GOVERNMENT COLLEGE OF ENGINEERING

YAVATMAL

MINI-PROJECT REPORT

ON

WALKING ROBOT

SUBMITTED BY

1. TUSHAR GAIKWAD (2110121372533)
2. AFTAB SHAIKH (2110121372527)
3. RUTUJA DOMB (2110121372532)
4. PIYUSH MESHRAM (2110121372535)

GUIDED BY

PROF. YOGESH CHAUDHARI

DEPARTMENT OF ELECTRONICS

AND TELECOMMUNICATION ENGINEERING

(2022-2023)

GOVERNMENT COLLEGE OF ENGINEERING

YAVATMAL

<u>CERTIFICATE</u>

This is clarify that of 3rd Year Electronics and Telecommunication students have submitted their Mini project report on "**WALKING ROBOT**" during academic session 2022- 2023 as a part of project work prescribed by Government College Of Engineering Yavatmal for partial fulfilment for the Degree in Electronics and Telecommunication.

The Project work is the record of students own work and is completed satisfactorily.

PROF. YOGESH CHAUDHARI	PROF. SAMRAT S. THORAT
GUIDED BY	**HEAD OF DEPARTMENT**

ACKNOWLEDGEMENT

We would like to place on record my deep sense of gratitude to our Guide, **Prof. Yogesh Chaudhari,** Department of Electronics and Telecommunication for his generous guidance, help and useful suggestions.

We express my sincere gratitude to **Prof. Samrat S. Thorat,** Head of Department of Electronics and Telecommunication, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We are extremely thankful to Principle **Prof. (Dr.) P M Khodke** , for providing us infrastructural facilities to work in, without which this work would not have been possible.

INDEX

ABSTRACT

This work deals with the four legged walking robot with obstacle overcoming capa- bilities. It was designed as a research robot platform, to be used in indoor environ- ment, walking straight and curved paths and detecting and overcoming known obsta-cles. Straight paths are done with gait matrix strategy and curved paths are accom- plished by the four legs which have differential strokes in inclined paths relatively to the robot longitudinal axis. Obstacle overcoming is done using only information fromcontact sensors installed on the robot's feet. Complex movements and tracking se- quences are proposed to be built from a small group of simple movements sequenced according to the contact keys switching sequence. Results include: gait generation for straight and curved paths ; integrated movement sequence to go one step up in the floor; and a description of the development, construction and testing of quadruped robot. This paper designs a four legged parallel walking robot to go through a narrow hole. Topol-ogy design is conducted for a leg mechanism composed of four legs, base and ground, which constitute a redundant parallel mechanism. This mechanism is subdivided intofour sub-mechanism composed of three legs. A motor vector is adopted to determine the 6x8 Jacobian of the redundant parallel mechanism and the 6x6 Jacobian of the sub- mechanisms, respectively. The condition number of the Jacobian matrix is used as an index to measure a dexterity. We analyze the condition numbers of the Jacobian over the positional and orientational walking space. The analysis shows that a sub-mechanismhas lots of singularities within workspace but they are removed by a redundant parallelmechanism improving a dexterity. From the results, we can propose a parallel typed walking robot to enlarge walking space and stability region. The robot is designed byinserting an intermediate mechanism between upper and lower leg mechanisms. The robot is reasonably small so that it can go through a narrow hole.

CHAPTER 1

1.1 Introduction

Researchers have proposed a variety of walking robots such as a multi-legged type like an insect (Garcia et al., 2006) and a biped type like a human being(Huang et al., 2001). The legged type robot requires lots of degrees of freedom with heavy weight. A Delta robot (Reg Dunlop, 2003) and a serial-parallel hybrid robot (Yusuke et al., 2001) are used for a walking robot to reduce the degrees of freedom and the weight. They consist of two leg-bases with three fixed legs each, and walk by moving each leg-base alternately. They can control the position and orientation of the bases to walk rough terrain as well as turn omni-directionally. However, the robots suffer from the lack of mobility and terrain adaptability since its walking space is limited due to unactuated legs.

