ENHANCEMENT OF A SEGMENTED HEXAPOD ROBOT

Report and Annexes

Author: Niveathitha Murugan
Raagul Tirupur Senthilkumar

Director: Jordi Cosp Vilella
Departament d'Enginyeria Electrònica (EEL)
Universitat Politècnica de Catalunya (UPC)

Call: June, 2019
INDEX

SUMMARY 4

ABSTRACT 8

ACKNOWLEDGMENTS 12

GLOSSARY 13

1. CHAPTER 1: INTRODUCTION 14

1.1. Objectives 15

1.2. Flaw Detection 15

1.2.1 Outdated Hardware 16

1.2.2 Power Regulation Circuit 16

1.2.3 Ultrasonic Sensor 17

1.3. Bio-Inspiration 17

1.4. Implementation 19

2. CHAPTER 2: PLATFORMS 22

2.1. Raspberry Pi 23
2.1.1 Raspberry Pi A+ V1.1 23
2.1.2 Raspberry Pi 3 B+ 26
2.1.3 Software Installation 28
2.2. Arduino Mini 29
2.3 Ultrasonic Sensor 31
2.4 Arduino Uno 33
2.5 Servo Motor 35

3. Chapter 3: Electronic Circuit 38

4. Chapter 4: Algorithm 41

4.1 Walking Gait 43
4.1.1 Wave Gait 44
4.1.2 Tripod Gait 45
4.1.3 Ripple Gait 46

5. Chapter 5: Performance Indices 48

6. Chapter 6: Programming 50

6.1 C Library for Broadcom BCM 2835 as used in Raspberry Pi 51
6.2 Program

6.2.1 Arduino Code 53

6.2.2 Raspberry Pi Code 54

7. CHAPTER 7: SCOPE 56

ANALYSIS OF THE ENVIRONMENTAL IMPACT 59

CONCLUSION 60

BIBLIOGRAPHY 61

ANNEXES 63

A1. ARDUINO CODE 63

A2. RASPBERRY PI CODE 65
SUMMARY

The main purpose of this project is to improve gait and movement of the Hexapod Bot. This project has been through various modifications by different sets of students who have improved the mechanics and electronics of the bot progressively. This robot consists of three modules which are joined to form the complete base platform to coordinate the six legs of the bot. It possesses two degrees of freedom per leg namely front, back, left, right.

The Raspberry Pi used by the previous batch- Raspberry Pi A+ V1.1 was an outdated version which has been replaced by one of the latest models- Raspberry Pi 3 B+ for attaining a better scope for the Hexapod Bot to undergo advanced changes in the near future. The power distribution has been the main issue faced which is not to be ignored and to be focused for further improvement of this bot. For temporary functioning of the bot, two DC power supplies of constant 5V-2A supply has been used as the previous circuit did not supply enough current for functioning of all the 12 motors simultaneously.

Since the Raspberry Pi serves as the Brain of the bot, the bot executes its orders through the Arduino Uno, a microcontroller platform used to control the movement of the bot based on the algorithm developed in such a way the gait of the bot is balanced smoothly. The electronic circuit designed to regulate the current for the motors to work efficiently was not achieved so a different circuit has been designed to make the robot move properly with the motors receiving proper current. The previous circuit was not stable and was sensitive to sudden variations in power supply. Sudden variations in current lead to damage of the circuit components. Hence, a new circuit was designed with two bypass capacitors to sustain the instability in the power supply. This circuit could not supply enough current to all the motors simultaneously. Thus, two DC power supplies
have been used. There is no issue with the variations in power supply as it is constant and will not affect any electronic component. The voltage is only 5V and will not affect any circuit.

Not only the gait and the movement is important for the Hexapod bot but also its efficient working according to the desired algorithm which is achieved through the Ultrasonic sensor to detect obstacles. By sensing the obstacles past the way, the bot makes the appropriate decision. Apart from this new sensors such as camera, power display and so on can be added to expand the applications.
El objetivo principal de este proyecto es mejorar la marcha y el movimiento del Bot Hexapod. Este proyecto ha sido objeto de varias modificaciones por parte de diferentes grupos de estudiantes que han mejorado progresivamente la mecánica y la electrónica del robot. Este robot consta de tres módulos que se unen para formar la plataforma base completa para coordinar las seis patas del robot. Posee dos grados de libertad por pata, es decir, la parte delantera, trasera, izquierda y derecha.

La Raspberry Pi utilizada por el lote anterior Raspberry Pi A + V1.1 fue una versión obsoleta que fue reemplazada por uno de los últimos modelos- Raspberry Pi 3 B+ para lograr un mejor alcance para que el Hexapod Bot experimente cambios avanzados en el futuro futuro. La distribución de energía ha sido el principal problema que se ha enfrentado, que no debe ignorarse y enfocarse para mejorar aún más este bot. Para el funcionamiento temporal del bot, se han utilizado dos fuentes de alimentación de CC de suministro constante de 5V-2A, ya que el circuito anterior no suministró suficiente corriente para que todos los 12 motores funcionaran simultáneamente.

Dado que la Raspberry Pi sirve como el cerebro del bot, el bot ejecuta sus órdenes a través del Arduino Uno, una plataforma de microcontrolador utilizada para controlar el movimiento del bot basado en el algoritmo desarrollado de tal manera que la marcha del bot se equilibra sin problemas.

El circuito electrónico diseñado para regular la corriente para que los motores funcionen de manera eficiente no se logró, por lo que se diseñó un circuito diferente para hacer que el robot se mueva correctamente con los motores que reciben la corriente adecuada. El circuito anterior no era estable y era sensible a las variaciones repentinas en fuente de alimentación. Las variaciones repentinas en la corriente provocan daños en los componentes del circuito. Por lo tanto, se diseñó
un nuevo circuito con dos capacitores de derivación para mantener la inestabilidad en la fuente de alimentación. Este circuito no pudo suministrar suficiente corriente a todos los motores simultáneamente. Por lo tanto, se han utilizado dos fuentes de alimentación de CC. No hay ningún problema con las variaciones en la fuente de alimentación, ya que es constante y no afectará a ningún componente electrónico. El voltaje es de solo 5 V y no afectará a ningún circuito.

