DESIGN OF WHEG ROBOT FOR MULTI TERRAIN MOBILITY

A PROJECT REPORT

Submitted by

SANYATJEET PAWDE [RA1711004010048]
ROHAN TEHALYANI [RA1711004010050]
MITRANSH CHOUBISA [RA1711004010058]
MAYANK KUMAR [RA1711004010064]

Under the guidance of

Dr. ESWARAN P, Ph.D

(Associate Professor, Department of Electronics and Communication Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

JUNE 2020

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**DESIGN OF WHEG ROBOT FOR MULTI TERRAIN MOBILITY**" is the bonafide work of "**SANYATJEET PAWDE[RA1711004010048], ROHAN TEHALYANI[RA1711004010050], MITRANSH CHOUBISA[RA1711004010058],MAYANK KUMAR [RA1711004010064],** , who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. ESWARAN P, Ph.D
GUIDE
Associate Professor
Dept. of Electronics and Communication Engineering

SIGNATURE

Mr. SRINATH B, M.E
CO-ORDINATOR
Assistant Professor(Sr.G)
Dept. of Electronics and Communication Engineering

ABSTRACT

With the advancements of robotics in different fields, there is an on going need for research and development of different types of robots required at different places. The need of today's robots is to be able to run, climb, jump and avoid obstacles. This gives a lot of scope for different designs of a robot's body in modern robotics. Whegged robots are one such robot specially designed to be capable of climbing and avoiding different hindrances because of their unique design. The goal of our project is to control these specially designed robotic legs with the help of available knowledge of electronics and make it more control friendly. The project aims to look for techniques to control such robots wirelessly over radio frequency, so that such robots can further be used in different fields, be it military use or space missions. Many renowned organisations around the world like MIT and NASA are developing such all-terrain robots for different purposes like research or space missions. Our project targets to design one such miniature Wheg robot and establish best mode of communication between robot and control station.

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to our guide, Dr. Eswaran P and to our co-ordinator Mr. Srinath B for their valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this research work.

Project Group

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

COM	Centre of Mass
UGV	Unmanned Ground Vehicle
ISR	Intelligence Surveillance and Reconaissance
COG	Centre of Gravity

LIST OF SYMBOLS

F	Force, N
T	Torque ,kgcm
L	length, cm
m	mass, gm
g	acceleration due to gravity, m/s^2
θ	critical angle(angle of elevation), $^\circ$degrees
μ	coefficient of friction,dimensionless
h_{cog}	height of centre of gravity,cm
w	width of vehicle,cm
a	side length,cm

CHAPTER 1

INTRODUCTION

1.1 Whegs

Engineering problems can be solved using biological inspiration in varying degrees, from a direct implementation to an abstracted one. Difficulties arise in the direct implementation of biological attributes because, oftentimes, the necessary technology has yet to be developed. The design and testing of this technology can be a slow process, which ultimately causes its employment to be long-term. However, implementing abstracted biological attributes utilizing existing technology enables mission capable solutions in the near-term.

One such solution is provided by the abstraction of arthropodic movement being implemented in mobile robots known as Whegs(wheel-legs or Wing-legs).Whegs are mechanisms for robot locomotion. Whegs use a strategy of locomotion that combines the simplicity of the wheel with the obstacle-clearing advantages of the foot.

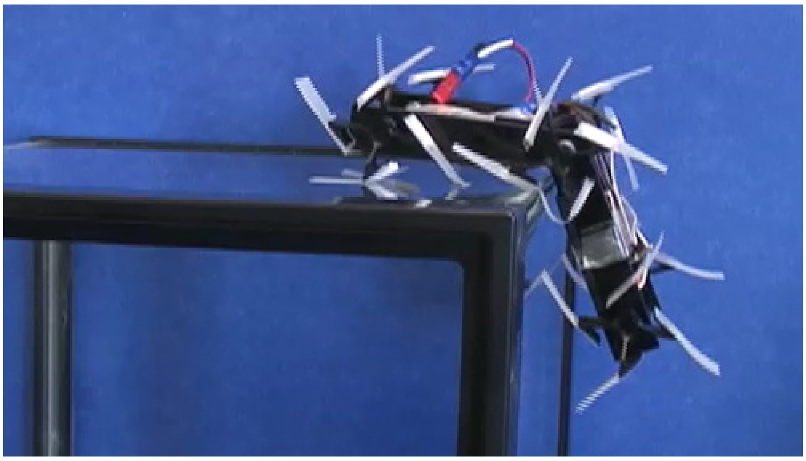

Figure 1.1: Obstacle clearence using whegs
[11]

Whegs were pioneered at the Biologically Inspired Robotics Laboratory at Case Western Reserve University. Whegs development and improvements based on cockroach climbing behavior in several robots has been done in collaboration with the Ritzmann lab in the Biology department at Case Western Reserve University on cockroach climbing behavior. Whegs robots were inspired by the Prolero robot, designed in 1996 at the European Space Agency, and the RHex robot, developed by a multiuniversity effort funded by the Defense Advanced Research Projects Agency. The mobility system is based on studies on the locomotion of the cockroach. Wing-legs are found on flying robots and are wings dual-purposed as legs for locomotion when the robot is on the ground.

1.1.1 Biologically inspired mechanisms

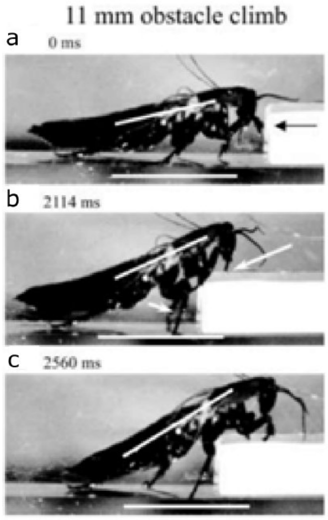

Figure 1.2: Cockroach climbing an obsatcle
[9]

Cockroaches have remarkable locomotion abilities. One solution to the problem of producing highly mobile amphibious robots is to design a vehicle with the mechanisms responsible for the mobility of a cockroach. In studies of cockroach movement, we have noted the following locomotion principles. A cockroach has six legs, which support and move its body. It typically walks and runs in a tripod gait where the front and rear legs on one side of the body move in phase with the middle leg on the other side. The front legs swing head-high during normal walking so that many obstacles can be surmounted without significant gait changes. However, its gait changes when it encounters larger barriers.[1]

The cockroach turns by generating asymmetrical motor activity in legs on either side of its body as they extend during stance. A cockroach enhances its climbing abilities by changing its body postures before and during a climb over an obstacle . It uses its middle legs to pitch its

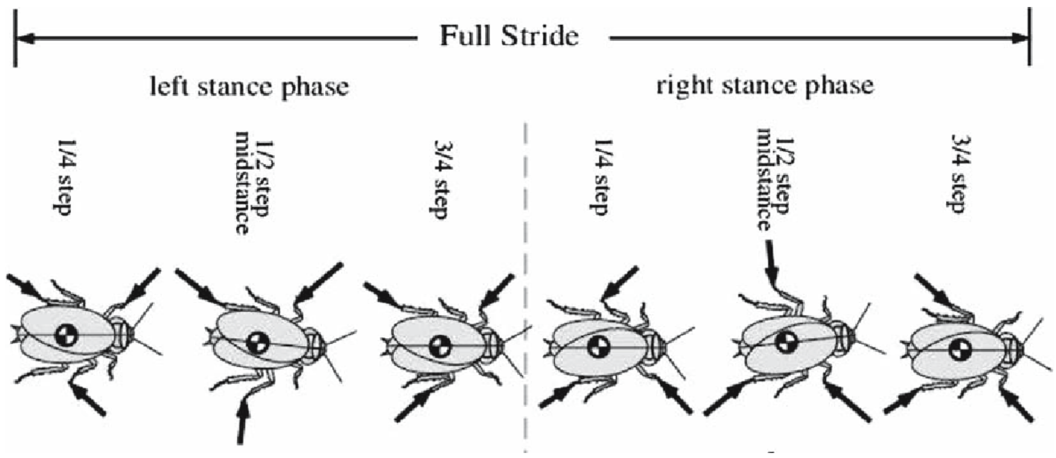

Figure 1.3: Double tripod gait of a cockroach
[7]

body up, prior to climbing obstacles that are higher than its head. This behaviour enables its front legs to reach higher. Also, during a climb it uses its body flexion joints to bend the front half of its body down to avoid high centering.

A cockroach enhances its climbing abilities by changing its body postures before and during a climb over an obstacle . It uses its middle legs to pitch its body up prior to climbing obstacles that are higher than its head. This behaviour enables its front legs to reach higher. Also, during a climb it uses its body flexion joints to bend the front half of its body down to avoid high centering.

1.1.2 Obstacle dependent speed and phase shift

Many legged animals change gaits when increasing speed. In insects, only one gait change has been documented so far, from slow walking to fast running, which is characterised by an alternating tripod. Studies on some fast-running insects suggested a further gait change at higher running speeds. Apart from speed, insect gaits and leg co-ordination have been shown to be influenced by substrate properties, but the detailed effects of speed and substrate on gait changes are still unclear. Here we investigate high-speed locomotion and gait changes of the cockroach Nauphoeta cinerea, on two substrates of different slipperiness.

Temporally distributed leg force application as resulting from metachronal leg coordination at high running speeds may be particularly useful in animals with limited capabilities for elastic energy storage within the legs, as energy efficiency can be increased without the need for

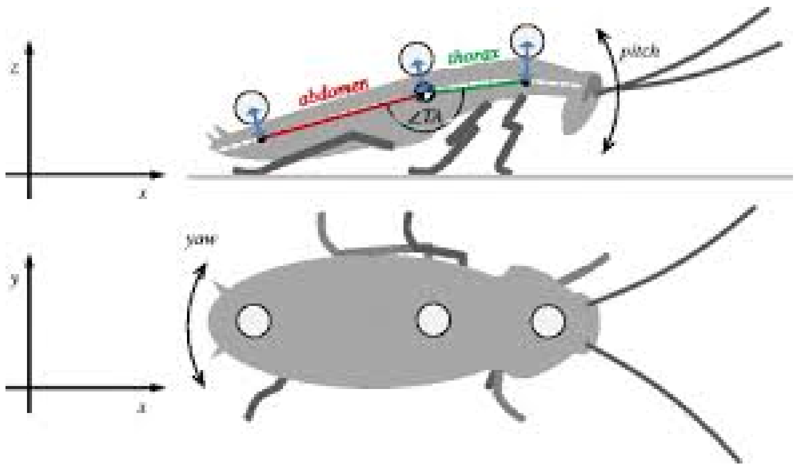

Figure 1.4: Speed Dependent phase shift in a cockroach
[10]

elasticity in the legs. It may also facilitate locomotion on slippery surfaces, which usually reduce leg force transmission to the ground. Moreover, increased temporal overlap of the stance phases of the legs likely improves locomotion control, which might result in a higher dynamic stability.