This research designs the Parallel Walking Robot (PWR) using the advantage of parallel mechanism such as dexterity and stability. Actuated legs are installed on the base, which produces the position and orientation arbitrarily. The robot can walk on the irregular surface of ground with small degree of freedom and has a large stability since the legs support the base in a wide space. Also parallel mechanism stacked in two layers with medium mechanism can generate a wide range of walking space to go through a narrow hole and climb up a wall. Therefore the proposed robot will be applied to painting and cleaning of in-and-out of a block for ship construction. This research proposes three layered parallel walking robot and designs its leg mechanism with dexterity analysis. To enhance the dexterity, an extra leg is added to leg mechanism, which yields a redundant parallel mechanism composed of four legs, base and ground. To analyze the roles of the extra leg, we subdivide the four legged mechanism into four sub mechanisms and obtain their Jacobians. From the Jacobians' condition number, we present that all the singularities of sub mechanism are eliminated by adding the extra leg and the dexterity is improved by the redundant mechanism We design a robot which is able to walk stably in various environments, such as on rough terrain and up/down slopes.Also the robot should be small and light to go through a narrow hole and climb up a wall for painting or cleaning in the process of ship construction.

For this purpose, we propose a parallel walking robot composed of three layered paral-lel mechanisms; an Upper Legged Mechanism (ULM) and a Lower Legged Mechanism (LLM) are attached to the upper base and the lower base, respectively, and Intermedi-ate Mechanism (IM) is inserted into between the bases. When the legs' feet step the ground to support a base, the base, support legs and ground form a parallel mecha- nism so that the legged mechanism generates the position and orientation of the base arbitrarily. When the legs' feet lift from the ground, they can swing for the next step. Therefore, the robot can walk rough terrain and up/down slopes with a large walking space and avoid the obstacle easily. Also, ULM, IM and LLM are independently oper-ated, which are combined to generate a complex walking space which can't be made bya conventional walking robot.

1.2 Literature Survey

YEAR	Author	Title of paper	Publication	Claim by author
2018	M.M.GOR, PM PATLAK,K ALAM, R SARKAR, D ANAND, J.M YANG	Development of a compliant legend quadruped robot	Indian academic of science	Experimental result of robot walking with amble gaits are demand rates, There are certain issue like body disturbance during dynamic walk and different type of joint failures.
2016	Patrick M Wensing , Albert wang, Sangok Seok, David Otten	Impact mitigation and high bandwidth physical interaction for Dynamic legend walking robot	Massachusetts Institute of technology	That paper proposed a new actuator paradigm for high speed robot, provided analysis to demonstrate the central tenants of paradigm And presented experiment result of force tracking tests.
2010	Manuel Fernando silra and JA Tenreiro Manchado	Optimization of legged robot	Scientific Repository the polytechnic institute of Porto	In this paper has presented a survey of several strategies, namely the mechatronic mimic of biological animal characteristics, the uses of evolutionary computation for optimization of the legend structure parameter. The adoption of good mechanical project rules, the optimization of power and energy .
2020	Priyaranjan Biswal, prases k mohantey	Development of quadruped walking robot	Ain shams Engineering journal	Previous actuators in quadrupedal robot is hydraulic and electrical but in this we use legs for less size of robot as over proportionally longs are mostly Strickly each other and high proportionally body structure requires more balance with less agile.

CHAPTER 2

Component

2.1 Arduino NANO

The Arduino NANO is a microcontroller board based on a removable, dual-inline- package (DIP) ATmega328AVR microcontroller.It is a board based microcontroller (small computer on a chip) with facilities for processing data and I/O (input and output) pins for receiving and sending signals to devices. It can be powered and programmedusing a computer or mobile phone.

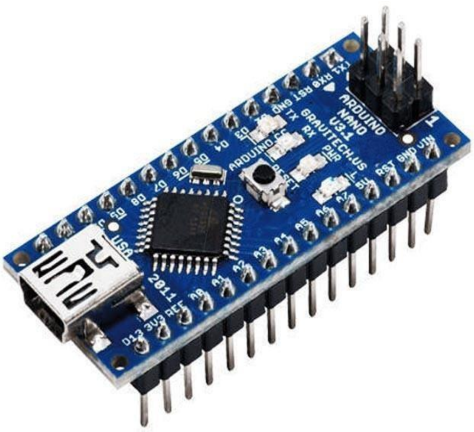


Fig 2.1. Arduino NANO

Features:

ATmega328P Microcontroller is from 8-bit AVR

familyOperating voltage is 5V

Input voltage (Vin) is 7V to

12VInput/Output Pins are 22

Analog i/p pins are 6 from A0 to A5

Digital pins are 14

Power consumption is 19

mAI/O pins DC Current is

40 mA

Flash memory is 32

KBSRAM is 2 KB

EEPROM is 1 KB

CLK speed is 16

MHzWeight-7g

Size of the printed circuit board is 18 X 45mm

Supports three communications like SPI, IIC,

USART

2.2 Servo Motors

Servo motors or “servos”, as they are known, are electronic devices and rotary or linearactuators that rotate and push parts of a machine with precision. Servos are mainly used on angular or linear position and for specific velocity, and acceleration. Companies heavily use servo motors because of how compact and potent it is. Despite its size, it generates quite the amount of power and is known to be incredibly energy-efficient.