No solo la marcha y el movimiento son importantes para el bot Hexapod, sino también su funcionamiento eficiente según el algoritmo deseado que se logra a través del sensor ultrasónico para detectar obstáculos. Al sentir los obstáculos más allá del camino, el robot toma la decisión apropiada. Además de estos nuevos sensores, como la cámara, la pantalla de alimentación, etc., se pueden agregar para expandir las aplicaciones.
ABSTRACT

The main objective of this project is to improve the gait and movement of the Hexapod robot, also to provide the required amount of power to the robot to function efficiently. Furthermore, improvements are executed through the Ultrasonic sensor to manage its movement in order to tackle any situations with obstacles in its path. Updating the electronics to the newer versions for an open scope to develop the project in the future is also a vital goal.

Improvisation of the gait and movement of the robot is achieved by altering the programming codes proposed by the previous set of students. Offsets for each leg segment have been calibrated to control the movement of the robot. Values have been updated according to the algorithm.

The right amount of power has to be supplied for enhancing the robot to have an efficient life in order to not damage the other platforms which act as the nervous system of the Hexapod robot. The nervous system as mentioned are denoted as the Raspberry Pi, Arduino mini, Servo motors and Power regulators. This system is to be circulated with proper current flow which is found to be a major flaw. The previously used electronic circuit does not withstand the current efficiently so a new circuit is developed to make this work properly without damaging any components of the robot.

Ultrasonic sensors are attached to the head of the robot in order to detect any obstacles in its way so that the algorithm defined program is initiated to do its work by making the robot take a turn either in right, left or backward direction. When an object is detected within the specified range, a signal is sent from the Raspberry Pi to the Arduino intimating it about the presence of an obstacle. This signal aids the Arduino in deciding the succeeding action of the robot.
Previously used platforms are found to be outdated which are to be replaced to make sure a scope for future updation. Raspberry Pi A+ V1.1, an old model of this platform has got its successor- Raspberry Pi 3 B+ replacing it. Due to the presence of inbuilt WiFi module and camera module, the Pi 3 B+ is a better solution for increasing the scope of the bot. The ultrasonic sensor can be replaced by a camera module for advanced applications.
L’objectiu principal d’aquest projecte és millorar la marxa i el moviment del robot Hexapod, també per proporcionar la quantitat necessària de potència al robot per funcionar de manera eficient. A més, les millores s’executen a través del sensor ultrasònic per gestionar el seu moviment per afrontar qualsevol situació amb obstacles al seu pas. L’actualització de l’electrònica a les versions més recents per a un àmbit obert per desenvolupar el projecte en el futur també és un objectiu vital.

La improvisació de la marxa i el moviment del robot s'aconsegueix modificant els codis de programació proposats pel conjunt anterior d’estudiants. Els desplaçaments per a cada segment de cama s'han calibrat per controlar el moviment del robot. Els valors s'han actualitzat segons l’algorisme.

S'ha de subministrar la quantitat adequada de potència per millorar el robot per tenir una vida eficient per no danyar les altres plataformes que actuen com a sistema nerviós del robot Hexàpode.

El sistema nerviós esmentat es denomina Raspberry Pi, Arduino reguladors mini, servomotors i potència. Aquest sistema es distribuirà amb un flux de corrent adequat i es troba com un defecte important. El circuit electrònic utilitzat prèviament no suporta la corrent de manera eficaç, de manera que es desenvolupa un nou circuit que fa que funcioni correctament sense danyar cap dels components del robot.

Els sensors d'ultrasons s'uneixen al capçal del robot per detectar els obstacles a la seva manera de fer que el programa definit per l'algorisme s'iniciï fent que el robot giri a la dreta, l'esquerra o cap a enrere. Quan un objecte es detecta dins del rang especificat; s’envia un senyal des de la Raspberry...
Pi fins a l’Arduino, que el fa notar sobre la presència d’un obstacle. Aquest senyal ajuda l’Arduino a decidir l’acció successiva del robot.

Les plataformes usades anteriorment s’han trobat obsoletes i han de ser substituïdes per assegurar-se que hi hagi un abast per a una actualització futura. Raspberry Pi A + V1.1, un model antic d’aquesta plataforma té el seu successor: Raspberry Pi 3 B + el substitueix. A causa de la presència de mòdul WiFi i mòdul de càmera incorporat, el Pi 3 B + és una millor solució per augmentar l'abast del bot. El sensor d'ultrasons es pot substituir per un mòdul de càmera per a aplicacions avançades.
ACKNOWLEDGEMENT

The final outcome of this project required a lot of supervision, guidance and assistance from our mentor. We are extremely privileged to have received this support throughout the completion of our project. For successful and timely completion of this project, we are grateful to our friends and family who have extended moral and financial support during this project.

First and foremost, we owe our deepest gratitude to our project guide, Prof. Jordi Vilella Cosp, Department Of Electronic Engineering, Polytechnic University of Catalunya (UPC) for providing us with the opportunity to work on this project and giving us complete support and guidance through his busy schedule which helped us produce the following results. We are grateful to him for providing us with the necessary information, suggestions and ideas and constantly inspiring us to pursue this particular topic.

We would like to extend our sincere thanks to Ms. Matilde Torne Ubeda, International Relations Office Escola d'Enginyeria de Barcelona Est. for providing us with this opportunity of Student Exchange and for guiding us from the beginning.

We would also like to thank Mr. Fernando and team at Lab Services for helping us throughout with the required electronic and technological components.

We are fortunate enough to get constant encouragement, support and guidance from all the staff of EEBE throughout this semester which has provided us with the strength to move forward.
GLOSSARY

bcm2835 - C library for Raspberry Pi which provides access to GPIO pins

SPI - Serial Peripheral Interface - common communication protocol used for two-way communication between two devices

Gait - manner of movement

Raspberry Pi - single board computer used for various development applications

Ultrasonic Sensor - acoustic sensor used to detect the presence of an obstacle

L7805 - 3 pin voltage regulator

Serial Communication - process of sending data one bit at a time

GPIO - General Purpose Input Output

Arduino UNO - a microcontroller board with digital and analog pins

PWM - Pulse Width Modulation

Bypass Capacitor - capacitor used to remove AC noise and produce pure DC
CHAPTER 1

INTRODUCTION
This project is the continuation of the work carried on by the previous set of students who had developed this Hexapod robot from the mechanical structure to the electronics. The sole idea of this project is to make the robot have its movement with the latest technology which could assist the robot into furthermore addition of sensors to improve its gait and movement.