1.1.3 Comparison and Deduction of wheg mobility by insect gait

The nominal locomotion pattern found in cockroaches and most other insects is the tripod gait in which the front and rear legs on one side move in approximate synchrony with the middle leg on the opposite side. This tripod alternates with a second tripod made up of the remaining legs. Each leg moves rapidly through the swing phase and then slows down as it enters the stance phase and pushes the body forward. The front legs move through an arc on the order of the animal's height in order to reach the top of small barriers without the need for any change in leg movement . The highlighted legs push forward while the remaining legs are lifted and moving forward preparing for the next step, then the highlighted legs lift and move forward while the remaining legs are now used to push. The Whegs design incorporates the six-legged locomotion (two wheel-legs on each of three axles) as well as compliance in the wheel-legs to enable the platform to passively go in and out of the tripod gait through the use of a modified limited slip differential.

During normal travel, the left and right wheel-legs (shown as black and gray, respectively) are 60 degrees out of phase. As the axle strikes the obstacle, the left wheel-leg is applying

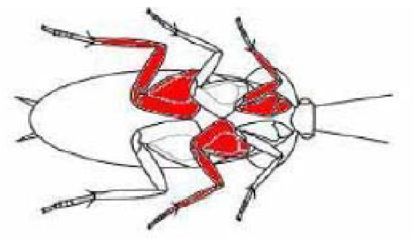

Figure 1.5: Synchronous Movement of Opposite Legs
[8]

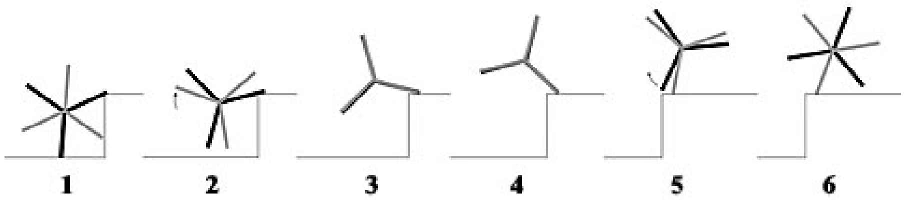

Figure 1.6: Single Axle response to an obstacle

most of the force to climb up, while at the same time, due to the compliance that is designed in, is rotating until it is in phase with the right wheel-leg and the force of lifting the axle over the obstacle is distributed for even support and maximum lifting power. Once the obstacle is overcome, the left wheel-leg quickly rotates back into phase allowing the platform to resume its tripod gait.

1.2 Objectives

To design a wheg and wheel transformable robot

Control it using a RC console

Program it using arduino ide to control the movement of the wheels, joints and whegs

Automate it.

To take inspiration from arthropodic movement and develop a model with similar ability of motion over various terrains.

Thereby making a transformable and convertible robot rather than a normal Wheg robot for more mobility and obstacle clearance. and develop a radio frequency based communication system for better accessibility of the robot and to achieve mobility over both rough and smooth surfaces, i.e., plain terrains, sand dunes, craters,multiterrain.

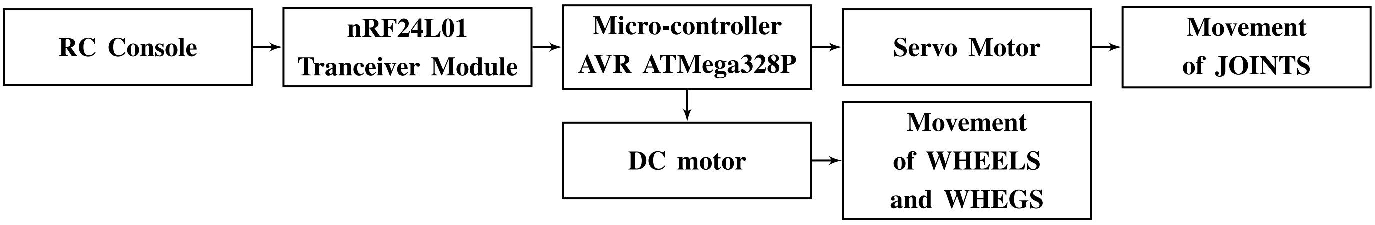

Figure 1.7: Overall Framework

1.3 Applications

Some of the fields in which our proposed design would prove to be an efficient aid are

Space Exploration

Archaeological Excavation

Military Reconnaissance

Underwater Exploration

Amphibious Operations

Rescue operations

Biomedical Designs

Automobile Designs

1.4 Motivation

To experiment with the already existing technology and try to make it more efficient and feasible for long range and all terrain purposes, such as a better communication system using radio frequency and a wheel cum wheg design for better mobility.

CHAPTER 2

LITERATURE SURVEY

2.1 Comparing Cockroach and Wheg Body Motions

In this paper it is found that abstracting cockroach locomotion principles with reduced actuation can lead to a simple yet effective cockroach-like vehicle able to traverse irregular terrain. One such hexapod robot is Whegs VP, which achieves a cockroach-like nominal tripod gait using only a single DC motor. Whegs VP uses compliant mechanisms in its axles to passively adapt its gait to the terrain such that it can climb obstacles 175 percent of its leg height. High speed video analysis of walking cockroaches and Whegs VP illustrates that Whegs VP walks with cockroach-like body motions. The experiments indicate that stepping patterns play a major role in an animal's or robot's overall body motions.[7]

2.2 Speed dependent Phase Shifts and Gait changes in Cockroaches running on substrates of different slipperiness

In this paper the analysis of leg co-ordination and body oscillations for straight and steady escape runs revealed that at high speeds, blaberid cockroaches changed from an alternating tripod to a rather metachronal gait, which to our knowledge, has not been described before for terrestrial arthropods. Despite low duty factors, this new gait is characterised by low vertical amplitudes of the Centre of Mass (COM), low vertical accelerations and presumably reduced total vertical peak forces. However, lateral amplitudes and accelerations were higher in the faster gait with reduced leg synchronisation than in the tripod gait with distinct leg synchronisation.[10]

2.3 Wheg Development for Advanced Mobility

There is a significant amount of interest in autonomous, amphibious robots capable of operating in the surf zone. Such robots would be capable of preparing amphibious landing zones by locating, classifying and mapping potential threats to a landing force such as obstacles and mines as well as performing general reconnaissance of the beachhead. In order to accomplish this, the vehicle would have to be rugged, highly mobile and capable of navigation in possibly extreme environments. In order to achieve the level of mobility required, recent research has focused on the use of legged or crawling platforms.[8]

2.4 Design Analysis of a Remote Controlled Robotic Vehicle

In this paper the design of a Remote Controlled Robotic Vehicle has been completed. A prototype was built and confirmed functional. This system would make it easier for man to unrivalled the risk of handling suspicious objects which could be hazardous in its present environment and workplace. Complex and complicated duties would be achieved faster and more accurately with this design.[6]

2.5 Design and Analysis of A Miniature Two-Wheg Climbing Robot with Robust Internal and External Transitioning Capabilities

In this paper the climbing robots have many benefits such as a highly expanded workspace and the ability to reach or accomplish otherwise impossible spots or tasks for ground robots. When climbing robots are used collaboratively with Unmanned Ground Vehicle (UGV) for indoor autonomous Intelligence Surveillance and Reconaissance (ISR) missions, they can be a powerful swarm that collectively completes tasks such as mapping, detection, monitoring and tracking. Specifically, the climbing robot can be used to provide an overall image of the area and as a bridge for communication for robots on different floors. To complete such tasks, the robot may be required to climb obstacles and transition from one plane to another, both internally (concave angle) and externally (convex angle).[2]

2.6 WHEGSTR: A Multiterrain Robot with C Shaped Whegs, Implementation of Error Minimization Technique and using Artifical Neural Network (ANN)

From this paper it was deduced that this typical design will provide maneuverability and stability according to the terrain. With its intrinsic mobility,robot will have the ability to adapt according to the terrain and will maintain its balance using algorithms based on differential equations. An unstrapped robot will easily traverse the complexity and diversity of the terrain with compliant legs. The extraordinary nature of the ladder gymnastics brings itself to a solid platform, whereas mountain platforms can be very difficult to stairs. Tracked vehicles must be able to overcome three steps to overcome these obstacles in a stable fashion, and still avoiding due to low traction of steps. Due to its rigid body and strong build, the robot can withstand impacts without compromising its performance. It will also transmit live broadcast of the surrounding area. Moreover, it will give a continuous feedback about the physical conditions i.e. temperature, pressure, altitude, humidity etc. of the nearby area.[3]

2.7 Application of Radio Frequency Controlled Intelligent Military Robot in Defense

In the present paper the authors tried to explore how a radio frequency controlled robot can be used in defense and in real war field. The military robot will be able to substitute the real human soldier in the battle field. The robot can be controlled from a base station by means of radio frequency. It also has the ability to re-establish contact with the base station in case of a signal failure by retracing its path back for some distance. [5]

CHAPTER 3

SYSTEM ANALYSIS

3.1 Robot's Stability Calculation

3.1.1 Case 1: Inclination

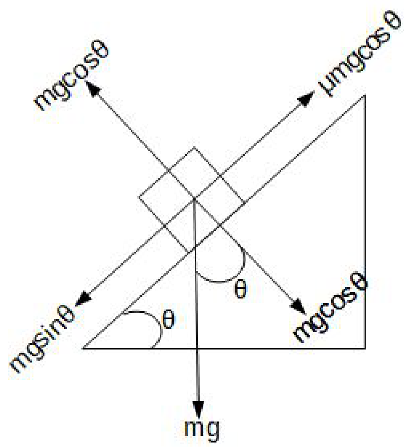

Figure 3.1: Robot on an inclination

Now robot can either climb an inclination or be moving forward on an inclined plane. In either case there is a chance for it to topple.

