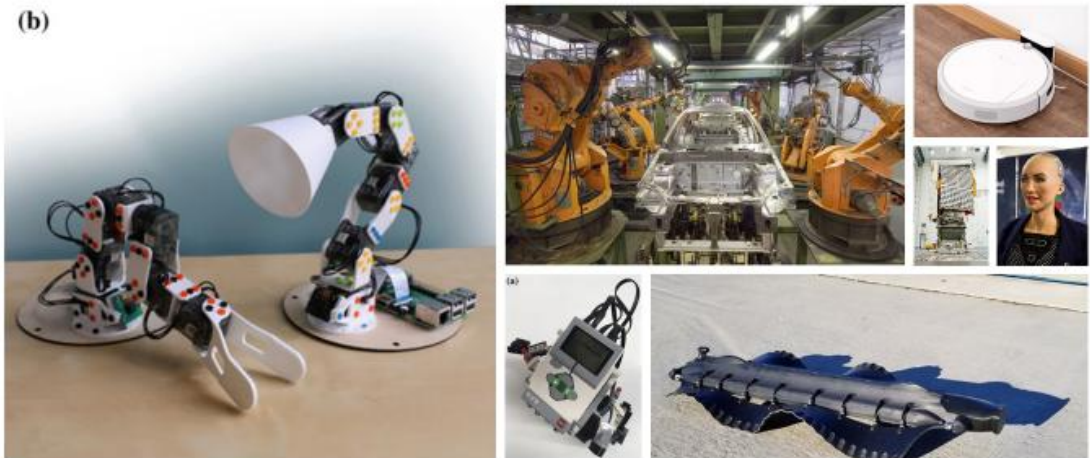
- Дослідити моделі та методи управління колісними роботами
- Порівняння моделей (їх переваги та недоліки)
- Виявлення кращої моделі управління

Хід роботи

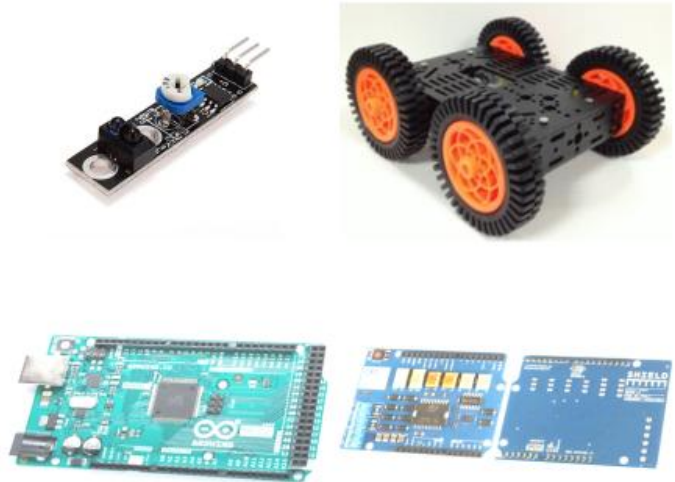


Огляд роботів: галузі використання, конструкції

(b)