1.1 Objectives

The main objective of this project is to improve gait and movement of the Hexapod robot but the robot was engaged with outdated hardware components. Secondly, communication has to be achieved between the Raspberry Pi and Arduino to propagate the signal from the Ultrasonic sensor. The power distribution was a main disadvantage for the successful working of the robot which is to be looked upon. Power regulators should be chosen in such a way that there is no power loss. Previously designed electronic circuit is witnessed to be not helping solve the issue of supplying appropriate current for the servo motors to work properly. To overcome this flaw, a new circuit is designed as the servo motors require proper supply of current to enhance properly. Along with this Ultrasonic sensor was used to detect obstacles which was not working after the mechanics of the robot was modified.

1.2 Flaw Detection

Applying power to the Hexapod robot in the first case, it is witnessed to be not moving. The robot was able to stand on its legs but did not make any further movement. The program was debugged for a period of time but that did not help. The robot was dismantled to check the circuit behind the power distribution. The circuit was soldered perfectly with the respective components as per the design of the previous batch. Examining the connections, the problem was detected. Connections
made through jumper cables were not functioning as it lost its connectivity. Once all the connecting wires were examined, all the damaged jumper cables were replaced with proper set of wires, yet the trail was a failure. Moving on with the Program expecting to find the error, accessing the Raspberry Pi was very difficult. It had only Bluetooth access which was a huge turn down situation as few library files could not be downloaded or sent over Bluetooth. The older version of the Raspberry Pi which was used previously did not have a built in WiFi adapter and the connection to the internet was not working. Moving on to the Ultrasonic sensor, the distance was not calculated. The primary flaws are jotted as above.

1.2.1 Outdated Hardware

The previously used Raspberry Pi was Raspberry Pi A+ V1.1 which had many drawbacks compared to the available features for the latest models of the same. It did not support the Bluetooth feature as well. This was tackled by replacing the old version with one of the latest versions, that is Raspberry Pi 3 B+ model. This version has all the latest features such as WiFi, increased number of Ports and so on which is discussed later.

1.2.2 Power Regulation Circuit

The main purpose of a power regulation circuit is to provide the required amount of power for the respective components but this was not achieved in the previous work. The power requirements of the various components in the circuit are widely different. The main issue faced is the supply of adequate power supply to the Servo motors as they require a large amount of current compared to the rest. A circuit is required which can provide a minimum voltage of 5V as the motor requires a supply of 4.8V-6V and a current of 1.5A for the motor to work efficiently. Due to excessive
variations in the supply while testing, the circuit was damaged. A test circuit was designed with a L7805CV regulator and two bypass capacitors to check the output current. This current was enough for a single Servo motor to work but could not power up twelve motors simultaneously. Hence, two DC power supplies of 5V-2A constant power supply are used. This current was enough for all the motors to work efficiently.

1.2.3 Ultrasonic Sensor

Detection of obstacle was not functioning properly as there was a discontinuity in one of the wires used to connect the Ultrasonic sensor. This was rectified by changing the connections of the ultrasonic sensor to the new Raspberry Pi. The codes were altered and the distance was calculated. The signal from the Ultrasonic sensor, i.e., the distance from the obstacle is sent from the Raspberry Pi to the Arduino through serial communication. With these changes this issue has been solved successfully. The working of the sensor is explained later.

1.3 Bio-Inspiration

The design of the robot is biologically inspired from the class of animals called Hexapoda embodied with a segmented body, each comprising of a set of legs for locomotion and the brain controls the actions of the entire body. Their body is categorized into an anterior head, thorax and posterior abdomen and the thorax consists of three segments each bearing a pair of legs.
The analogy between an insect and the Hexapod robot is mapped below for comparison.

The design of the Hexapod also depends on the symmetry followed. There exists two common types of symmetry in biology - Radial Symmetry and Bilateral Symmetry. In radial symmetry, the body cannot be differentiated into left and right while this is possible in bilateral symmetry. There exists a comprehensible line of symmetry in bilateral symmetry that divides the body into left and right symmetrical segments. The body parts are arranged equally on both sides and are mirror images of each other. In radial symmetry, the distribution of equal body parts are found along the
central axis of the body. Since a segmented Hexapod has been developed, bilateral symmetry has been administered.

Two different designs of Hexapod robots are shown above based on the symmetry.

1.4 Implementation

![Figure 1.4 Outline of Hexapod Robot](image)
As discussed the Raspberry Pi acts as the brain of the robot, the three other Arduino mini platforms will be acting as the organs receiving functional orders that is, the Raspberry Pi will be sending orders in the form of signals to the three individual segments controlled by the Arduino mini platform. This is done to achieve the feat to work in a parallel way not affecting each other in any way. The only common feature that will be shared between all the platforms containing segments is the Power supply. This has been the primary issue for the Hexapod robot. To improve the movement and gait of the robot, a single segment of the Hexapod robot was removed to be studied as the bot was not in a working condition. Working on the new electronic circuit, the removed segment controlled by an Arduino Uno and the Raspberry Pi was put under test using the developed code. This experiment produced positive results. But the power supply was not enough for running 12 motors simultaneously. Hence two DC power supplies have been used. The Arduino Uno was used to test the movement of the entire Hexapod. The same code can be inherited by the Arduino mini platform. The code in Raspberry Pi is to calculate the distance of the obstacle using the Ultrasonic sensor and to communicate the same to the Arduino Uno through serial communication. The robot is signalled to stop if an obstacle is detected. This function has to be extended to make the robot take turns based on the presence of the obstacle.

![Single segment of Hexapod robot](image)

**Figure 1.5 Single segment of Hexapod robot**
Figure 1.6 Initial offset position of Hexapod Robot

Figure 1.7 Swing Phase 1 of Hexapod Robot

Figure 1.8 Swing Phase 2 of Hexapod Robot
CHAPTER 2

PLATFORM
2.1 Raspberry Pi

Raspberry Pi is a mini computer which has its own OS namely the Raspbian. It can be connected to Monitor for the display while it has separate ports for the keyboard and mouse. This mini computer has gone under various upgrades with new features added in each of its new model over time to maintain the updation of technology.