So, for the stability of robot we must calculate the critical angle for inclination, i.e., the angle of inclination at which the robot may topple.

The basis of our calculation is **Newton's law of motion**. We are considering the free body diagram of robot with a mass 'm', and a gravitation acceleration acting on it as 'g'.

From the figure we can see the force mg acting towards ground and its cosine component acting towards inclination. Now from the figure

$$mgsin\theta = mgcos\theta \tag{3.1}$$

$$tan\theta = \mu \tag{3.2}$$

$$\theta = tan^{-1}(\mu) \tag{3.3}$$

The other case of our robot in which it may topple is when the robot's legs are not parallel to the body. This can be deduced by calculating the height of centre of gravity of the bot. This can be calculated from the critical angle. We are taking the assumption of the static stability factor. **In eq^n(3.3) we calculated critical angle**.

$$tan\theta = \frac{w}{2 * h_{cog}} \tag{3.4}$$

$$h_{cog} = \frac{w}{2 * tan\theta} \tag{3.5}$$

3.1.2 Case 2: The change in centre of mass and centre of gravity when legs are changing their inclinations with respect to others

Now this theory is assumed by creating an analogy. We consider the body of the robot, i.e., the robot's chassis to be a square for simplification. Now this square is divided into four more equal squares.

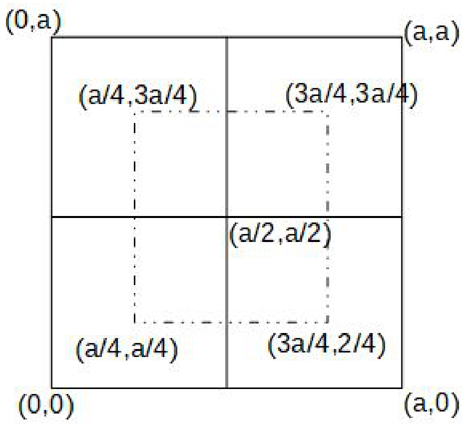

Figure 3.2: Analogy of chassis body with a square in Cartesian system

From the figure we can understand how the assumptions are made. The very first analogy is that of chassis being a square, the second is considering this square to be placed on the Cartesian coordinate system.

Now we are assuming that whenever a leg of body is changing its alignment with respect to other legs than the cog and com of the body will move towards the opposite direction. This theory is based on the fact that centre of gravity of square is at the same place as centre of mass, i.e., in the centre of the square. Thus, if the leg changes direction this will be assumed as a square going missing from the set. Let us see one case.

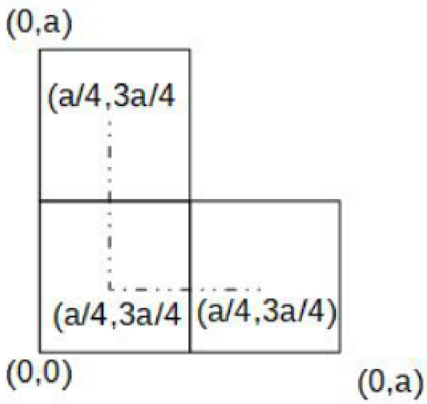

Figure 3.3: Analogy of one leg in different position

Here we are assuming that when one leg is not aligned with other ones than we my remove one square to calculate the shift of centre of mass or centre of gravity, using this calculation we will be able to understand the change in the servo motor's angles.

The calculation is as follows,

$$X = \frac{m * ((a/4) + (a/4) + (3a/4))}{3m} \tag{3.6}$$

$$X = \frac{5a}{12} \tag{3.7}$$

$$Y = \frac{m * ((a/4) + (a/4) + (3a/4))}{3m} \tag{3.8}$$

$$Y = \frac{5a}{12} \tag{3.9}$$

In the above equations, m is mass of each square body, a is side length.

From eq (3.7) and (3.8), we see the shift in the Centre of Gravity (COG) and COM of the system. This analogy helps us to understand the shift of balance and mass in the robot as well. If we consider the robot as the same system, then these assumptions help us to understand the shift in angle when the legs are moving independently.

This was one such case where the system is changing, there will be other cases as well but this basis of calculations will easily help us to deduce a system for them as well.

CHAPTER 4

SYSTEM DESIGN

4.1 System Integrated Design

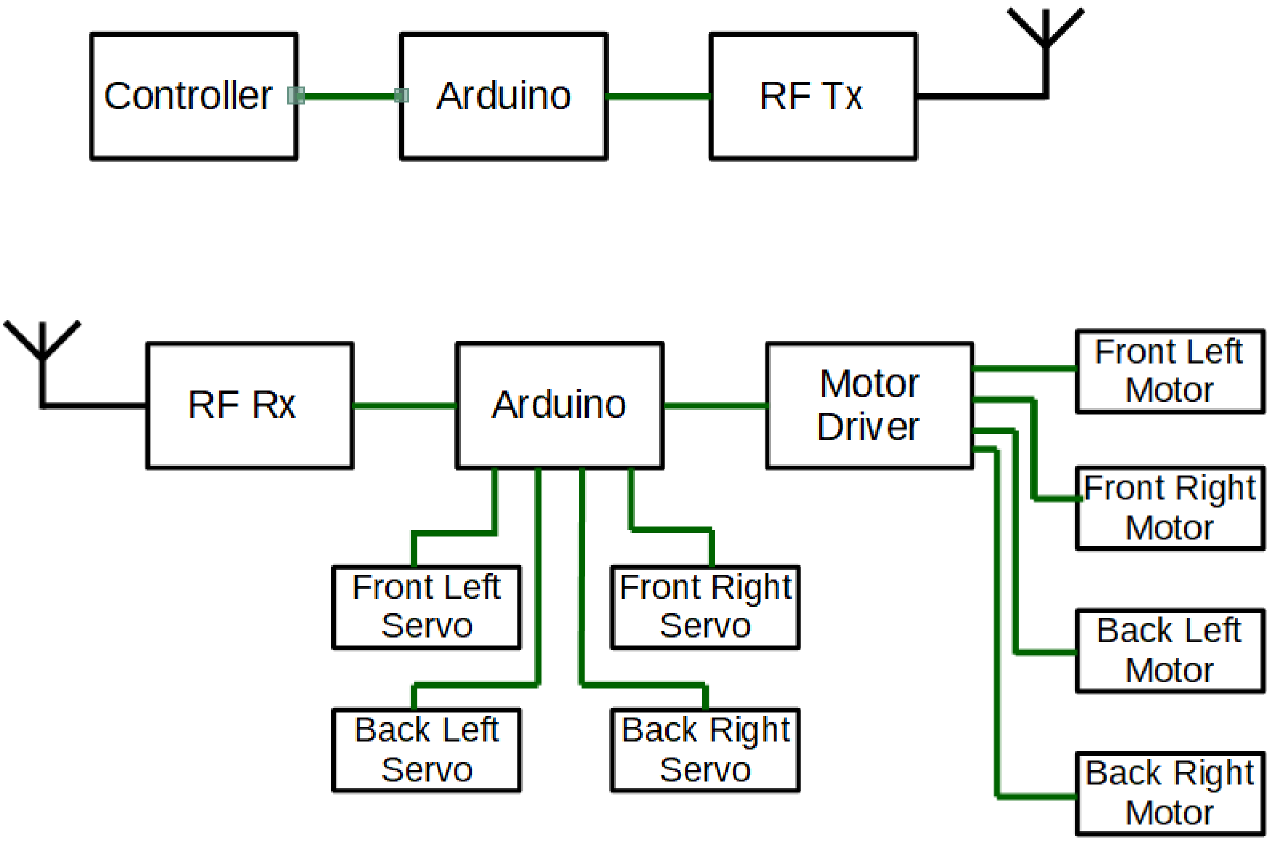

Figure 4.1: System Integrated Design

By considering the several approaches, to avoid many technical challenges such that of a shaft and axle design which was inefficient over inclines as the whole chassis of the robot would topple[4], we have proposed a conceptualised integrated design of both wheels and whegs.The robot would be controlled by a simple gaming console by the user which is feasible while using radio frequencies to communicate with the bot. Our design incorporates a chassis which would act as a cabinet to house the electronic circuitry and components.

The whole body of the bot would be made of **Polyacrylonitrile** also know as Acrylic material which is light and durable polymer , to reinforce the body we would add sheets of **Aluminium** metal.The legs would be joined to the servo motors on four corners of the chassis and subsequently the wheels would be joined to the legs via DC motors.The whegs would be integrated at the **Thoracic** region of the bot as compared to cockroach and their motion would be controlled by DC motors.

To enable a long range and less attenuated communication channel between user console and the bot we would use **nRF** module , which is Radio Frequency Tranceiver.

By the aforementioned concepts we would be able to overcome the technical challenges in already existing technologies.