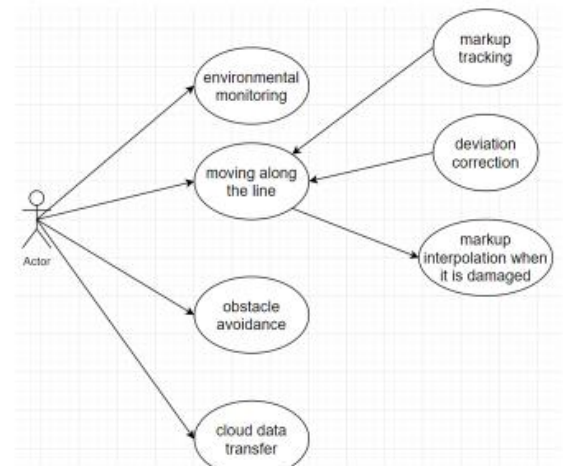


Фізична модель роботи

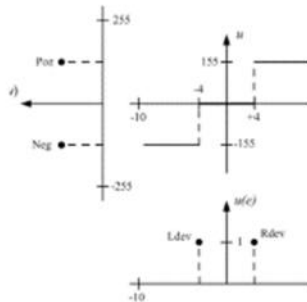


Концептуальна модель досліджувального робота

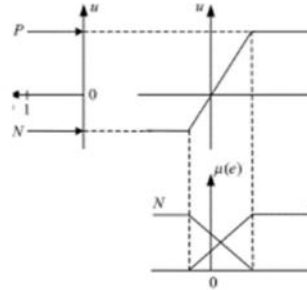
- environmental monitoring;
- moving along the line;
- obstacle avoidance;
- cloud data transfer.



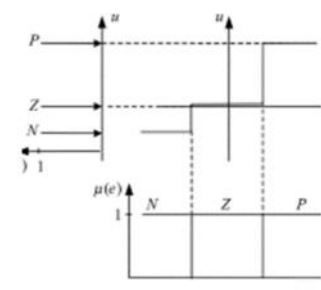
Дослідження моделей та методів управління



Нечітка двопозиційна модель контролеру з одноточковими функціями належності



Нечітка двопозиційна лінійна модель контролеру з насиченням

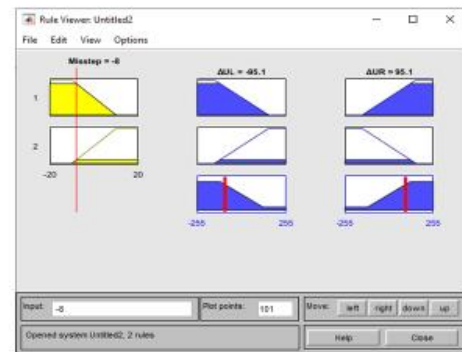
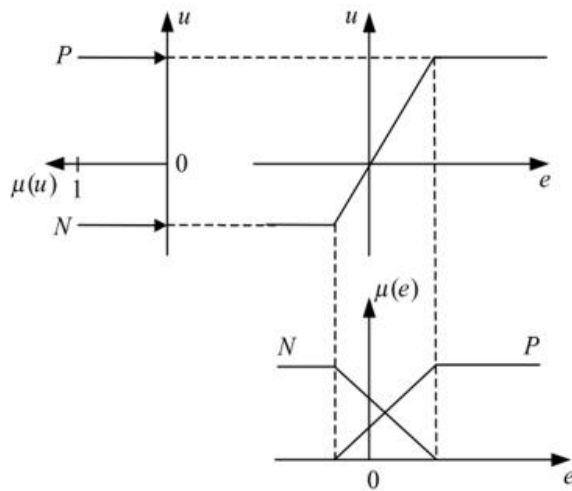


Нечітка трьохпозиційна лінійна модель контролеру з насиченням

Дослідження моделей та методів управління. Нечітка двопозиційна модель контролеру з одноточковими функціями належності

t	U_R^t	U_L^t	V^t	ω^t	S^t	α^t	e^t
0	125	125	0	0	0	0	-8
1	252	0	0	59,29	0	59,29	-8
2	255	0	0	60	0	60	-8
3	255	0	0	60	0	60	-8
4	255	0	0	60	0	60	-8
5	255	0	0	60	0	60	-8
6	255	0	0	60	0	60	-8
7	255	0	0	60	0	60	-8
8	255	0	0	60	0	60	-8
9	255	0	0	60	0	60	-8
10	255	0	0	60	0	60	-8
11	255	0	0	60	0	60	-8

Дослідження моделей та методів управління. Нечітка лінійна двопозиційна модель контролеру з насиченням