Fig 2.2. Servo Motors

Connectors, Power, Tools Accessories Elements :

Jumper/ Connecting cables

Male-to-Female:

Single / Group of Electric wires with connector or pin at each end. It is used to inter-connect the components. Female ends are used to plug things

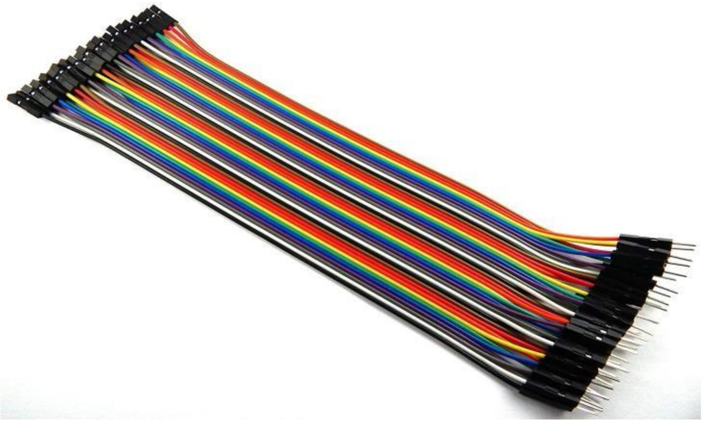


Fig 2.3. Male-to-Female

Male-to-Male:

The difference between these wires is in the endpoint of the wire. Maleends have a pinprotruding and can plug into things.

Fig 2.4. Male to Male

2.3 Battery packet / Power supply

It is a simple case with terminals, for holding the batteries.

Fig 2.5. Battery Case

2.4 lithium battery

A lithium-ion (Li-ion) battery is an advanced battery technology that uses lithium ionsas a key component of its electrochemistry.

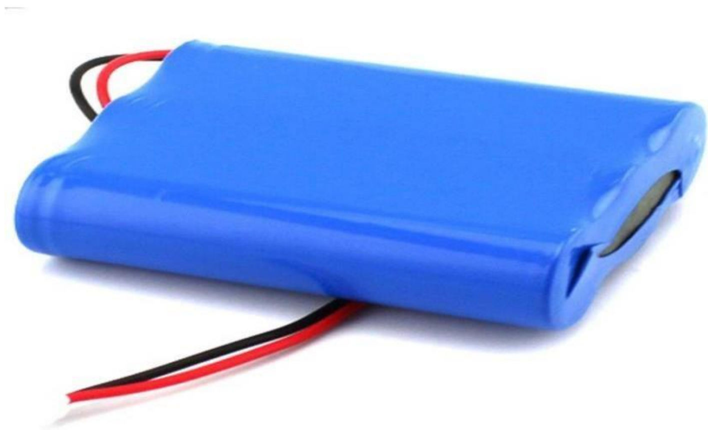


Fig 2.6. lithium battery

2.5 Ultra Sonicsensor

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ul- trasonic pulses that relay back information about an object's proximity. High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