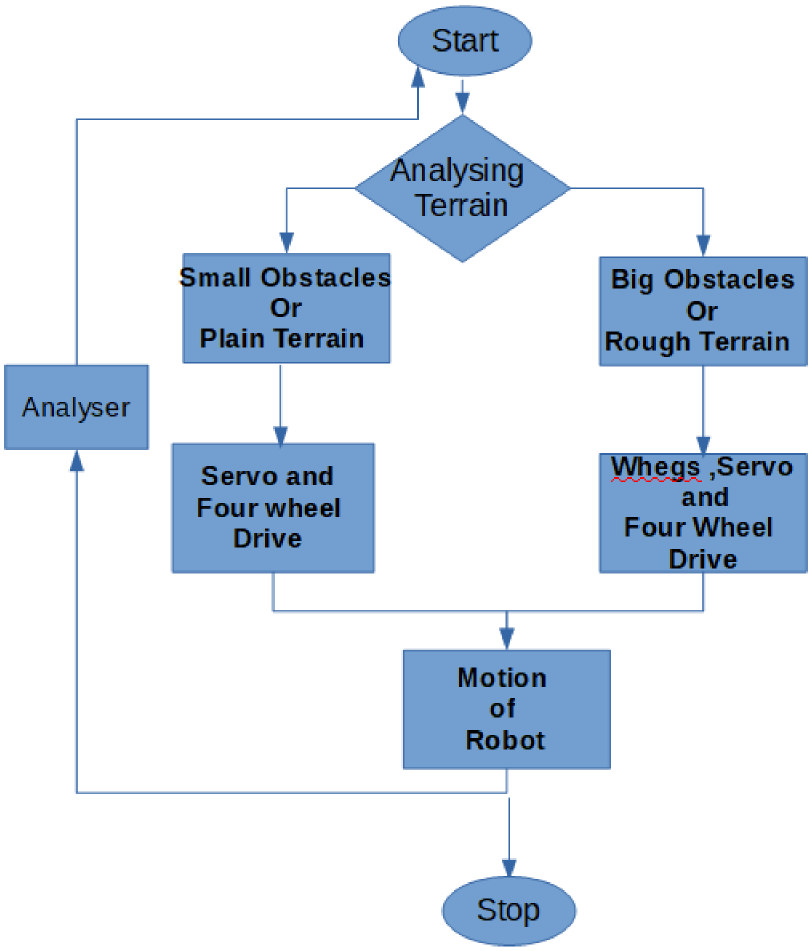

Figure 4.2: Conceptual Flowchart

.

Table 4.1: 3D model Specifications

Name of the part	Motion	Dimensions
Leg	extension/linear	15*3cm
Body(Base)		18*12cm
Body(Top plate)		18*4cm
Wheels	rotation	Diameter: 4cm
Whegs	revolute(swoop)	Diameter:3.5cm

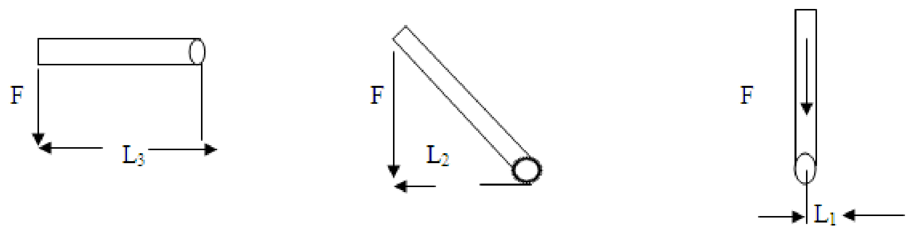

Figure 4.3: Required torque at each joint

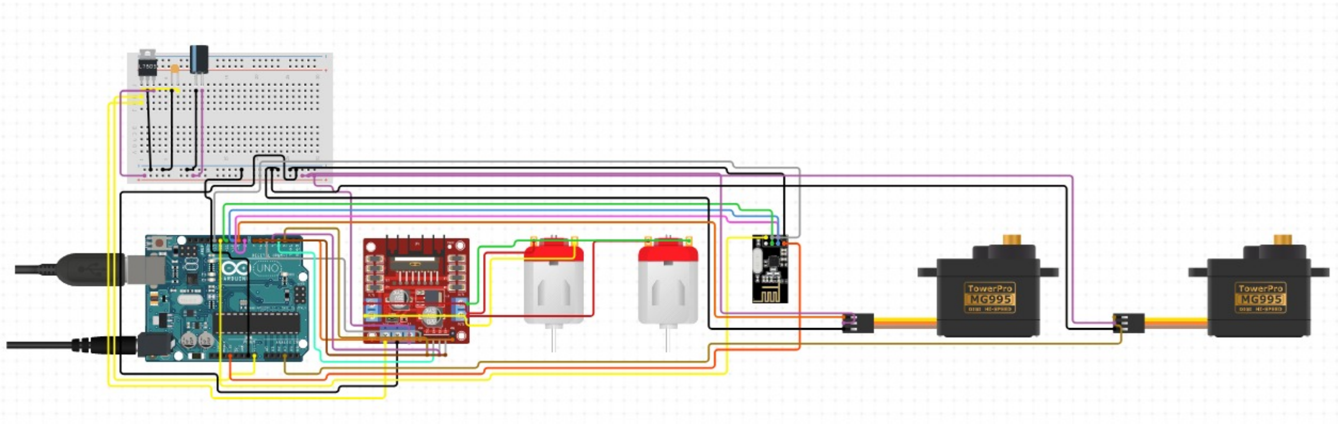

Figure 4.4: Prototype Model

4.2 Design Specifications

4.2.1 Servo Motors(MG995)

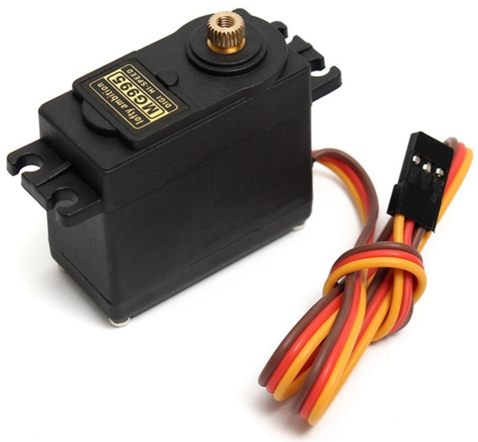

Figure 4.5: MG995 motor to be used for movement of joints

1. **Size** 40.7 x 19.7 x 42.9mm
2. **Weight** 55g (Include a cable and a connector)
3. **Speed** 0.2sec/60° (4.8V) 0.16sec/60° (6.0V)
4. **Torque** 9.4kgf-cm (4.8V) 11kgf-cm (6.0V)
5. **Voltage range** 4.8V-7.2V
6. **Operating Temperature** -30° to 60° C
7. **Rotational Degree** 180°
8. **Dead Band Width** 1 micro-sec
9. **Connector type** JR type '(Yellow: Signal, Red: VCC, Brown:GND)'

4.2.2 DC Motor

1. **Gearbox Diameter** 37mm
2. **Motor Diameter** 25mm

Figure 4.6: DC Motor to be used for movement of wheels

3. **Motor Length** 75mm
4. **Shaft Length** 22mm
5. **Shaft Diameter** 06mm
6. **Weight** 80g
7. **Rated RPM** 300
8. **Stall Torque** 3.5kgf-cm
9. **Rated Torque** 1.2kgf-cm
10. **Voltage range** 4-12V(DC)
11. **Load Current** 0.3A
12. **No load current** 60mA

4.2.3 Dual Shaft Motor

1. **Operating Voltage** 3-12V(DC)
2. **Rated Speed** 100RPM
3. **Rated Torque** 0.35kg-cm
4. **Shaft Length** 8.5mm
5. **Shaft Diameter** 5.5mm
6. **No Load Current** 40-180mA

Figure 4.7: Dual Shaft Motor to be used for movement of Whegs

4.2.4 MICROCONTROLLER Specifications

MICROCONTROLLER- Arduino UNO board with 8 bit ATmega328p AVR

1. **Power** Vin,3.3V,5V,GND pins for power I/O. operating voltage of 5V and recommended input of 7-12 volts. 50mA current is drawn.

2. **I/O pins** digital pins 0-13, analog pins(A0-A5 0-5V),PWM pins(3,5,6,9,11) for 8 bits pwm I/O.

3. **Serial pins** Rx(0);Tx(1); for transmission and reception of TTL serial data

4. **Microcontroller** ATmega 328p AVR 8 bit microcontroller high performance ,low power controller from Microchip previously manufactured by atmel.

5. **CPU** AVR CPU - Advanced Virtual RISC CPU RISC - reduced instruction set computer Alf-Egil Bogen Vegard Wollan RISC .

Figure 4.8: Arduino Uno

6. **Crystal Oscillator** used for square pulse generation as clock with 16MHz frequency. as it takes (1/16micro seconds) for 1 state
7. **EEPROM** 1Kb
8. **SRAM** 2Kb
9. **Flash Memory** 32Kb for storage of input code

4.2.5 Transceiver Module Specifications

nRF24L01 Module

1. **Power** (Vcc) Powers the module. Connect to +5V Supply voltage
2. **TX** Transmitter Transmits Serial Data.
3. **RX** Receiver Receive Serial Data.
4. **Operating Voltage**: 3.3V
5. **ChannelRange** :125
6. **Maximum pipeline/node**: 6

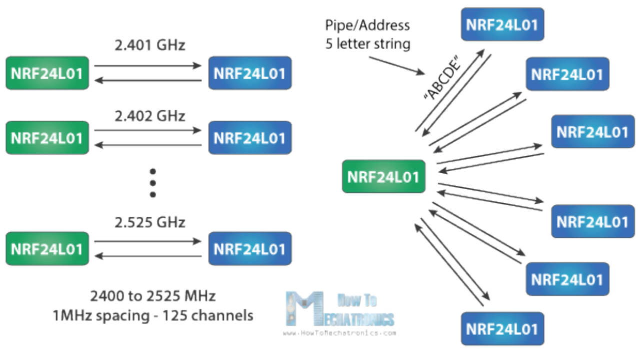

Figure 4.9: nRF TX/RX channels

7. **Frequency**:2.4GHz ISM band
8. **Operating Current**: 50mA-250mA(maximum)
9. **Range**: 50-200feet
10. Works with Serial communication (USART) and TTL compatible
11. SPI communication protocol.
12. Can operate in Master, Slave or Master/Slave mode
13. Supported **baud rate**: 250kbps to 2mbps.

