

```

#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;};

int avgSpeed=150;
int kP=10;
int error;

error=bot_position();
correction= kP* error;

motor1.move(avgSpeed*(1+correction));
motor2.move(avgSpeed*(1-correction));
motor1.setSpeed( baseSpeedMaxLeviy);
motor2.setSpeed(baseSpeedMaxPraviy
motor1.run(FORWARD);
motor2.run(FORWARD);
delay(50);

```

В

Харківський національний університет радіоелектроніки  
Кафедра ЕОМ

## Модель і метод управління переміщенням колісного робота на базі ПІД контролеру на платформі Arduino

Атестаційна робота бакалавра

Другий (магістерський) рівень

Автор:  
Ласуков В.С  
Ст.гр. СПм-18-3

Керівник:  
Ляшенко О.С.  
Доц.каф. ЕОМ

Харків 2020

### Мета та задачі проекту

Метою атестаційної роботи є розробка і метод управління переміщеннями колісного робота вздовж стіни .

Для досягнення поставленої мети необхідно вирішити такі задачі :

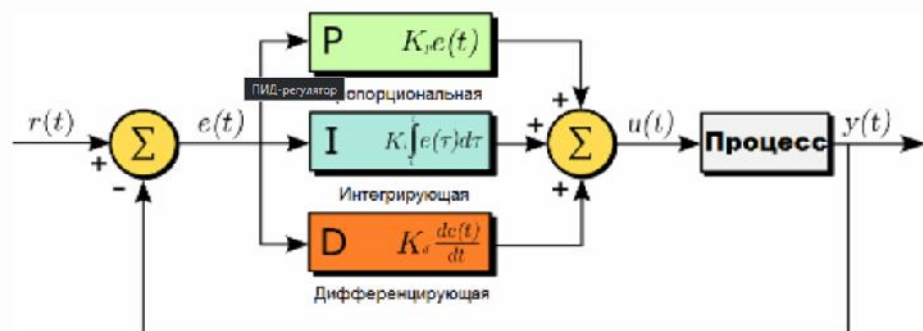
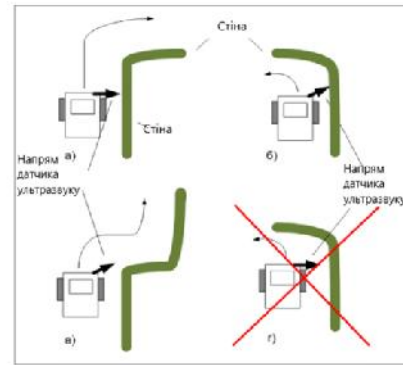
- Провести аналіз та класифікацію способів управління мобільних роботів ;
- розробити імітаційну модель руху мобільного робота вздовж заданого маршруту;
- дослідити та налаштувати ПІД алгоритм управління мобільним роботом ;
- розробити прототип системи управління мобільним роботом.



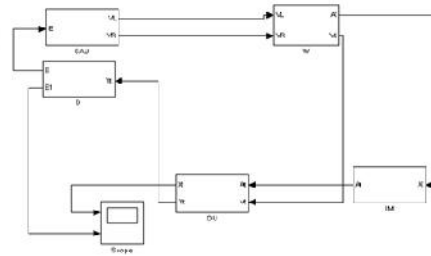
Будь-який мобільний робот може бути представлений у вигляді сукупності трьох основних систем - транспортної, спеціальної і системи управління. Транспортна система є транспортним засобом, призначеним для доставки спеціального і технологічного устаткування до місця виконання поставленого завдання. Спеціальні системи. Спеціальні системи служать для безпосереднього виконання поставлених завдань. Система управління забезпечує управління рухом і роботою технологічного устаткування робота.



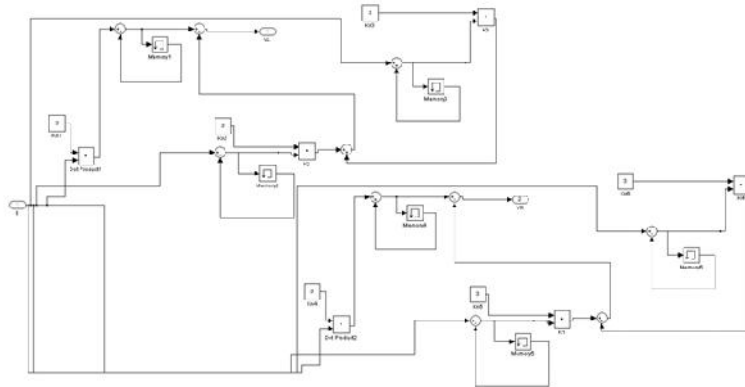
Схема розміщення робота і датчика  
ультразвуку в залежності від типу  
повороту



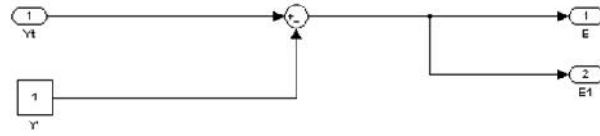
Імітаційна модель нараховує в собі 5 блоків:  
 SAU – система автоматичного управління  
 W – перерахунок оборотів  
 IM – виконуючий механізм  
 OU – об'єкт управління  
 D – датчик