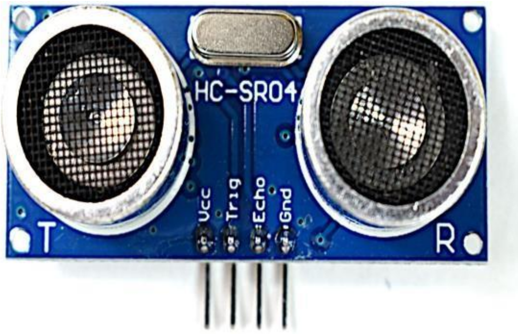


Fig 2.7. Ultra Sonicsensor

Features-

• Power Supply :+5V DC

• Quiescent Current : <2mA

• Working Current: 15mA

• Effectual Angle: <15°

• Ranging Distance : 2cm – 400 cm/1 – 13ft

• Resolution : 0.3 cm

• Measuring Angle: 30 degree

• Trigger Input Pulse width: 10uS TTL pulse

• Echo Output Signal: TTL pulse proportional to the distance range

• Dimension: 45mm x 20mm x 15mm

CIRCUIT DIAGRAM

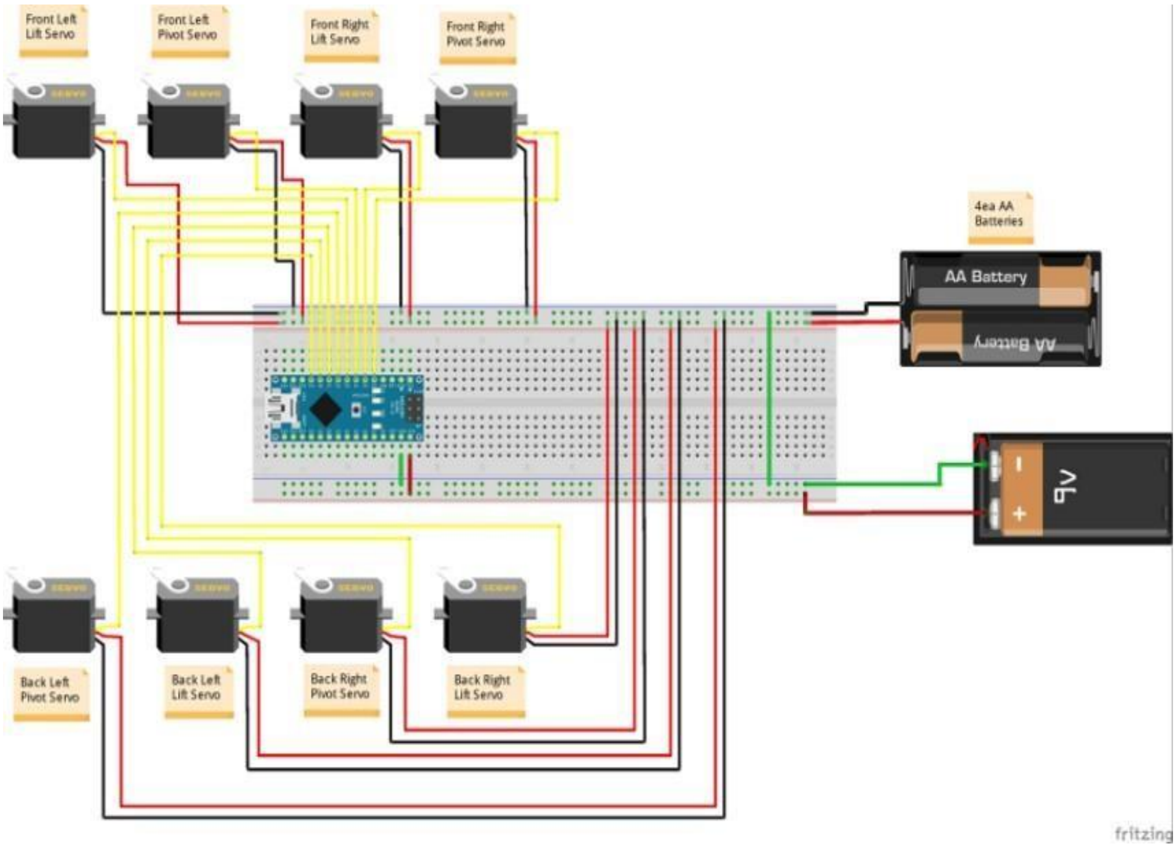


Fig 3.1. Cicuit Diagram

COMPONENT REQUIREMENT-

- Arduino Nano
- Servo motors
- Chassis for the robot (3D printed or handmade)
- Jumper wire
- Ultrasonic sensor
- Battery pack / Power supply

CHAPTER 4

Working And Design

The robot consists of four legs, each with three joints that can move in multiple direc-tions. The joints are connected to servos, which are controlled by the Arduino. The servos move the joints in a coordinated manner to allow the robot to walk. The Arduino is programmed to control the servos using a simple algorithm that calculates the anglesfor each joint based on the desired movement of the robot. The algorithm takes into account the weight of the robot and the terrain it is walking on to ensure stable move-ment. The robot is powered by a battery and can be controlled remotely using a wireless module, such as Bluetooth or Wi-Fi. This allows the user to control the movement ofthe robot from a distance. Overall, building a four legged walking robot using Arduino is a fun and educational project that allows enthusiasts to learn about robotics and pro-gramming.

The walking robot can be easily operated with Mobile Application. Working of the robot is as follows-

- When an operator commands the robot to perform a specific task via an android app.
- The command then travels to the Arduino Nano via Bluetooth module in the form of a bit number.
- Arduino Nano receives this bit number and executes the command programmed for that number like walk, turn etc.
- Execution of the command results in movement of the servo motors in a specific pattern to perform a move.

• Sensors can sense the surrounding temperature and any obstacles.

• This data is sent to the user via Bluetooth module.

• The app then displays the surrounding condition to the user.

• In the mean time the Camera display is continuously providing the audio-visual to the operator.