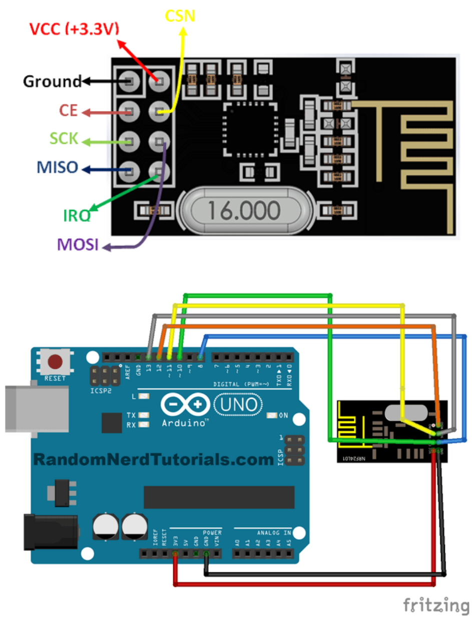

Figure 4.10: nRF module pinout and arduino connection

4.2.6 Motor Driver IC L293D Specifications

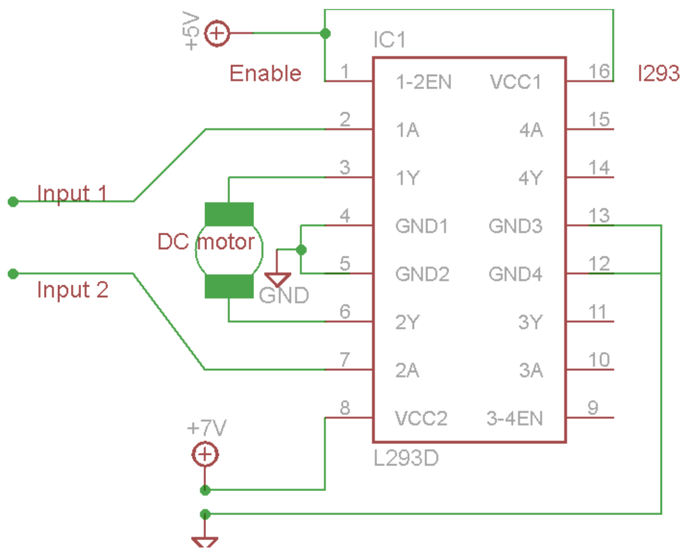

Figure 4.11: L293D Motor Driver IC

1. **Supply voltage,** VCC_1 36V(MAX)
2. **Output supply voltage,** VCC_2 36V(MAX)
3. **Input voltage,** V_I 7V(MAX)
4. **Output voltage,** V_O -3 to VCC_2 +3 V
5. **Peak output current,** I_O **(nonrepetitive, t < 100microsec)** -1.2 to 1.2 A

6. **Continuous output current, I_O: L293D** -600 to 600mA

7. **Maximum junction temperature,** T_J 150°

8. **Storage temperature,** T_stg -65° to 150°C

4.2.7 Controller Console Specifications

Logitech F310 Gamepad

Figure 4.12: Controller

1. 8-Way Programmable D-pad
2. Two Programmable Analog Mini-Joysticks
3. 10-Programmable Buttons
4. Programmable Left and Right Analog Triggers
5. Product Dimensions 17.2 x 7.5 x 20.5 cm
6. Batteries: 2 AAA batteries required.
7. Item Weight 295 g

4.2.8 Conceptual Sketches using CAD

These sketches are made using **Fusion360** depicting various stages of conceptual design.