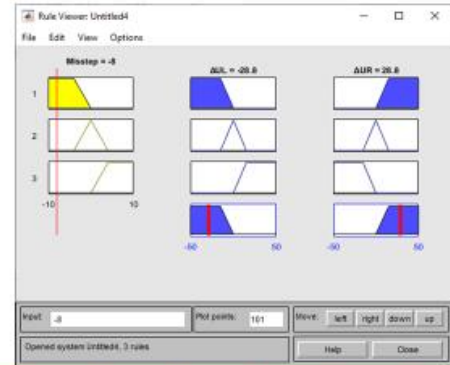
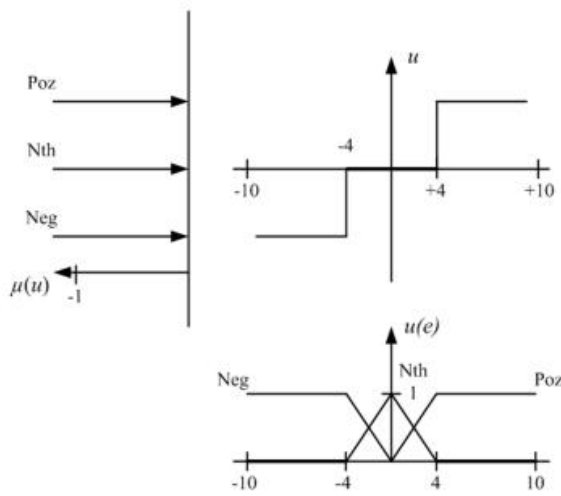


Misstep	-10	-8	-6	-4	-2	0
ΔU_R	113	95.1	73.9	50.5	25.6	0
ΔU_L	-113	-95.1	-73.9	-50.5	-25.6	0
Misstep	+10	+8	+6	+4	+2	0
ΔU_R	-113	-95.1	-73.9	-50.5	-25.6	0
ΔU_L	113	95.1	73.9	50.5	25.6	0

Дослідження моделей та методів управління. Нечітка лінійна двопозиційна модель контролеру з насиченням

t	U_R^t	U_L^t	V^t	ω^t	S^t	α^t	e^t
0	125	125	0	0	0	0	8
1	29	220,1	2,9	-44,8	2,9	-44,8	6
2	0	255	0	-60	0	-60	6
3	0	255	0	-60	0	-60	6
4	0	255	0	-60	0	-60	6
5	0	255	0	-60	0	-60	6
6	0	255	0	-60	0	-60	6
7	0	255	0	-60	0	-60	6
8	0	255	0	-60	0	-60	6
9	0	255	0	-60	0	-60	6
10	0	255	0	-60	0	-60	6
11	0	255	0	-60	0	-60	6

Дослідження моделей та методів управління. Нечітка трьохпозиційна лінійна модель контролеру з насиченням



Misstep	-10	-8	-6	-4	-2	0
ΔU_a	28,8	28,8	28,8	28,8	19,6	0
ΔU_L	-28,8	-28,8	-28,8	-28,8	-19,6	0
Misstep	+10	+8	+6	+4	+2	0
ΔU_a	-28,8	-28,8	-28,8	-28,8	-19,6	0
ΔU_L	28,8	28,8	28,8	28,8	19,6	0

Дослідження моделей та методів управління. Нечітка трьохпозиційна лінійна модель контролеру з насиченням

t	u_a^t	u_L^t	V^t	ω^t	S^t	α^t	e^t
0	125	125	0	0	0	0	-8
1	153,8	96,2	9,431	13,553	9,431	13,553	-6
2	182,6	67,4	6,608	27,106	6,608	27,106	-2
3	202,2	47,8	4,686	36,329	4,686	36,329	0
4	202,2	47,8	4,686	36,329	4,686	36,329	0
5	202,2	47,8	4,686	36,329	4,686	36,329	2
6	182,6	67,4	6,608	27,106	6,608	27,106	6
7	153,8	96,2	9,431	13,553	9,431	13,553	10
8	125	125	12,255	0,000	12,255	0,000	10
9	96,2	153,8	9,431	-13,553	9,431	-13,553	8
10	67,4	182,6	6,608	-27,106	6,608	-27,106	6
11	38,6	211,4	3,784	-40,659	3,784	-40,659	2