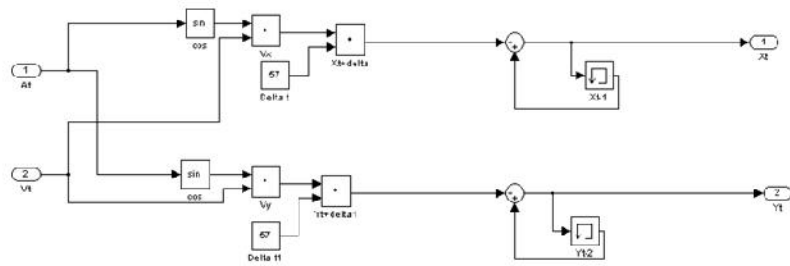
### Блок SAU



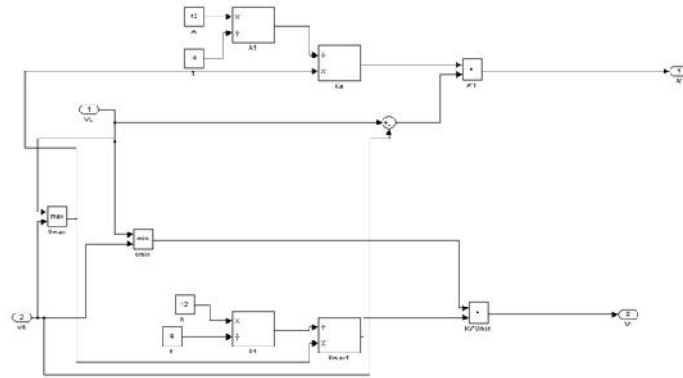
### Блок D



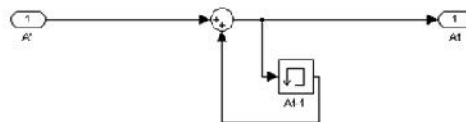
### Блок OU

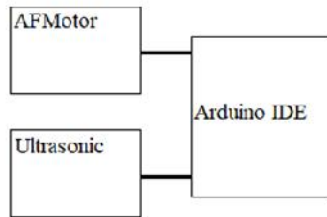
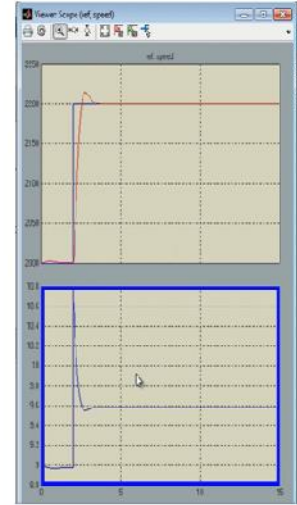
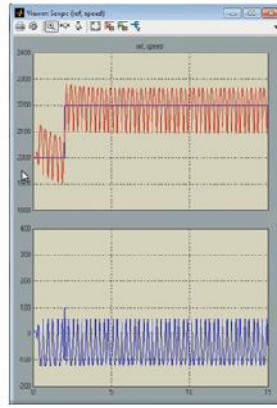
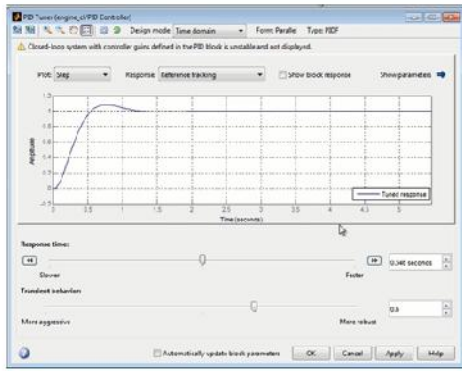
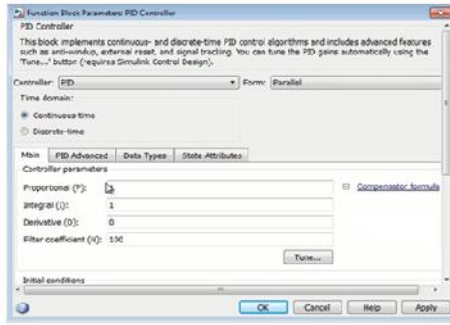


Блок W



Блок IM





```

sketch_dec06e | Arduino 1.8.7 (Windows Store 1.8.13.0)
Файл Правка Скетч Инструменты Помощь

sketch_dec06e
#include <AFMotor.h>

AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

void setup() {
  motor1.setSpeed(200);
  motor2.setSpeed(200);
  motor3.setSpeed(200);
  motor4.setSpeed(200);

  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);

  delay(4000);

  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  
```

## Висновки

Розробка інтелектуальних мобільних роботів для різних виробничих і дослідницьких цілей є дуже важливим і актуальним завданням. В даний час виконано величезну кількість досліджень, пов'язаних з розробкою алгоритмів управління, які забезпечують вирішення за допомогою мобільних роботів таких нетривіальних операцій, як: складання карти місцевості, її уточнення, планування траєкторій, обхід перешкод, що виявляються при русі, проникнення в важкодоступні зони і т. д. Подальші дослідження нових типів мобільних роботів стимулюється численними додатками в самих різних областях людської діяльності (автоматизація управління рухом транспортних засобів, боротьба з тероризмом і розмінування підозрілих предметів, робота в умовах сильної задимленості під час пожежогасіння, інспекція територій, забруднених хімічними речовинами, самостійне патрулювання призначених територій та ін.).