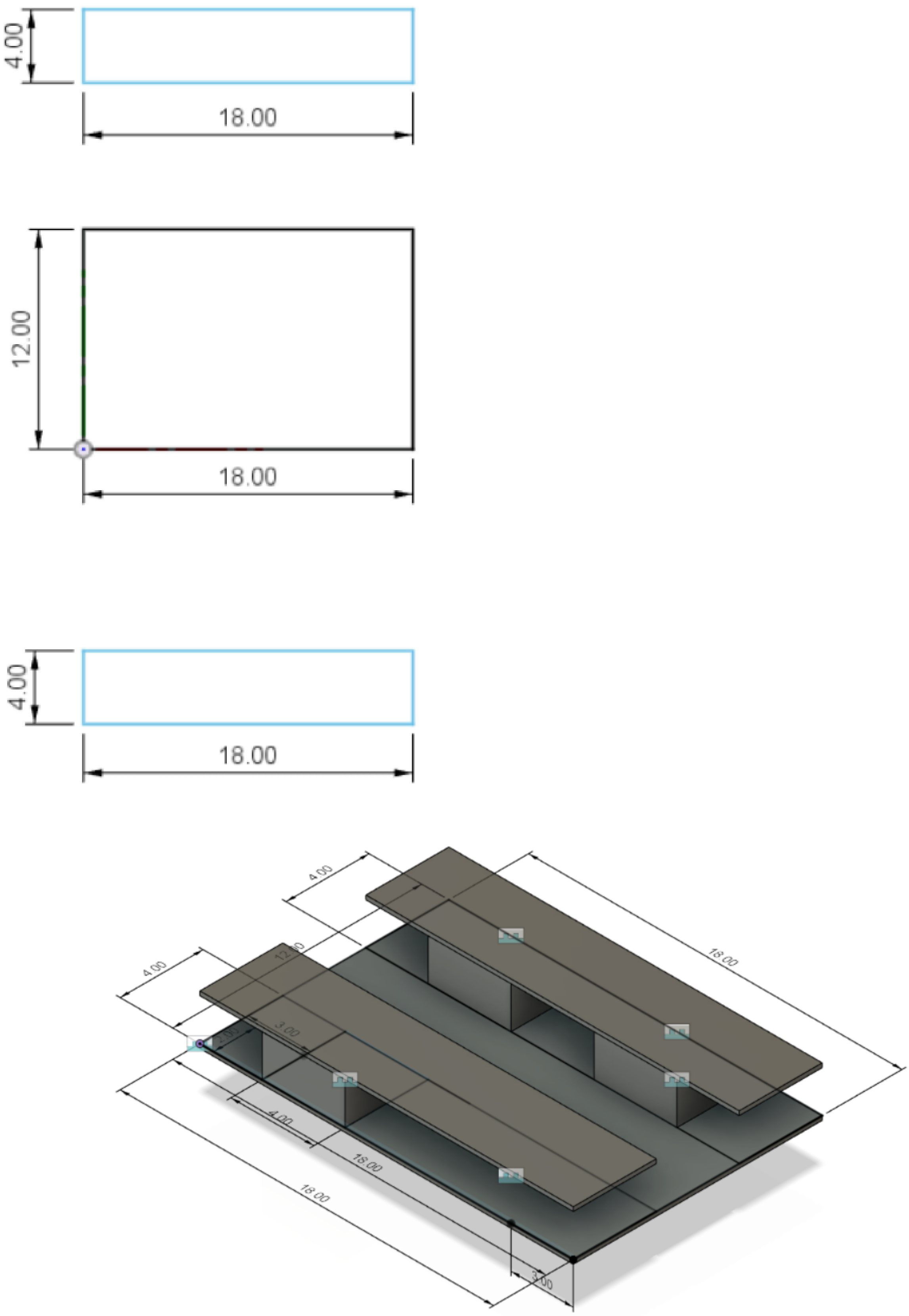

Figure 4.13: Chassis Body/Electronic Cabinet

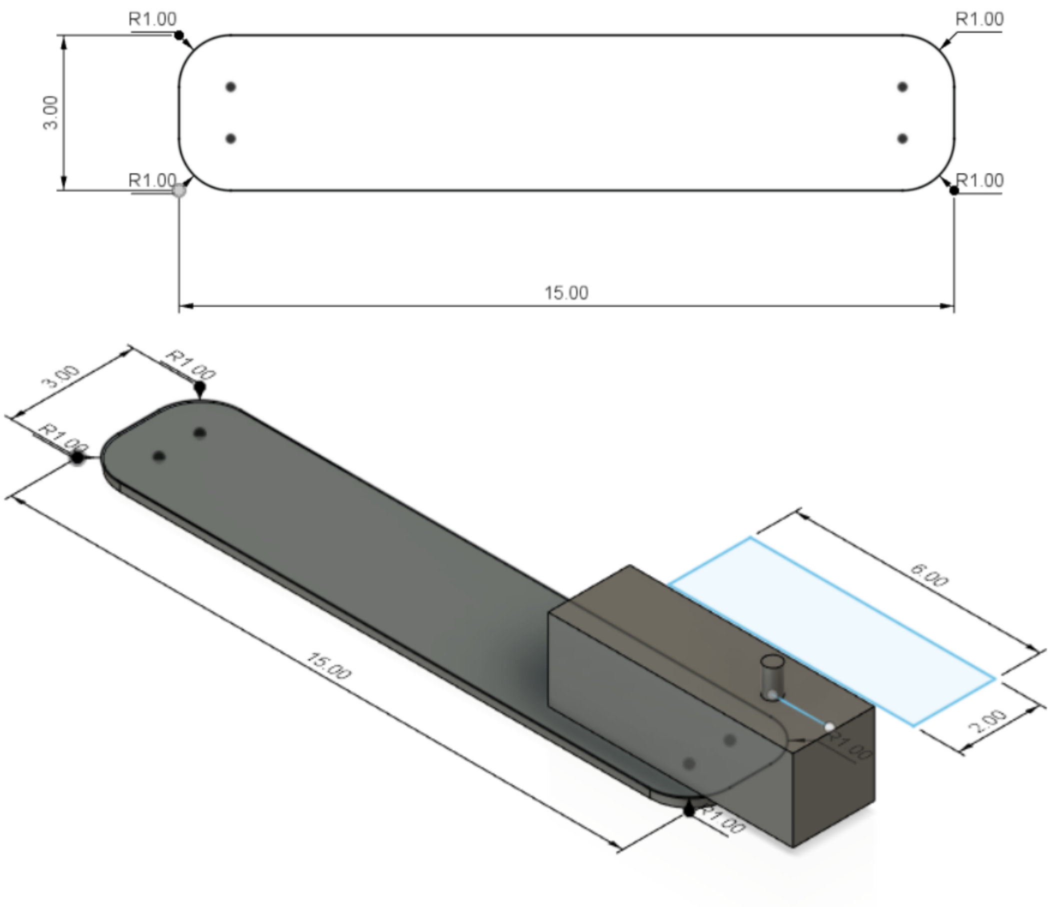

Figure 4.14: Leg Design

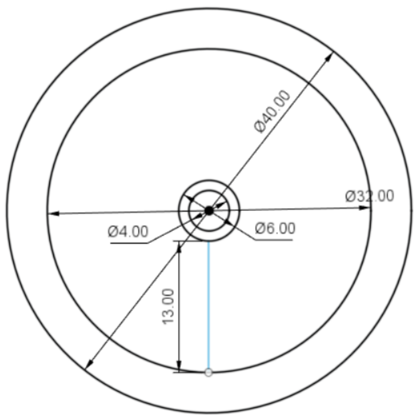

Figure 4.15: Wheel Dimensions

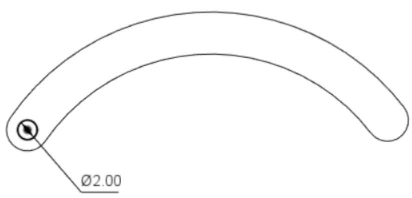

Figure 4.16: Wheg Dimension

Figure 4.17: Whegs Design with Shaft

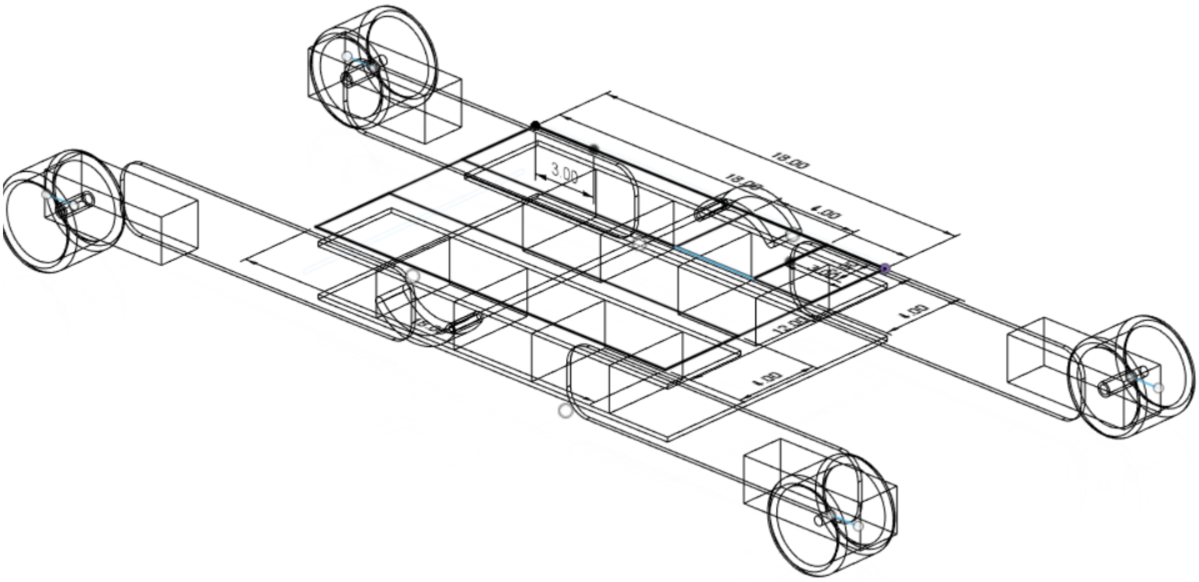

Figure 4.18: Final Assembly Sketch

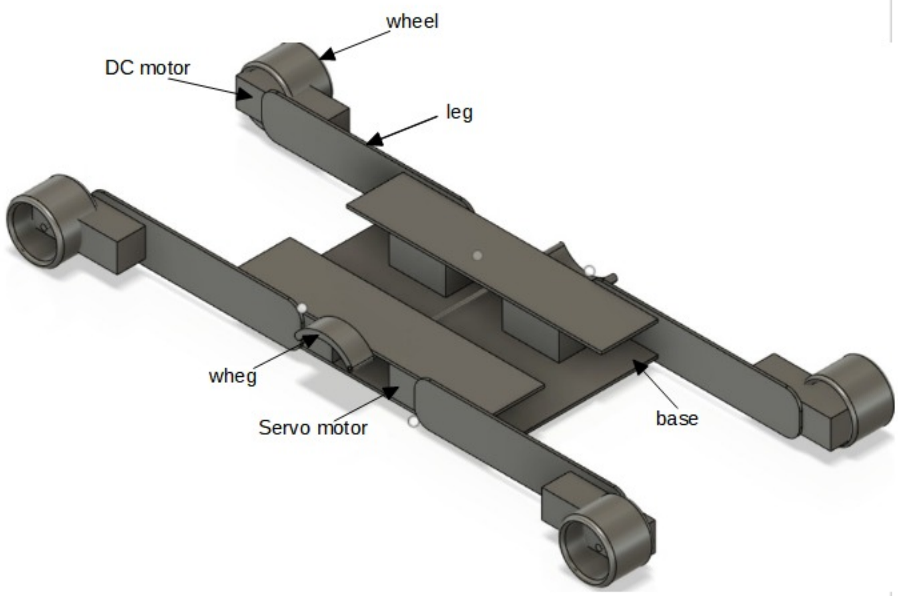

Figure 4.19: Final Assembly Design

CHAPTER 5

CODING

5.1 CODE

5.1.1 Receiving user input from controller and sending it the Arduino

The task is to take user input from a gaming console and use that data to control the robot's movement. The controller used here is Logitech f310 gamepad. It has two analog inputs and rest digital push buttons. The data from the controller can be taken by using the library "pygame". Pygame is a Python wrapper module for the SDL multimedia library. It contains python functions and classes that will allow you to use SDL's support for playing cdroms, audio and video output, and keyboard, mouse and joystick input. Thus, by using "pygame" module we are taking the input values from the controller. The input is taken in the form of an array of 17 elements.

Now the controller has two analog joysticks which send a value of -10 to 10 to the computer, other than this the push buttons are used to send either 1 or 0. From 17 elements for the array four are dedicated for analog inputs and the rest for push buttons. The data received is read and processed accordingly for controlling the robot's movements. The task to control the robots is supposed to be done using atmega328p chip. The data must be sent to the IC from the controller. This is done using serial communication. The best way to communicate between python and atmega IC is via a serial medium. The library used for this is "serial" library. This establishes a connection between the python and microcontroller over the comport it is attached to the computer. This microcontroller IC will be used to transmit the data to the robot.

The data from the python serial monitor is retrieved and send to the com port where the IC is connected to the computer. It is sent in the form of an array itself. This helps the user to assign each index for specific purpose. So as soon as the code starts running the data from the controller is directly sent to the microcontroller without any delay.

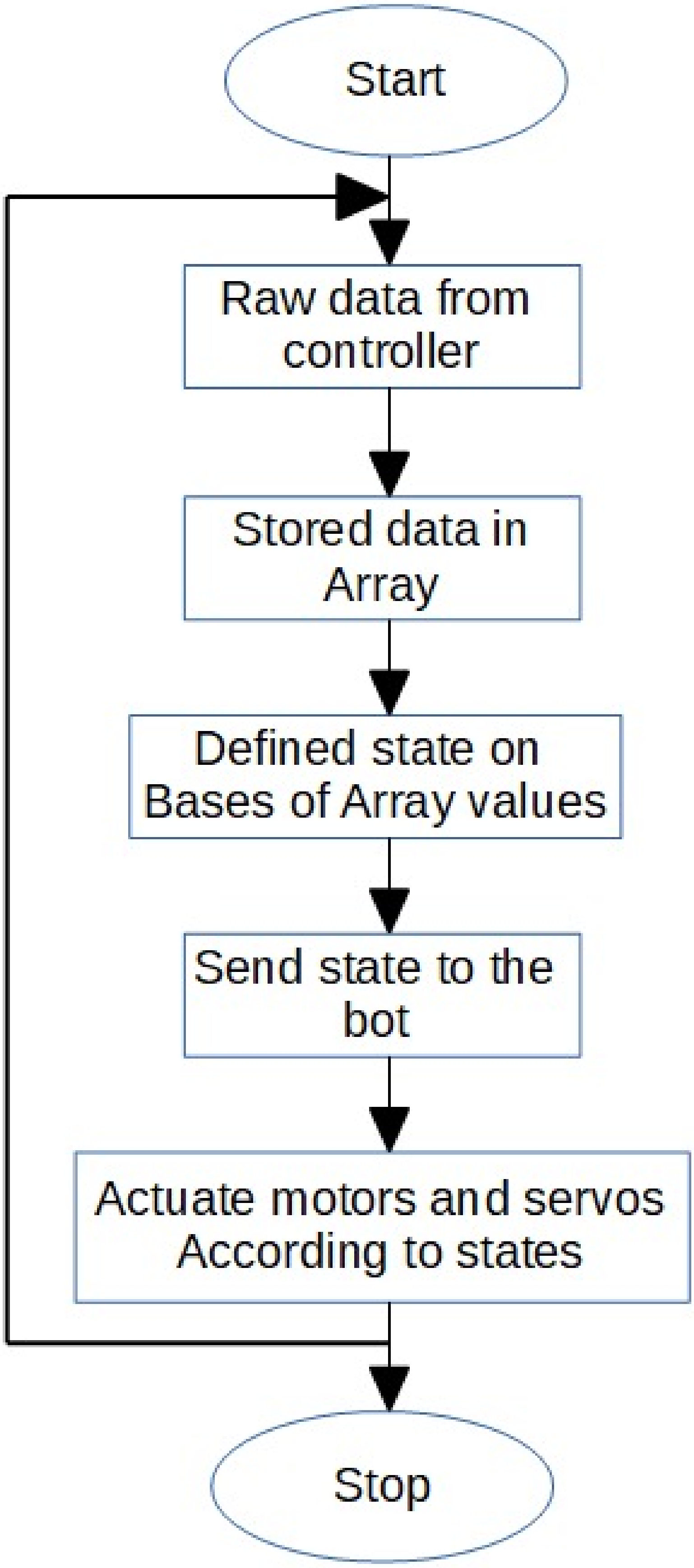

Figure 5.1: Code Flowchart

```python
import serial
import time
import pygame


pygame.init()
j = pygame.joystick.Joystick(0)
j.init()
print ('Initialized Joystick : %s' % j.get_name())

def get():
    out = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
    it = 0 #iterator
    pygame.event.pump()
    #Read input from the two joysticks
    for i in range(0, j.get_numaxes()):
        out[it] = round(j.get_axis(i),1)
        it+=1
    #Read input from buttons
    for i in range(0, j.get_numbuttons()):
        out[it] = j.get_button(i)
        it+=1
    if(abs(out[1])>abs(out[0])):
        out[0]=0.0
    else:
        out[1]=0.0
    if(out[1]==0.1 or out[1]==-0.1):
        out[1]=0.0
    if(out[0]==0.1 or out[0]==-0.1):
        out[0]=0.0
    if(out[3]==0.1 or out[3]==-0.1):
        out[3]=0.0
    if(out[4]==0.1 or out[4]==-0.1):
        out[4]=0.0
    out[1]=out[1]*10
    out[0]=out[0]*10
    out[3]=out[3]*10
    out[4]=out[4]*10
    #s=str(out).strip('[]')
    #data = pickle.dumps(s)
    #return data
    return out
```

```python
arduino=serial.Serial('COM4', 9600)
time.sleep(2)

print("Enter 1 to turn ON LED and 0 to turn OFF LED")

while 1:

    '''datafromUser=input()
    datafromUser=datafromUser+'\r'
    arduino.write(datafromUser.encode())  '''
    x = get()
    for i in range(0,17):
        x[i]=107+int(x[i])
        x[i]=chr(x[i])

    ch=''
    for i in range(0,17):
      ch=ch+(x[i])
    ch=ch+'\r'
    print(ch)
    arduino.write(ch.encode())
```