2.1.1 Raspberry Pi A+ V1.1

This model of Raspberry was released in the year 2014. It is used as a single board computer to carry on small tasks according to the user. It is widely used in IoT and home automation applications. It should be powered by a power source in order to activate the OS to work on it. The Raspberry Pi runs on the Raspbian OS which is a Debian-based Linux distribution available for download on the official Raspberry Pi Foundation website. The OS also supports IDEs like Arduino IDE which can be used for programming the Arduino. The editor Geany has been used to program the Raspberry Pi as it supports both Python and C/C++ language. The program is written in C for easy debugging and understanding.

Figure 2.1 Raspberry Pi A+ Version 1.1
The specifications are as follows:

**SOC:**

**SOC Type:** Broadcom BCM2835

**Core Type:** ARM1176JZF-S

**No. of Cores:** 1

**GPU:** VideoCore IV

**CPU Clock:** 70Mhz

**RAM:** 256 MB
Wired Connectivity:
USB Ports- 1
Ethernet- Nil
SATA Ports-Nil
HDMI port- 1
Analog Video Out- 1
Analog Audio In- Nil
Analog Audio Out- 3.5mm jack
SPI- Yes
I2C- Yes
GPIO- Yes
LCD Panel- Yes
Camera- Yes
SD/MMC- microSD
Serial- No

Wireless Connectivity(On Board):
Wi-Fi- Nil
Bluetooth- Nil

Power:
Power ratings- 200 mA
Power sources- microUSB or GPIO
Power over Ethernet- Nil
2.1.2 Raspberry Pi 3 B+

This model of Raspberry was released in the year 2018. It is one of the latest version of Raspberry Pi. It has been updated with most of the required features used as a computer. The OS- Raspbian has got its newer look which is similar to a normal computer. It has dual band WiFi support and an accelerated CPU. It possesses four USB 2.0 ports, the RJ45 port for Ethernet connections, the headphone jack, the HDMI connector, the Micro USB port for power or the already classic GPIO port. It supports Bluetooth 4.2 compared to 4.1 that was supported in the previous Raspberry Pi 3 Model B. Power-over-Ethernet (PoE) is also supported.

Figure 2.3 Raspberry Pi 3 B+
The specifications are as follows:

**SOC:**

SOC Type- Broadcom BCM2837B0

Core Type- Cortex- A53 64-bit

No. of Cores- 4

GPU- VideoCore IV

CPU Clock- 1.4Ghz

RAM- 1GB DDR2

**Wired Connectivity:**

USB Ports- 4xUSB 2.0

Ethernet- Gigabit- Over USB 2.0

HDMI port- 1

Analog Video Out- Shared with audio jack
Analog Audio In- Nil
Analog Audio Out- 3.5mm jack
SPI- Yes
I2C- Yes
GPIO- 40 pins
LCD Panel- Yes
Camera- Yes
SD/MMC- microSD
Serial- RX/TX UART

Wireless Connectivity(On Board):
Wi-Fi- 2.4GHz and 5GHz 802.11 b/g/n/ac
Bluetooth- 4.2, BLE

Power:
Power ratings- 1.13 A@5V
Power sources- microUSB or GPIO
Power over Ethernet- with PoE Hat

2.1.3 Software Installation

The OS for Raspberry Pi is famously known as the Raspbian. In order to work on a Raspberry Pi platform, this OS should be installed. The basic procedures are listed below:

1. Ensure that SD card( atleast 8GB) and Card reader are available.
2. Insert the SD card into the Card reader and log into a PC.
3. Format the SD card.
4. In order to install the Raspbian, the software called NOOBS, an Operating System Installer should be installed first.


6. Extract the files from NOOBS and paste it into the formatted SD card and eject it.

7. Insert the SD card into the slot provided in the Raspberry Pi.

8. Connect the Mouse and Keyboard through the available USB ports.

9. Using a HDMI cable, establish a connection for the display through a monitor.

10. If the internet access is provided through an Ethernet cable then connect the cable to the respective port else connect the WiFi once the Raspberry Pi is turned ON.

11. Make sure all the above are done before starting the Raspberry Pi.

12. Use a micro USB power supply to power up the platform.

13. The Welcome to Raspberry Pi application will pop up with few options regarding Language, Country, Timezone, Password, WiFi connection (ignore this if Ethernet is used) and Software Update (to verify if it is the latest version).

14. Click on “Done” once the above process is done successfully.

15. There will be a pop up screen from NOOBS asking which OS is to be installed.

16. Select Raspbian and click “Install”

17. Once the OS is installed, start working on the preferred Application based on the project.

2.2 Arduino Mini

Arduino Mini is a microcontroller board which can be used to perform small tasks based on the codes used to define an algorithm. This microcontroller has 14 digital input/output pins out of
which 6 pins of PWM outputs, 6 pins of analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. It runs on the software called Arduino IDE where one can convert their algorithms into codes to witness the expected output.

Figure 2.5 Arduino Mini

Figure 2.6 Arduino Mini pinout diagram

The specifications are as follows:
2.3 Ultrasonic Sensor

The Ultrasonic sensor used in this Hexapod robot is the model- HC-SR04. It uses SONAR to calculate the distance of any obstacle. It has an ultrasonic transmitter and receiver which are used to transmit and receive signals where ultrasound is emitted at 40kHz. It can detect objects ranging within a distance of 2 cm to 400 cm. This sensor has four terminals that are +5V, trigger, Echo and Ground. The Vcc pin powers up the sensor. Trigger pin is an input pin which is to be kept high for 10us to initialize measurement by sending ultrasonic wave. Echo pin is an output pin which is to be kept
high to receive the reflected wave. The ground pin is to be connected to the common ground of the circuit.