Task performing using robotic arm

- A robotic arm has links of kinematic chain mechanism which is same as that of mechanical arm which can be programmed to perform similar functions to a human arm.
- The arm is the sum of total of the manipulators connections. Depending on the program any designed task to perform any desired activity in dangers of treating suspicious objects in their current world and workplace such as welding, grasping turning etc. would make it possible for humans to avoid dangers
- The project in Tamil Nadu intends to make it easier for humans to avoid dangers of treating suspicious items in their current atmosphere and workplace by making complex and complicate duties faster and accurately with this design.
- The key benefit of this robot is its gentle grabbing motion; this technology is designed to prevent applying extra pressure to the suspicious target for safety purposes and to avoid the dangers of treating suspicious objects in their current world and workplace

METHODOLOGY

The Methodology of the project included the following steps

Step 01: Finding the issue in the Surveillance system

Step 02: Reviewing the existing literature

Step 03: Defining problem statement

Step 04: Selection of sensor, motor controller etc.

Step 05: Design of the body

Step 06: Design of robotic arm and effectors

Step 07: Design Calculation

Step 08: Design Assembly

Step 09: Writing source code for robot ad controller

Step 10: Checking for optimality and if YES the moving ahead, if NO the going for redesign from step 5.

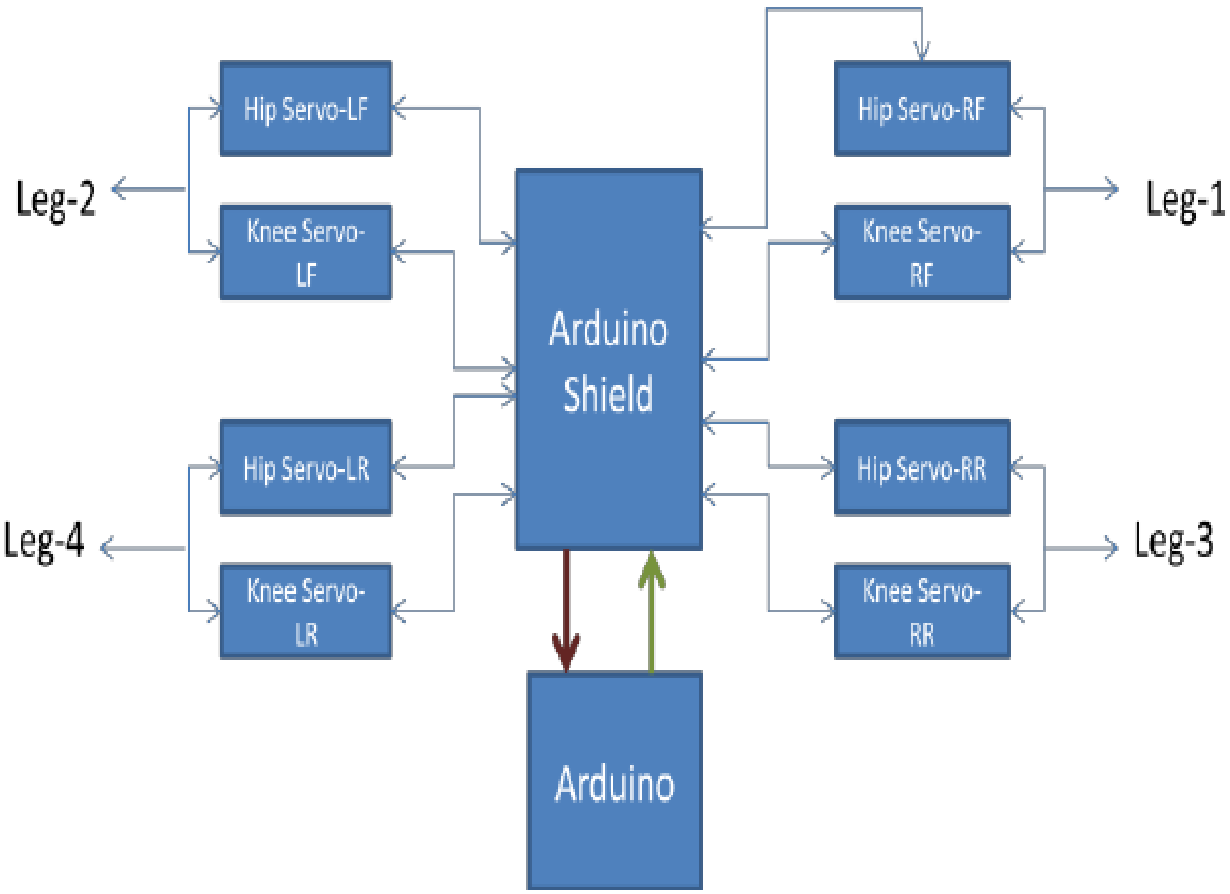
Step 11: Fabrication of the robot

Step 12: Solving the issues encountered

Application

1. They benefit from increased stability over bipedal robots, especially during move-ment
2. At slow speeds, a quadrupedal robot may move only one leg at a time, ensuring astable tripod.
3. Four-legged robots also benefit from a lower center of gravity than two-legged sys-tems.