Висновок

Нечітка трьохпозиційна лінійна модель
контролеру з насиченням

Модель піддається вдосконаленню

Робот піддається вдосконаленню

```

#include<AFMotor.h>

#define kCoef 30
#define baseSpeed 200

#define Sensor_one 22
#define Sensor_two 23
#define Sensor_three 24
#define Sensor_four 25
#define Sensor_five 26
#define Sensor_six 27
#define Sensor_seven 28
#define Sensor_eight 29
#define Sensor_nine 30
#define Sensor_ten 31

byte baseSpeedMaxLeviy;
byte baseSpeedMaxPraviy;

AF_DCMotor motor1(1);
AF_DCMotor motor2(2);

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

void loop() {
  int value_one = digitalRead(Sensor_one) ;
  int value_two = digitalRead(Sensor_two);
  int value_three = digitalRead(Sensor_three);
  int value_four = digitalRead(Sensor_four);
  int value_five = digitalRead(Sensor_five);
  int value_six = digitalRead(Sensor_six);
  int value_seven = digitalRead(Sensor_seven);
  int value_eight = digitalRead(Sensor_eight);
  int value_nine = digitalRead(Sensor_nine);
  int value_ten = digitalRead(Sensor_ten);

  Serial.print(value_one);
  Serial.print("\t");
  Serial.print(value_two);
  Serial.print("\t");
  Serial.print(value_three);
  Serial.print("\t");
  Serial.print(value_four);
  Serial.print("\t");
  Serial.print(value_five);
  Serial.print("\t");
  Serial.print(value_six);
  Serial.print("\t");
  Serial.print(value_seven);
  Serial.print("\t");
  Serial.print(value_eight);
  Serial.print("\t");
  Serial.print(value_nine);

```

```

Serial.print("\t");
Serial.println(value_ten);
  byte Praviydelta12;
  byte Praviydelta34;
  byte maxPraviy;
  byte Leviydelta12;
  byte Leviydelta34;
  byte maxLeviy;

  byte Leviydelta1=(4* (value_one ^ value_two));
  byte Leviydelta2=(3* (value_two ^ value_three));
  byte Leviydelta3=(2* (value_three ^ value_four));
  byte Leviydelta4=(1* (value_four ^ value_five));
  byte Praviydelta1=(1* (value_six ^ value_seven));
  byte Praviydelta2=(2* (value_seven ^ value_eight));
  byte Praviydelta3=(3* (value_eight ^ value_nine));
  byte Praviydelta4=(4* (value_nine ^ value_ten));
if(Praviydelta1 < Praviydelta2 ){
  Praviydelta12=Praviydelta2;
}else{
  Praviydelta12=Praviydelta1;
}
if(Praviydelta3 < Praviydelta4 ){
  Praviydelta34=Praviydelta4;
}else{
  Praviydelta34=Praviydelta3;
}

if(Praviydelta12 < Praviydelta34 ){
  maxPraviy=Praviydelta34;
}else{
  maxPraviy=Praviydelta12;
}
if(Leviydelta1 < Leviydelta2 ){
  Leviydelta12=Leviydelta2;
}else{
  Leviydelta12=Leviydelta1;
}
if(Leviydelta3 < Leviydelta4
  Leviydelta34=Leviydelta4;
}else{
  Leviydelta34=Leviydelta3;
}

if(Leviydelta12 < Leviydelta34 ){
  maxLeviy=Leviydelta34;
} else {
  maxLeviy=Leviydelta12;};

If (kCoef <=-4) and (kCoef =>-10) then ((maxLevyi = -28.8) and (maxPraviy
=28.8));
If else (kCoef ==-2) then ((maxLevyi = -19.6) and (maxPraviy =19.6));
If else (kCoef =0) then ((maxLevyi =0) and (maxPraviy =0));
If else (kCoef =2) then ((maxLevyi = 19.6) and (maxPraviy ==-19.6));
If else (kCoef =>4) and (kCoef =>10) then ((maxLevyi = 28.8) and (maxPraviy
==28.8));

baseSpeedMaxLeviy = baseSpeed - (kCoef * maxLeviy);
baseSpeedMaxPraviy = baseSpeed - (kCoef * maxPraviy);

motor1.setSpeed( baseSpeedMaxLeviy);
motor2.setSpeed(baseSpeedMaxPraviy
motor1.run(FORWARD);
motor2.run(FORWARD);
delay(50);
}

```