Figure 5.2: code for receiving user input

5.1.2 Serially receiving and decoding the input from the controller and sending it to bot

First, we need to include the SPI standard library for the serial peripheral interface with the **nrf-24l01** transceiver module. Then we have used an external library **RF24.h** for handling SPI communication and other functionalities of the transceiver module. Radio object of RF24 class is created with chip select and chip enable connected to the 7^{th} and 8^{th} pins of Arduino, respectively. The five bit address for transmission is set to "00001". The serial data is received in and array of 17 elements which indicates the state of different buttons of the controller

1. In the **setup section** first the radio is initialized for all the rf communication. Then the pipe is set up using **openWritingPipe()** function to write the data to a particular address. All the reading pipes are closed using the **stopListening()** function. Serial communication is initialized at 9600 baud rate.

2. Next, in the loop section, the received data is decoded and proper combinations are made for selecting the required action to be taken by the bot, which is referred to as particular states of the bot. There are in total 17 states of the bot which refers to particular actions it can perform when a combination of buttons of the controller is pressed.

3. After deciding the state ,the data is stored in the form of a number and sent via nrf-24l01 module using the write() function.

5.1.3 Receiving the state and actuating it using servo and dc motors

In this also we include the SPI standard library for the serial peripheral interface with the nrf-24l01 transceiver module of the Blue- tooth module as well as the servo library. Then we have used an external library RF24.h for handling SPI communication and other functionalities of the transceiver module. Radio object of RF24 class is created with chip select and chip enable connected to the 7^{th} and 8^{th} pins of Arduino, respectively. The five bit address for reception is set to "00001".

In addition Servo library is included and four servo instances are created and required global variables are declared

1. In the setup section first the radio is initialized for all the rf communication. Then the reading pipe is set up using **openReadingPipe()** function to receive data from a particular

Controller_to_arduinoFinal | Arduino 1.8.12 (Windows Store 1.8.33.0)

File Edit Sketch Tools Help

Save

Controller_to_arduinoFinal §

```
#include <SPI.h>
#include <RF24.h>

RF24 radio(7, 8); // CE, CSN
const byte address[6] ="00001";

//int datafromUser = 0;
int data;
int i = 0;
int d[17] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
/*initialise the pins to be used here

*/
int state=0;
void setup() {
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
  Serial.begin(9600);

}

void loop() {
  while(Serial.available() > 0) {
    if(i<15){
    data = Serial.read();
    d[i]=data;
    i++;
    }
    else{
      i=0;
      break;
```

```
Error downloading https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Boarc
Error downloading https://downloads.arduino.cc/packages/package_index.json
```

Controller_to_arduinoFinal §

```
    else if(d[11]==1&&d[9]==1){
      state=8;
    }
    else if(d[11]==1&&d[10]==1){
      state=9;
    }
     else if(d[11]==1){
      state=10;
    }
     else if(d[12]==1){
      state=11;
    }
    else if(d[1]>0){
      state=12;//wheels forward
      }
      else if(d[1]>0&&d[13]==1){
        state=13;//whegs forward
        }
        else if(d[2]>0){
      state=14;//wheels back
      }
      else if(d[2]>0&&d[13]==1){
        state=15;//whegs back
        }
        else if(d[3]>0){
      state=16;//wheels left
      }
         else if(d[4]>0){
      state=18;//wheels right
      }
      radio.write(state, sizeof(state));
    }

```

```
Error downloading https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Bc
Error downloading https://downloads.arduino.cc/packages/package_index.json
```

1

Figure 5.3: code for sending data to bot

address. All the writing pipes are closed using the **startListening()** function. Serial communication is initialized at 9600 baud rate. All the four servo instances are attached to the required pins of Arduino. We have used two motor controllers for controlling four dc motors for leg wheels and two dc motors for the whegs.Then all the servos are brought to the initial position

2. We have defined a custom function moveWheelsDC() to control the forward ,backward ,left and right motion of wheels as well as the whegs.

3. Next, in the loop section, dc motors and servo motors are actuated based on the received state which are described as follows:

   State 0: Reset all the servo motors by bringing them to the initial position

   State 1: Put the bot only on whegs by raising all the legs using servo motor

   State 2 to 5: Increment individual servos

   State 6 to 9: Decrement individual servos

   State 10: Increment all the servos together

   State 11:Decrement all the servos together

   State 12,14,16,18: Move the bot forward, back, left, right respectively

   State 15 and 17:Move the whegs front and back ardMotorControlRohan

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Servo.h>
RF24 radio(7, 8); // CE, CSN

const byte address[6] = "00001";
int state=0;
int i=0;
int pos=0;
Servo servolf;
Servo servorf;
Servo servolb;
Servo servorb;
int servolfp,servorfp,servolbp,servorbp;

/* declare all the pins to be used here
 *
 */
/*  STATES
   0-Arms down .All serrvo at 45 degree
   1-All arms up.All servo at 135 degrees
   2-Increment serrvo S1
   3-Increment serrvo S2
   4-Increment serrvo S3
   5-Increment serrvo S4
   6-Decrement servo s1
   7-Decrement servo s2
   8-Decrement servo s3
   9-Decrement servo s4
   10-Increment all servos
   11-Decrement all servos
   12-Wheels DC motor forward
```

ardMotorControlRohan

```
   19-Whegs DC motor right
   */
void setup() {
 Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_MIN);
  radio.startListening();
/* declare the functioning of pins here
 *
 */
 servolf.attach(2);
  servorf.attach(3);
  servolb.attach(4);
  servorb.attach(5);
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
  pinMode(14,OUTPUT);
  pinMode(15,OUTPUT);
  delay(100);
  int pos=0;
  servolf.write(pos);
  servorf.write(pos);
  servolb.write(pos);
  servorb.write(pos);
  servolfp=pos;servorfp=pos;servolbp=pos;servorbp=pos;
}



void moveWheelsDC(String motion){
```

ardMotorControlRohan

```


void moveWheelsDC(String motion){
  if(motion =="Forward")
  {
    digitalWrite(10,HIGH);
    digitalWrite(11,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(13,LOW);
Serial.println("Forward");

  }
  if(motion =="Back")
  {
    digitalWrite(10,LOW);
    digitalWrite(11,HIGH);
    digitalWrite(12,LOW);
    digitalWrite(13,HIGH);
Serial.println("Backward");


  }
  if(motion =="Right")
  {
    digitalWrite(10,LOW);
    digitalWrite(11,HIGH);
    digitalWrite(12,HIGH);
    digitalWrite(13,LOW);
Serial.println("Right");


  }
  if(motion =="Left")
```

ardMotorControlRohan

```

void loop() {
  if (radio.available()) {
    radio.read(&state, sizeof(state));
    //On the basis of various states control different motors
     if(state==0)
      {
           for(int pos=0;pos<45;pos=pos+2)
           {
           servolf.write(pos);
           servorf.write(pos);
           servolb.write(pos);
           servorb.write(pos);}
           servolfp=pos;servorfp=pos;servolbp=pos;servorbp=pos;
      }
    else if(state==1){
          for(int pos=0;pos<135;pos=pos+2)
           {
           servolf.write(pos);
           servorf.write(pos);
           servolb.write(pos);
           servorb.write(pos);}
          servolfp=pos;servorfp=pos;servolbp=pos;servorbp=pos;

      }
      else if(state==2){
        servolfp=servolfp+5;
        servolf.write(servolfp);
      }