![Image of HC-SR04 ultrasonic sensor]

**Figure 2.7 Ultrasonic sensor- HC-SR04**

The specifications of this sensor are as follows:

**Power Supply:** +5V DC

**Quiescent Current:** <2mA

**Working Current:** 15mA

**Effectual Angle:** <15°

**Ranging Distance:** 2cm – 400 cm or 1" – 13ft

**Resolution:** 0.3 cm

**Measuring Angle:** 30 degree

The Ultrasonic sensor consists of a transmitter and a receiver. The transmitter sends an Ultrasonic wave which when blocked by an object within its range gets reflected back. The time taken for the wave to propagate from the transmitter and back to the receiver is used to calculate the distance at which the object is located. The formula used for this is Distance=Speed x Time. The speed of Ultrasonic sound wave is known to be 330m/s under standard room conditions. The time is calculated by the inbuilt circuitry using the time for which the echo pin is high. Using these two parameters, the distance at which
the obstacle is located is calculated. The final distance is calculated using the following formula:

\[
\text{Distance} = 0.5 \times \text{Sonic speed} \times \text{Time between emission and reception.}
\]

![Figure 2.8 Emission and Reception of Ultrasonic Signal](image)

### 2.4 Arduino UNO

Arduino Uno is a microcontroller used for specific tasks. It has 14 digital input or output pins. Out of the 14 pins, 6 pins can be used as PWM outputs, 6 pins as analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. This microcontroller can be connected to a computer using a USB cable or power it with a battery. This model is supposed to be the first version of all the Arduino platform versions with USB ports.
The specifications are as follows:

**Microcontroller**- ATmega328

**Operating Voltage**- 5V

**Input Voltage**- 7-12V
Enhancement of a Segmented Hexapod Robot

Input Voltage (limit)- 6-20V

Digital I/O Pins- 14 (6 pins provide PWM output)

PWM Digital I/O Pins- 6

Analog Input Pins- 6

DC Current per I/O Pin- 20 mA

DC Current for 3.3V Pin- 50 mA

Flash Memory- 32 KB (0.5 KB of that is used by bootloader)

SRAM- 2 kB

EEPROM- 1 kB

Clock Speed- 16 MHz

LED_BUILTIN- 13

2.5 Servo motor

An electronic device which can be used to rotate at a particular angle or be pushed to a desired distance using either an AC or DC power supply. In this project we have used DC power supply. All servo motors work directly with +5V supply but require a current of more than 1.5A to perform efficiently. A high amount of torque can be produced in these motors. To be specific, the motors used in this project is called as Positional rotation motor and the model used is RC Tecnic S3006. The output shaft has the ability to rotate around 180 degrees. It can turn 90 degree from either direction from its neutral position. In order to maintain this range, physical stops are located in the gear mechanism to keep a check on them. Twelve motors are used in this project as there are three modules which will be used to move. The working principle of Servo motor are as follows:
These motors use a Closed loop system with a positive feedback system. This can be understood from the following diagram. Closed loop system is called so as it takes a part of the output signal and adds it to the input signal as part of the loop. If the output signal bit taken is subtracted, then it becomes a negative feedback system. These signals tend to control the motion and final position of the shaft.

![Closed Loop System Diagram](image)

**Figure 2.11 Closed Loop System**

A servo motor consists of a motor (DC), a potentiometer, gear assembly, and a controlling circuit. The gear assembly is used to reduce RPM and increase torque.

Each motor consists of three wires that is for positive and negative terminal of supply and to provide the input signal.

Servo motor is controlled by PWM which is provided by the control wires. They produce a minimum pulse, a maximum pulse, and a repetition rate. It is expected to deliver a pulse every 20 milliseconds (ms) and the length of the pulse will let us know the angle rotated.

The angle of rotation is controlled by the duration of applied pulse to pin which provides input which is called a Control Pin.
Servo checks the pulse in every 20 milliseconds. Pulse of 1 ms (1 millisecond) width can rotate servo to 0 degrees, 1.5 ms can rotate 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degrees. The figure below explains in detail.

![Figure 2.12 Pulse Width Modulation of Servo motor](image)

Figure 2.12 Pulse Width Modulation of Servo motor
CHAPTER 3

ELECTRONIC CIRCUIT
Initially, the robot consists of three segments each containing a power shield circuit, an Arduino mini and two sets of legs with four servo motors. There was also a Raspberry Pi A+ which acted as the brain of the bot. The Ultrasonic sensor was connected to the Raspberry Pi. The Raspberry Pi detected the presence of any obstacle in front of the bot and sent appropriate signals for movement to the Arduino mini in each segment accordingly. The power shield circuit consisted of a L7805 regulator for each motor to regulate the voltage and also a Traco Power TSR-0.5-2490 to regulate the power supply to the Arduino mini as it cannot withstand high current.

Figure 3.1 Schematic diagram of Hexapod Robot
This is the circuit designed for temporary testing purposes. Bypass capacitors have been added with the regulator to stabilise the supply in case of surge or peaks. But this circuit could not power up multiple motors simultaneously. Two DC power supplies of constant 5V-2A is used to fulfil the requirement of current. An Arduino Uno is used instead of a mini for temporary testing purposes due to its flexibility and ease of use. Once a circuit for appropriate power supply is designed an Arduino mini can be used as it is compact. This circuit can be replicated for all the three segments.
CHAPTER 4:

ALGORITHM
The movement algorithm is designed based on the body and the six end points of the legs. It can be body-driven or leg-based depending on the focus. There are multiple complexities faced by the bot while traversing uneven terrain such as foot placement, load distribution, obstacle avoidance and general stability which affects the development of mechanical structure and movement strategies.

The algorithm is framed with the concept of Support Polygon which is described in brief below. It takes into account the center of mass of the robot, i.e., the point at which weighted relative position of the distributed mass sums to zero. The support polygon represents the necessary conditions for the body to be at equilibrium under the force of gravity. The center of mass must lie within the support polygon to achieve static stability. In this case, the forces acting over the region of contact will counteract the force of gravity which is a necessary condition for stability. The structure formed when the legs of the robot touch the ground forms the shape of the support polygon. Triangles are the most supportive polygons. For the Hexapod, supportive polygon is two triangles as mentioned below.