Flow Chart



Hip Servo-LF
Knee Servo-LF
Leg-2
Hip Servo-LR
Knee Servo-LR
Leg-4
Arduino Shield
Hip Servo-RF
Knee Servo-RF
Leg-1
Hip Servo-RR
Knee Servo-RR
Leg-3
Arduino

Code

```
#include <Servo.h>

//
Servo joint2;  //
Servo joint3;  //
Servo joint4;  //
Servo joint5;  //
Servo joint6;  //
Servo joint7;  //
Servo joint8;  //
Servo joint9;  //
Servo neck_servo;  //

//
int home_joint2 = 120;
int home_joint3 = 15;
int home_joint4 = 60;
int home_joint5 = 165;
int home_joint6 = 100;
int home_joint7 = 15;
int home_joint8 = 70;
int home_joint9 = 165;

#define neck_survoPin 10
#define trigPin 13
#define echoPin 12



void setup(){

  joint2.attach(2);
```

```
joint3.attach(3);
joint4.attach(4);
joint5.attach(5);
joint6.attach(6);
joint7.attach(7);
joint8.attach(8);
joint9.attach(9);

neck_servo.attach(neck_survo Pin ,  500, 2400);
neck_servo.write(90);

pinMode(echo Pin, INPUT);
pinMode(trigPin,OUTPUT);

joint2.write(home_joint2);
joint3.write(home_joint3);
joint4.write(home_joint4);
joint5.write(home_joint5);
joint6.write(home_joint6);
joint7.write(home_joint7);
joint8.write(home_joint8);
joint9.write(home_joint9);

delay(3000);

}
void  loop(){

sithome();
delay(2000);

stand1();
delay(2000);
```

```
neckrotate();
delay(2000);

forward(5);
delay(2000);

backward (5);
delay(2000);

rightturn(5);
delay(2000);

leftturn(5);
delay(2000);

twist();
delay(2000);

sithome();
delay(2000);

stand1();
delay(2000);

stand2();
delay(2000);

stand3();
delay(2000);

downaction (4);
delay(2000);
```

```
    wink (4);
    delay(2000);

}

//
void idle(){
    delay(100);
    }

//
void standhome(){
    joint2.write(home_joint2);
    joint3.write(home_joint3);
    joint4.write(home_joint4);
    joint5.write(home_joint5);
    joint6.write(home_joint6);
    joint7.write(home_joint7);
    joint8.write(home_joint8);
    joint9.write(home_joint9);
}

//
void sithome(){
        joint2.write(135);
        joint3.write(65);
        joint4.write(45);
        joint5.write(110);
        joint6.write(135);
        joint7.write(80);
        joint8.write(45);
        joint9.write(110);
```

```
}
//                    1
void stand1(){
  sithome();
  joint2.write(170);
  delay(300);
  joint2.write(home_joint2);
  delay(300);
  joint4.write(10);
  delay(300);
  joint4.write(home_joint4);
  delay(300);
  joint6.write(170);
  delay(300);
  joint6.write(home_joint6);
  delay(300);
  joint8.write(10);
  delay(300);
  joint8.write(home_joint8);
  delay(300);

  joint3.write(home_joint3);
  joint5.write(home_joint5);
  joint7.write(home_joint7);
  joint9.write(home_joint9);
}

//                    2
void stand2(){
  sithome();
  joint2.write(175);
  joint4.write(5);
  joint6.write(175);
```

```
    joint8.write(5);
    delay(600);

    joint2.write(home_joint2);
    joint4.write(home_joint4);
    joint6.write(home_joint6);
    joint8.write(home_joint8);
    delay(600);

    joint3.write(home_joint3);
    joint5.write(home_joint5);
    joint7.write(home_joint7);
    joint9.write(home_joint9);
}

//                3
void stand3()
{
    sithome();
    int i;
    int j = 90;
    int k =90;
    joint2.write(home_joint2);
    joint4.write(home_joint4);
    joint6.write(home_joint6);
    joint8.write(home_joint8);
    for(i=90; i<165; i++)
    {
        joint5.write(i);
        j=j-1;
        joint3.write(j);
        delay(20);
    }
```

```
    for(i = 115; i < 165; i++)
    {
        joint9.write(i);
        k = k - 1;
        joint7.write(k);
        delay(20);
    }
}

//                                   (                    )
void downaction(unsigned int step){
    while(step-- > 0){
    sithome();
    delay(100);
    standhome();
    delay(100);
    }
}

//
void wink(unsigned int step){
    standhome();
    joint9.write(home_joint9 - 30);

    while (step-- > 0){
    joint5.write(30);
    joint4.write(home_joint4 + 60);
    delay(300);
    joint4.write(home_joint4 - 60);
    delay(300);
    }
}
```

```
//
void twist(){
  joint3.write(home_joint3);