      else if(state==3){
        servorfp=servorfp+5;
        servorf.write(servorfp);
```

Figure 5.4: Code for motor actuation

CHAPTER 6

CONCLUSION

1. Our model of multi terrain bot employs both movable arms as well as whegs mechanism which makes it suitable for almost any kind of rough terrain , unlike the previously existing models which generally employs either of the two.
2. Our model would be suitable for terrain with small obstacle like stones, stair like incline, terrain with depressions or mild holes and normal slopes.
3. Wheels attached to the arms can move the bot in forward ,backward, left and right directions ,while the whegs can clear the obstacle in forward and backward direction.
4. While encountering a rough surface the robot adjusts its structure by either raising or lowering the four arms to efficiently use both whegs and wheels.
5. At present these adjustments in the structure are made manually through a controller using RF communication which can be modified after proper testing and be made semi autonomous or even autonomous using certain sensors

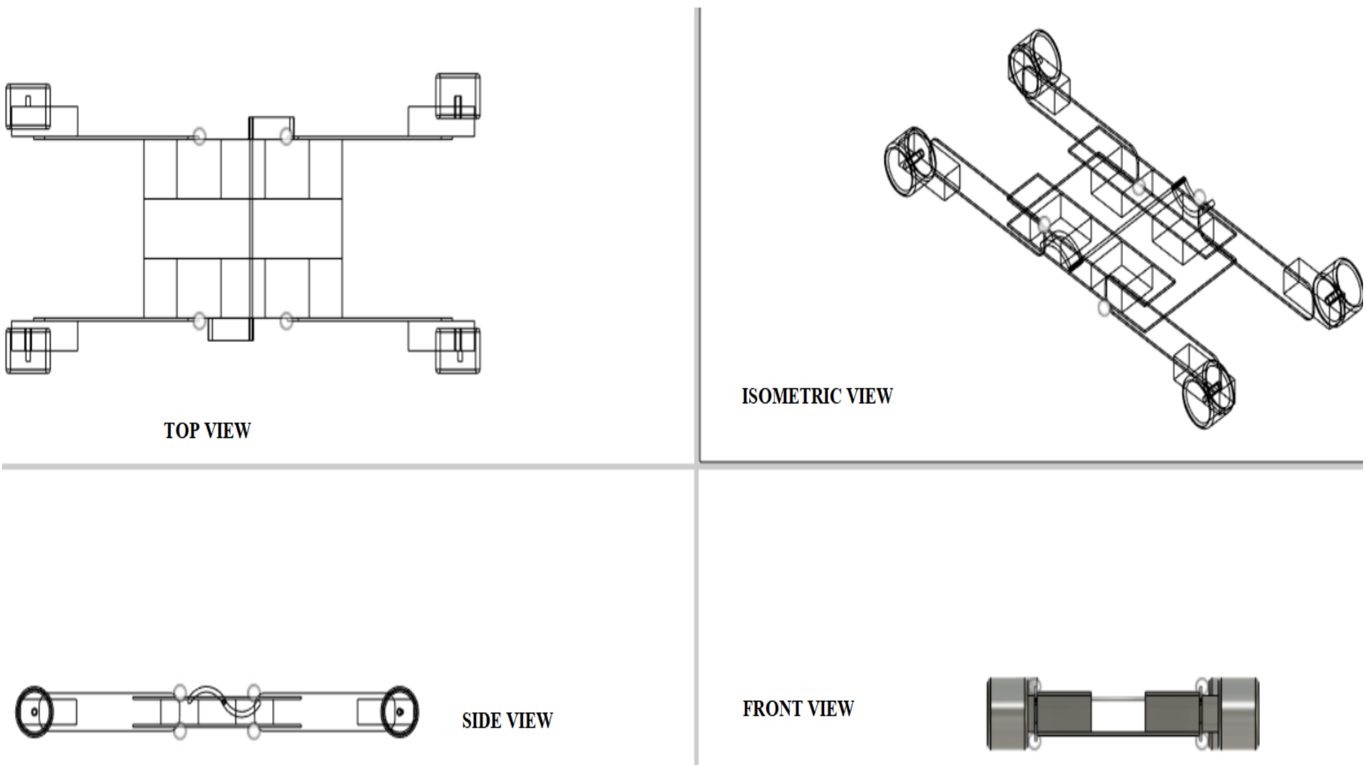

Figure 6.1: Different Views Of Design

CHAPTER 7

FUTURE ENHANCEMENT

1. This prototype can be implemented for vast area of exploration sciences and military purposes by replacing the current transceiver by a more efficient **Radio Frequency Technology** module with larger range of communication and also replacing servo motors by more efficient **stepper motors**

2. It can be made amphibious by protecting the housing cabinet and other modifications.

3. It can be made autonomous by feeding into GPS way-point pattern and using Artificial Intelligence.

4. In medical fields the bot can be modified into wheelchair designs.

5. In automobile industry , military vehicles.

APPENDIX A

MATERIALS

A.1 Hardware

1. Arduino Microcontroller:Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started

2. Polyacrylonitrile Acrylic Polymer/Aluminium Metal Sheets

3. nRF24L01 transceiver module:nRF24L01 transceiver module it uses the 2.4 GHz band and it can operate with baud rates from 250 kbps up to 2 Mbps. If used in open space and with lower baud rate its range can reach up to 100 meters.The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time.The power consumption of this module is just around 12mA during transmission, which is even lower than a single LED. The operating voltage of the module is from 1.9 to 3.6V, but the good thing is that the other pins tolerate 5V logic, so we can easily connect it to an Arduino without using any logic level converters.Three of these pins are for the SPI communication and they need to be connected to the SPI pins of the Arduino, but note that each Arduino board have different SPI pins. The pins CSN and CE can be connected to any digital pin of the Arduino board and they are used for setting the module in standby or active mode, as well as for switching between transmit or command mode. The last pin is an interrupt pin which doesn't have to be used.

4. L293D Motor Driver IC:L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. It works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. As you know voltage need to change its direction for being able to rotate the motor in clockwise or anticlockwise direction, hence H-bridge IC are ideal for driving a DC motor. In a single L293D chip there are two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors. There are two Enable pins on l293d. Pin 1 and pin 9, for being able to drive the motor, the pin 1 and 9 need to be high. For driving the motor with left H-bridge you need to enable pin 1 to high. And for right H-Bridge you need to make the pin 9 to high. If anyone of the either pin1 or pin9 goes low then the motor in the corresponding section will suspend

working. It's like a switch. VCC is the voltage that it needs for its own internal operation 5v; L293D will not use this voltage for driving the motor. For driving the motors it has a separate provision to provide motor supply VSS (V supply). L293d will use this to drive the motor. It means if you want to operate a motor at 9V then you need to provide a Supply of 9V across VSS Motor supply. The maximum voltage for VSS motor supply is 36V. It can supply a max current of 600mA per channel. Since it can drive motors Up to 36v hence you can drive pretty big motors with this l293d.

5. DC motor:The metal gears have better wear and tear properties. Gearbox is sealed and lubricated with lithium grease and requires no maintenance. Although motor gives 300 RPM at 12V, motor runs smoothly from 4V to 12V and gives the wide range of RPM, and torque. The shaft has a hole for better coupling.

6. Servomotors:The term servomotor does not refer to one single kind of motor. Instead it refers any type of motor that receives a command signal from a controller. In this same respect, any closed loop system can be referred to as a servo system.This flexibility allows for several suitable types of electric motors to be used in servo systems. These electric motors include:

   Permanent Magnet DC Motor

   Brushless DC Motor

   Induction AC Motor

   Electromagnetic motors operate based on the principle that the magnetic force on an electrical conductor in a magnetic field is perpendicular to that field. This force is defined by,

   F= qv*B

   Where:

   F is the vector describing the magnetic force

   q is the magnitude of the electrical charge

   v is the vector magnitude of the charged particles velocity

   B is the vector describing the magnetic field

   However, in the case of an electric motor the force can be quantified as a scalar

   F=I*L*B

   Where: F is the magnetic force

   I is the electric current in the coil

   L is the length of the coil contained within the magnetic field

   B is the strength of the magnetic field

A.2 Software

1. Arduino IDE:Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development

Environment) that runs on your computer, used to write and upload computer code to the physical board.

2. IDLE:IDLE is Python's Integrated Development and Learning Environment.IDLE has the following features:

    coded in 100% pure Python, using the tkinter GUI toolkit

    cross-platform: works mostly the same on Windows, Unix, and macOS

    Python shell window (interactive interpreter) with colorizing of code input, output, and error messages

    multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features

    search within any window, replace within editor windows, and search through multiple files (grep)

    debugger with persistent breakpoints, stepping, and viewing of global and local namespaces configuration, browsers, and other dialogs

3. Fusion360:Fusion360 is a solid modeling computer-aided design (CAD) and computer-aided engineering (CAE) computer program that runs on Microsoft Windows. Fusion360 is published by Autodesk.

REFERENCES

[1] Harkins, R., Ward, J., Vaidyanathan, R., Boxerbaum, A. X., and Quinn, R. D. (2005). "Design of an autonomous amphibious robot for surf zone operations: part ii - hardware, control implementation and simulation." *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 1465–1470.

[2] Koh, D. C., Dharmawan, A. G., Hariri, H. H., Soh, G. S., Foong, S., Bouffanais, R., Low, H. Y., and Wood, K. L. (2019). "Design and analysis of a miniature two-wheg climbing robot with robust internal and external transitioning capabilities." *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 9740–9746.

[3] Kousar, H., Atif, M., Ali, H., and Kundi, M. H. (2019). "Whegstr: A multiterrain robot with c-shaped whegs, implementation of error minimization technique and using artifical neural network (ann)." *2019 International Conference on Green and Human Information Technology (ICGHIT)*, IEEE, 139–141.

[4] Nakajima, S. (2009). "Development of four-wheel-type mobile robot for rough terrain and verification of its fundamental capability of moving on rough terrain." *2008 IEEE International Conference on Robotics and Biomimetics*, IEEE, 1968–1973.

[5] Naskar, S., Das, S., Seth, A. K., and Nath, A. (2011). "Application of radio frequency controlled intelligent military robot in defense." *2011 International Conference on Communication Systems and Network Technologies*, 396–401.

[6] Omijeh, B., Uhunmwangho, R., and Ehikhamenle, M. (2014). "Design analysis of a remote controlled pick and place robotic vehicle." *International Journal of Engineering Research and Development*, 10(5), 57–68.

[7] Schroer, R. T., Boggess, M. J., Bachmann, R. J., Quinn, R. D., and Ritzmann, R. E. (2004). "Comparing cockroach and whegs robot body motions." *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, Vol. 4, 3288–3293 Vol.4.

[8] Ward, J. L. (2005). "Design of a prototype autonomous amphibious whegs robot for surf-zone operations." Ph.D. thesis, Monterey, California. Naval Postgraduate School, Monterey, California. Naval Postgraduate School.

[9] Watson, J. T., Ritzmann, R. E., Zill, S. N., and Pollack, A. J. (2002). "Control of obstacle climbing in the cockroach, blaberus discoidalis. i. kinematics." *Journal of Comparative Physiology A*, 188(1), 39–53.

[10] Weihmann, T., Brun, P.-G., and Pycroft, E. (2017). "Speed dependent phase shifts and gait changes in cockroaches running on substrates of different slipperiness." *Frontiers in zoology*, 14(1), 54.

[11] Wile, G. D., Daltorio, K. A., Palmer, L. R., Witushynsky, T. C., Southard, L., Ahmad, M. R., Malek, A. A., Gorb, S. N., Boxerbaum, A. S., Ritzmann, R. E., and Quinn, R. D.

(2008). “Making orthogonal transitions with climbing mini-whegsâĎć.” *2008 IEEE International Conference on Robotics and Automation*, 1775–1776.