![Figure 4.1 Support Polygon](image-url)
4.1 Walking gait

Gait refers to the repetitive motion of the legs of a person or a robot which gives rise to locomotion. There exist three approaches, inspired from nature, to determine the pattern of movement of legs of the Hexapod robot. Based on the requirement of motion, the gait can be chosen between wave, tripod and ripple. There can also be a transition between the gaits during the motion of the robot which would increase the efficiency. Periodic gaits require synchronisation of movement between the legs.

Movement of each limb consists of two phases: the stance phase and the swing phase. Stance phase is when the limb is in contact with the walking surface and swing phase is when the limb is in the air. A continuous cycle of alternative swing and stance phases are required to process movement. During the swing phase in forward movement, the limb is retracted back and this is
known as power stroke or support stroke. In the stance phase, the limb moves towards the front of the body thus inducing forward movement. This is known as return stroke or recovery phase. A combination of these give rise to change in position.

4.1.1 Wave Gait

The gait that maximizes stability is the wave gait. In this pattern, each of the legs move individually, one at a time, while the other five provide support, in a wave pattern to slowly move the bot forward. It starts from the rear leg and moves forward to the middle and the front leg of the same side and continued on to the opposite side. It is most suitable for an uneven terrain. The disadvantage being that it is the slowest mechanism as only one leg moves at a time. It is the most stable algorithm as it requires only one leg to move at a time and all the other legs are present on the ground providing stability for the entire body of the bot. Although it is the best mechanism to achieve maximum stability, it does not provide effective speed.

![Figure 4.3 Wave gait pattern of movement](image)
4.1.2 Tripod Gait

The most efficient gait has been discovered to be the tripod gait as it is fast and easy. It supports all terrains thereby increasing the scope of the applications. In the tripod gait, the front and back leg of one side and the middle leg of the other side moving together and thus the name tripod. While one set of legs is swing phase the other three (tripod) are in stance phase. Keeping the step size constant makes the gait faster than the others.

Figure 4.3 Wave Gait

<table>
<thead>
<tr>
<th>L3</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>L2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.1.3 Ripple Gait

The ripple gait, although complex, supports a combination of speed and stability. Each side of the robot consists of a local wave of non-overlapping swing phases while the opposite side is also a
local wave which is exactly out of phase with respect to it. These are the three gaits commonly found in nature. It is also possible to come up with new ideas which are comparatively efficient.

Figure 4.7 Ripple gait pattern of movement

Figure 4.8 Ripple Gait
CHAPTER 5

PERFORMANCE INDICES
Performance index helps in calculating the efficiency of the design to meet the motive of the project. Performance indices are used to compare various designs with different shapes, sizes, and masses. The factors used for such comparison are mentioned below:

- **Stability Margin:**

  The stability margin for a dynamic robot is defined as the minimum distance of the ZMP to the boundaries of the support polygon, since the ZMP is the natural extension of a projected CoM on the ground. The minimum potential energy required to tumble the robot is the energy stability margin for a legged robot on a rough terrain.

- **Duty Factor:**

  Duty factor is calculated by dividing the support period by the total cycle time. It determines whether the bot is in walking or running cycle. The value of duty factor greater than or equal to 0.5 indicates walking and less than 0.5 for running.

  \[ \beta = \frac{\text{support period}}{\text{cycle time}} \]

- **Froude Number:**

  Froude number is calculated using the formula

  \[ F_{\text{Fr}2} = \frac{V^2}{gh} \]

  where \( V \) is the walking or running speed, \( g \) is the acceleration due to gravity, and \( h \) is the height of hip joint from the ground. \( V \) is a characteristic speed of the motion.

- **Specific Resistance:**

  The specific resistance is a dimensionless quantity that is used to evaluate the energy efficiency of a robot.
CHAPTER 6

PROGRAMMING
C language was chosen due to its benefits over other programming languages. As a middle level language, C combines the features of both high level and low level languages. It can be used for low-level programming, such as scripting for drivers and kernels and it also supports functions of high level programming languages, such as scripting for software applications. It is a structured programming language which allows a complex program to be broken into simpler programs called functions. C is highly portable and is used for scripting system applications which form a major part of Windows, UNIX and Linux operating systems. C language has a rich library which provides a number of built-in functions. It also offers dynamic memory allocation.

6.1 C library for Broadcom BCM 2835 as used in Raspberry Pi

Broadcom BCM 2835 library is the C library used for Raspberry Pi. It provides access to GPIO pins on the IDE plug on the RPi board which gives control and interface with various external devices and other IO functions on the Broadcom chip in the Raspberry Pi. It provides functions for reading digital inputs and setting digital outputs, using SPI and I2C and for accessing the system timer. Pin event detection is supported by polling (interrupts are not supported). It is C++ compatible, and installs as a header file and non-shared library on any Linux-based distro. The bcm2835_spi_* functions allow you to control the BCM 2835 SPI0 interface, allowing you to send and receive data by SPI (Serial Peripheral Interface). When `bcm2835_spi_begin()` is called it changes the behaviour of the SPI interface pins from their default GPIO behaviour in order to support SPI. While SPI is in use, there is no control over the state of the SPI pins through the usual bcm2835_spi_gpio_write(). When `bcm2835_spi_end()` is called, the SPI pins will all revert to inputs, and can then be
configured and controlled with the usual bcm2835_gpio_* calls. The bcm2835_aux_spi_* functions allows to control the BCM 2835 SPI1 interface, allowing us to send and receive data by SPI (Serial Peripheral Interface). In order for bcm2835 library SPI to work, it is important to disable the SPI kernel module using:

```bash
sudo raspi-config
```

under Advanced Options - enable Device Tree

under Advanced Options - disable SPI

Reboot.