  joint5.write(home_joint5);
  joint7.write(home_joint7);
  joint9.write(home_joint9);

  for(int right=90;right<170;right++){
      joint2.write(right);
      joint6.write(right);
      joint4.write(right-90);
      joint8.write(right-90);
      delay(10);
  }

   for(int right=170;right>90;right--){
      joint2.write(right);
      joint6.write(right);
      joint4.write(right-90);
      joint8.write(right-90);
      delay(10);
  }

}

//        (                    )
void forward(unsigned int step){w
  hile(step-- > 0){
  joint3.write(home_joint3+30);
  joint7.write(home_joint7+30);
  delay(100);
```

```
    joint2.write(home_joint2+30);
    joint8.write(home_joint8-30);
    joint4.write(home_joint4);
    joint6.write(home_joint6);
    delay(100);
    joint3.write(home_joint3);
    joint7.write(home_joint7);
    idle();

    joint5.write(home_joint5-30);
    joint9.write(home_joint9-30);
    delay(100);
    joint2.write(home_joint2);
    joint8.write(home_joint8);
    joint4.write(home_joint4-30);
    joint6.write(home_joint6+30);
    delay(100);
    joint5.write(home_joint5);
    joint9.write(home_joint9);
    idle();
    }
}

//          (                    )
void backward(unsigned int step){w
    hile(step-- > 0){
    joint3.write(home_joint3+30);
    joint7.write(home_joint7+30);
    delay(100);
    joint2.write(home_joint2);
    joint8.write(home_joint8);
    joint4.write(home_joint4-30);
    joint6.write(home_joint6+30);
```

```
    delay(100);
    joint3.write(home_joint3);
    joint7.write(home_joint7);
    idle();

    joint5.write(home_joint5-30);
    joint9.write(home_joint9-30);
    delay(100);
    joint2.write(home_joint2+30);
    joint8.write(home_joint8-30);
    joint4.write(home_joint4);
    joint6.write(home_joint6);
    delay(100);
    joint5.write(home_joint5);
    joint9.write(home_joint9);
    idle();
    }
}


//                (                        )
void rightturn(unsigned int step){w
    hile(step-- >0){
    joint5.write(home_joint5-30);
    joint9.write(home_joint9-30);
    delay(100);
    joint2.write(home_joint2+30);
    joint8.write(home_joint8-30);
    joint4.write(home_joint4-30);
    joint6.write(home_joint6+30);
    delay(100);
    joint5.write(home_joint5);
    joint9.write(home_joint9);
```

```
    idle();

    joint3.write(home_joint3+30);
    joint7.write(home_joint7+30);
    delay(100);
    joint2.write(home_joint2);
    joint8.write(home_joint8);
    joint4.write(home_joint4);
    joint6.write(home_joint6);
    delay(100);
    joint3.write(home_joint3);
    joint7.write(home_joint7);
    idle();
    }
}


//                (                    )
void leftturn(unsigned int step){w
    hile(step-- > 0){
    joint3.write(home_joint3+30);
    joint7.write(home_joint7+30);
    delay(100);
    joint2.write(home_joint2+30);
    joint8.write(home_joint8-30);
    joint4.write(home_joint4-30);
    joint6.write(home_joint6+30);
    delay(100);
    joint3.write(home_joint3);
    joint7.write(home_joint7);
    idle();

    joint5.write(home_joint5-30);
```

```
    joint9.write(home_joint9-30);
    delay(100);
    joint2.write(home_joint2);
    joint8.write(home_joint8);
    joint4.write(home_joint4);
    joint6.write(home_joint6);
    delay(100);
    joint5.write(home_joint5);
    joint9.write(home_joint9);
    idle();
    }
}

//                                        (          OFF          )
void neckrotate(){
    int i=90;
    while(i < 150){
        neck_servo.write(i);
        i++;
        delay(5);
    }
    while(i > 30){
        neck_servo.write(i);
        i--;
        delay(5);
    }
    while(i <= 90){
        neck_servo.write(i);
        i++;
        delay(5);
    }
}
```

```
//
void neck_leftrotate(){
  int i=90;
  while(i < 150){
    neck_servo.write(i);
    i++;
    delay(5);
  }
}


//
void neck_rightrotate(){
  int i=90;
  while(i > 30){
    neck_servo.write(i);
    i--;
    delay(5);
  }
}


//
void neck_home(){
  neck_servo.write(90);
}
```