This is the procedure which was followed previously. Though all programming languages are directly or indirectly derived from the C Programming language, each one possesses its own advantages and disadvantages over it depending on the application. Since it is a middle-level language, it does not provide all the built-in function which can be found in high-level languages. For extended applications, coding in C would make the program more complex and incomprehensible. Therefore, Python has been adapted to code the Raspberry Pi for ease of understanding. Python has a simple syntax and a variety of built-in functions which helps in saving time and space. It is very similar to C and that makes it easier to comprehend. Although, indentation has to be taken care of in Python.
6.2 Program

6.2.1 Arduino Code

The program for the Arduino is written in the Arduino IDE platform. Servo library <Servo.h> is used to control the Servo motors. The library supports up to 12 motors on a single Arduino board. The functions in the Servo library include the following:

- attach() - to attach the variable of the servo to the corresponding pin on the Arduino
- write() - to write a value to the servo, moving the gear to the respective position
- read() - to read the current angle of the gear of the servo
- detach() - to remove the servo variable from the respective pin
- attached() - returns TRUE or FALSE depending on whether the servo is attached to the pin or not
- writemicroseconds() - to write a value to the servo in microseconds directly instead of the angle to be specified

```cpp
Servo servo_3;
Servo servo_3; // This is to set the variable servo_3 as a Servo pin
```

```cpp
Serial.begin(9600);
Serial.begin(9600); // This is to set the data or baud rate in bits per second to setup serial communication
```

```cpp
delay(1000);
delay(1000); // This is used to pause the running of the program for the specified time (in ms)
```

```cpp
if (Serial.available())
if (Serial.available()) // This is used to check if there is any data to be received through serial communication
```

```cpp
a=Serial.readStringUntil(\'\n\');
a=Serial.readStringUntil(\'\n\'); // This is used to read data from serial buffer to a string
```
6.2.2 Raspberry Pi Code

The pre-installed Thonny editor on the Raspbian OS is used to run the Python code for serial communication as it is user-friendly and makes it easy to debug. It has buttons for Run, Debug, Stop, etc as seen below which makes it easy to navigate.

![Figure 6.1 Layout of Thonny Editor](image)

The code to measure distance as explained earlier using an Ultrasonic sensor is incorporated into this program. The signal from the Ultrasonic sensor is received and the distance is calculated accordingly. If the obstacle distance is within the specified range, a specific signal is sent to the Arduino to halt the bot and if the obstacle is out of range then a signal is sent to Arduino to make the robot move.
The package RPi.GPIO is imported to control the GPIO pins on the Raspberry Pi. Other necessary libraries such as time, math, and serial are also imported to use several forthcoming functions.

**GPIO.setmode(GPIO.BCM)** - This is used to set the board to BCM mode

**GPIO.setup(GPIO_TRIGGER, GPIO.OUT)** - This is to set the specified GPIO pin to input or output configuration

```python
ser = serial.Serial ('/dev/serial1', 9600)  # This is to open the serial port and to set the baud rate to 9600
```

**ser.flushInput()** - This is used to flush the serial buffer

```python
ser.write('STOP'.encode())  # This is to send the string “STOP” to Arduino UNO through the serial port
```

**GPIO.cleanup()** - This is used to clean up all the ports used in the program
CHAPTER 7

SCOPE
Enhancement of a Segmented Hexapod Robot

The negative aspects of the project can be overcome by making a few modifications or by improving several mechanisms or electronics. Major disadvantages and possible changes are explained in brief below to enhance the deliverables of the Hexapod project.

- The sensing accuracy of the Ultrasonic sensor is affected by the type of material and the temperature of the obstacle. Objects covered in a very soft fabric absorb more sound waves making it hard for the sensor to see the target. It has difficulties in reading reflections from soft, curved, thin, and small objects.

- Ultrasonic sensor has a limited detection range, the maximum range being 10 meters and only in one direction as it cannot move or turn. If the obstacle is detected sooner, the bot can be given time for transition of gait if needed.

- A stable circuit is required for ample power regulation throughout the electronic circuit of the Hexapod as the various electronic components like the servo motors, Arduino, and Raspberry Pi require different voltage and current. The present circuit is not efficient as it limits the running time of the Hexapod robot.

- No communication link between the Raspberry Pi and the PC. No manual interference possible during the working of the robot.

Suggested ideas for improvement:

- A servo motor can be added for the ultrasonic sensor so that it can rotate 360 degrees for surveillance around the bot.

- A mobile application can be used to control the bot, through Bluetooth or WiFi as the Pi already contains the required module, when the requirement arises.
- The ultrasonic sensor can be replaced with a camera for enhancing obstacle detection. As Raspberry Pi 3 is used and it supports a camera module which can be attached to it directly. Image processing can be used to detect the object. This way the scope and applications of the bot can be expanded. 3D mapping of the surroundings can be achieved through this. This would enable the bot to be used in surveillance.

- As the bot can surface in any terrain, with the addition of a camera it can be used for military and surveillance purposes in places where human interaction may be dangerous. The data can be transmitted through a radio-link for monitoring purposes. The bot can also be controlled via radio-link if necessary.

- Appropriate regulators and power supply are a prerequisite for the robot to work continuously for a longer duration.

- Instead of sticking to a single gait throughout the locomotion, a transition between the gaits can be used based on the speed of motion to achieve stability, speed or efficiency based on the requirement.

- If it is presented with a fully functioning camera, it can also be used for disaster management. Surveillance after a disaster like an earthquake has occurred.
Enhancement of a Segmented Hexapod Robot

ANALYSIS OF

ENVIRONMENTAL IMPACT

Lithium-polymer (Li-poly) batteries have become dominant in consumer electronic products due to the advantages associated with energy density and product longevity. The lack of regulatory policy on their disposal implies that lithium batteries contribute substantially to environmental pollution and adverse human health impacts due to potentially toxic materials. If not disposed properly, over discharged or punctured by accident, they may burst out in flames. Overcharging the LiPo may also cause fire. Using a LiPo-safe case may prevent this. The following steps have to be followed while disposing the battery. The battery must be discharged until its voltage reaches 1V per cell or lower. The battery should be submerged into a bucket, with a lid, filled with salt water (but not air-tight). It should be allowed to remain in the bucket for at least 2 weeks. The LiPo battery should be removed from the salt water and wrapped in newspaper or paper towels and placed in the normal trash as they are landfill safe. The material used for 3D printing the chassis should be easily disposable or biodegradable. Care has to be taken while disposing the failed electronic components, it has to be segregated to E-Waste.
CONCLUSION