Conclusion

The method used to address the problem of obstacle overcoming was to divide a com-plex movement task in a group of simple movements implemented in a known sequence that ensures robot stability, rather than develop a complex model to search a generic so- lution for the problem. The main characteristic of this approach is that it allows the solution space dimension reduction and that it has no limits, being able to implement complex tasks such as going up in a stairs. The limits are the robot's leg's kinematic limits, which are very closed in robot. Robots with stronger actuators and wider kine- matic its could support the implementation of libraries of obstacles, associated with thework environment of low speed robots. It could be seen that that, according to obstacleshape and dimensions acquired from contact keys and to the robot kinematic limits, sets of simple movements are combined to overcome the obstacle giving navigation skill and autonomy to the legged robot in an environment with known obstacles. Although it'snot known the time of implementation and enforcement of more sophisticated modelsto perform the same tasks, the approach probably makes the implementation slower, but allows complex movements sequences implementations while ensuring the robot stability.

Future Scope

(1) The robustness of the mobile robot is increased due to high precise joint actuatorsand controller. Joint actuators play a significant role in accounting for the complexity,cost, and weight of robots. It is very essentially to ensure that the joint actuators have ahigh torque output to the weight ratio. In hydraulic and pneumatic actuating types have a high output power to weight ratio, fast response and easy implementation.

(2) It is also well known that quadruped mammals can be selected their synchro- nized pattern of leg movement (e.g., walking, galloping, trotting) according to speed. The quadruped robots are more favorable to incorporate Artificial Intelligence (AI) dueto their mobility and stability of locomotion.

(3) The current mobile robot systems are limited by the visual perception system.The features like recognize, memorizes, and learn from the environment should be fo-cused to implement to help the mobile robot for more autonomy.

(4) The human-robot interaction may be required in the form of physical and socialor emotional. The physical interaction of human-robot will help directly physical hand- icraft or use as service robotics for daily assistance.

References

- https://www.irjet.net/archives/V8/i5/IRJET-V8I5281.pdf
- https://www.sciencedirect.com/science/article/pii/S2090447920302501
- http://www.robotpark.com/academy/all-types-of-robots/legged-robots/four-legged- robots/
- Taha, I. A. & Marhoon, H. M. (2018). Implementation of Controlled Robot for Fire Detection and Extinguish to Closed Areas Based on Arduino. TELKOMNIKA, Vol. 16(2), pp 654-664.
- Kadir, W. M. H. W., Samin, R. E. & Ibrahim, B. S. K. (2012) Internet Controlled Robotic Arm. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), Procedia Engineering, Vol. 41, pp. 1065- 1071.
- Buchli, J., Kalakrishnan, M., Mistry, M., Pastor, P. & Schaal, S. (2009). Compliant Quadruped Locomotion over Rough Terrain. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 814-820.
- Filho, A. B., Amaral, P. F. S. & Pinto, B. G. M. (2010). A Four Legged Walking Robot with Obstacle Overcoming Capabilities. 3rd International Conference on Human System Interaction, pp. 374-379.
- Singh, A., Gupta, T. & Korde, M. (2017). Bluetooth controlled spy robot. IEEE, International Conference on Information, Communication, Instrumentation and Control (ICICIC 2017), pp. 1-4.
- Fankhauser, P. (2018). Perceptive Locomotion for Legged Robots in Rough Terrain. Autonomous Systems Lab & Robotic Systems Lab, Institute for Robotics and Intelligent Systems, ETH Zurich (2018), DISS. ETH No: 24849, pp. 1 -174.
- Geva, Y. & Shapiro, A. (2014). A Novel Design of a Quadruped Robot for Research Purposes", International Journal of Advanced Robotic Systems (2014), Vol. 11(7), pp. 1-13.
- Dat, T. T. K. & Phuc, T. T. (2014). A Study on Locomotions of Quadruped Robot. Lecture Notes in Electrical Engineering (January 2014), pp. 595-604.
- Oak, S. & Narwane, V. (2014). Design, Analysis and Fabrication of Quadruped Robot with Four bar Chain Leg Mechanism. International Journal of Innovative Science, Engineering & Technology (IJISET), Vol. 1(6), pp. 1-6

FINAL PROJECT FIGURE