A detailed study of the Hexapod robot was carried, analysed and defects deduced. Various improvisations were done for enhancing it. The Raspberry Pi is a powerful little beast and a great platform for building low-cost, but highly capable, embedded systems. The interfaces built into its GPIO connector make it easy to bolt on modules using simple low-cost electronics and to create functional and flexible systems. The inclusion of a dedicated camera interface and networking interfaces gives easy access for improvement. Several ideas to improve the design, ease and efficiency of the robot have been proposed. An effective integrated power supply unit to supply constant current to the motors consisting of regulators and the inclusion of the camera module was not possible due to lack of time. There was trouble with the hardware as the bot was not in a working state. As the existing regulators burnt during the last moment due to improper power supply from the lab power source under supervision, it was not possible to replace them due to insufficient time frame. A single segment has been disassembled from the Hexapod and certain electronics modified to test the working mechanism of the 3-segmented Hexapod. Similar mechanism is replicated in the other segments. Ultrasonic sensor is connected to the Raspberry Pi and the distance is calculated and the appropriate command is sent to the Arduino from the Pi. The Arduino Uno can be replaced with a mini once the electronics of the bot are finalised. Uno has been used for temporary experiment of the program. Without an appropriate power supply to the motors, various algorithms cannot be tested and verified. The ideas mentioned for improvement could not be implemented due to lack of proper power supply and lack of time.
Enhancement of a Segmented Hexapod Robot

BIBLIOGRAPHY

https://hexyrobject.wordpress.com/2015/11/20/common-walking-gaits-for-hexapods/

http://www.airspayce.com/mikem/bcm2835/

https://books.google.es/books?hl=en&lr=&id=bDTCzyzpCe8C&oi=fnd&pg=PR4&ots=54Uw1bdf66&sig=Aya6-ExfhZqC0fm2eLeFHEs3x4&redir_esc=y#v=onepage&q&f=false


https://www.arxterra.com/the-wave-gait/?v=e71be9e013d9

https://raspberry-projects.com/pi/pi-hardware/raspberry-pi-3-model-b-pi-hardware/rpi3-model-b-hardware-general-specifications-2


https://books.google.es/books?id=2l3KqsVPiwcC&pg=PA148&lpg=PA148&dq=support+polygon&source=bl&ots=B0kirBeF5L&sig=ACfU3U1FCruFhpHmuCYWet9gIgzb2t1SaQ&hl=en&sa=X&ved=2ahUKEwjHpa7Nj-fiAhWrAWMBHfunDcw4ChDoATABegQIBxAB#v=onepage&q=support%20polygon&f=false

https://www.mdpi.com/2218-6581/3/2/181/htm


https://circuitdigest.com/article/servo-motor-basics

https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/
ANNEXES

A1. Arduino Code

#include <Servo.h> //Importing Servo library
String a; //Initialising String a for Serial communication
Servo servo_3; //Vertical Left Front, Vertical Left Back, Vertical Right Middle
Servo servo_5; //Horizontal Left Front, Horizontal Left Back
Servo servo_9; //Vertical Left Middle, Vertical Right Front, Vertical Right Back
Servo servo_10; //Horizontal Right Front, Horizontal Right Back
Servo servo_6; //Horizontal Left Middle
Servo servo_11; //Horizontal Right Middle

int offset1=85; //Initial offset angles for Servo motors
int offset2=80;
int ang1=40; //Angle to be added or deducted from the initial offset for movement
int ang2=25;
int ang3=15;

void setup()
{
  Serial.begin(9600);
servo_3.attach(3); //Attach the signal pin of Servo motors to the respective pins of Arduino
servo_5.attach(5);
servo_9.attach(9);
servo_10.attach(10);
servo_6.attach(6);
servo_11.attach(11);

  servo_3.write(offset1); //Initialising the Servo motors to move to the offset positions for the stance phase
void loop()
{
    delay(1000);

    //Serial Communication - receiving the Ultrasonic Sensor signal from RPi
    if (Serial.available()) {

        a=Serial.readStringUntil("\n");

        while (a=='RUN')
        {
            delay(1000);

            servo_3.write(offset1+ang1); //Swing phase 1
            servo_5.write(offset2+ang2);
            servo_9.write(offset1);
            servo_10.write(offset2+ang2);
            servo_6.write(offset2-ang2);
            servo_11.write(offset2-ang2);

            delay(1000);

            servo_3.write(offset1); //Stance phase 1
            servo_5.write(offset2);
            servo_9.write(offset1);
        }
    }
}
servo_10.write(offset2);
servo_6.write(offset1);
servo_11.write(offset1);

delay(1000);

servo_3.write(offset1);  //Swing phase 2
servo_5.write(offset2-ang3);
servo_9.write(offset1-ang1);
servo_10.write(offset2-ang3);
servo_6.write(offset2+ang3);
servo_11.write(offset2+ang3);

delay(1000);

servo_3.write(offset1);  //Stance phase 2
servo_5.write(offset2);
servo_9.write(offset1);
servo_10.write(offset2);
servo_6.write(offset2);
servo_11.write(offset2);

delay(1000);

}

A2. Raspberry Pi Code

#Import required libraries
import RPi.GPIO as GPIO
import time
import serial
import math
from time import sleep
GPIO.setwarnings(False)

# Set to GPIO BCM mode
GPIO.setmode(GPIO.BCM)

# Set necessary GPIO Pins
GPIO_TRIGGER = 18
GPIO_ECHO = 24
do=4

# set GPIO direction - IN / OUT
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
GPIO.setup(do,GPIO.OUT)

ser = serial.Serial("/dev/serial1", 9600) # Open serial port 1 with baud rate 9600
ser.flushInput()

def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()
# Save emission time
while GPIO.input(GPIO_ECHO) == 0:
    StartTime = time.time()

# Save reception time
while GPIO.input(GPIO_ECHO) == 1:
    StopTime = time.time()

# time difference between emission and reception
TimeElapsed = StopTime - StartTime
# calculate the distance at which the obstacle is present
distance = (TimeElapsed * 34300) / 2

return distance

if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print("Measured Distance = %.1f cm" % dist)
            sleep(1)
            if dist > 15: # check the range of obstacle
                ser.write("STOP").encode()
                GPIO.output(do, True)
            else:
                ser.write("RUN").encode()
                GPIO.output(do, False)

        # Reset by pressing CTRL + C
    except KeyboardInterrupt:
        GPIO.cleanup()