SpidoBot

Submitted By:

- Usama Azam (Syndicate leader)
- Hassan Nawaz
- Arhama Riaz
- Aimen Tariq

Supervisor

- Dr. Adil Masood Siddiqui

Submitted to the faculty of Department of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, in partial fulfillment for the requirements of B.E Degree in Electrical Engineering

JULY,2020

CERTIFICATE OF CORRECTIONS & APPROVAL

Confirmed that work contained in this proposition titled"SpidoBot",carried out by Arhama Riaz, Aimen tariq, Usama Azam, Hassan Nawaz under the management of _Dr. Adil Masood for incomplete satisfaction of Degree of Bachelors of Electrical Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad during the scholarly year 2019-2020 is right and endorsed. The material that has been utilized from different sources it has been appropriately recognized/alluded.

Approved by

Supervisor

____

____

Date:__________________

DECLARATION

No part of work introduced in this postulation has been submitted on the side of another honor or capability in either this foundation or anyplace else.

Plagiarism Certificate (Turnitin Report):

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

ABC

Registration Number

Signature of Supervisor

Acknowledgements

We bow before Allah Almighty in outrageous appreciation for his advantage and altruism that he presented to us. Without His gifts it would have been out of inquiry to have gotten this far and accomplish what we have today.

We are obliged to our folks for their ceaseless help and for never letting our spirits capitulate to ominous conditions that went along that way.

We are incredibly appreciative to our director DR. Adil Masood Siddiqui whom we generally found close by to help us in each progression en route. We express gratitude toward him for having confidence in us and for drawing out the best of our capacities. His direction was really an encouraging sign for us in difficult stretches

At long last we might want to thank every one of our educators and personnel at MCS who helped us in one manner or the other to arrive at this objective of our own which appeared to be dreamlike at its initiation

Dedicated to all the doctors, nurses, paramedical staff and everyone else who has risked their life to safe others in this notorious pandemic

Abstract

Because of their capacity of playing out a wide scope of burdensome, intellectual and grave assignments, the utilization of robots in the business and our day by day life has gotten key. Because of its flexibility of utilization, mechanical autonomy has become a center point of logical exploration Legged robots have assumed control over the traditional wheel robots as they are better capable in movement and mobility as they can accomplish both static and dynamic

soundness by moving their focal point of gravity. The primary inspiration driving our fyp is to plan and construct a quadruped arachnid robot. Our robot would be mounted with a camera that would record live take care of from zones that are blocked off to people and send the input to the base station. This could be amazingly useful in reconnaissance purposes.

Table of Contents

List of figures:

List of tables:

CHAPTER 1: INTRODUCTION

Etymology of a Robot

The term starts from a Czech word, robota, implying "compelled work"; the word 'robot' was first used to mean a recounted humanoid in a 1920 play R.U.R. (Rossumovi Univerzální Roboti - Rossum's Universal Robots) by the Czech writer, Karel Čapek yet it was Karel's kin Josef Čapek who was the word's genuine inventor.[1]

1.1 Robot and its History

A robot performs tasks automatically with the instructions that are fed into it. There are a plethora of robots that exist in today’s day and age and each have their specific set of instructions they perform. Humans have been trying to replicate their functionality since . Although ancient Greeks, Chinese and Egyptians have been trying to achieve making a robot from the origin of time Unimate was the first ever robot made by Kawasaki in 1954. it was a robotic arm programmed to stack and weld hot pieces of metal.

1.2 All types of Robot

The sorts of robots that are as of now being made and acquired use by our Industry is huge including fixed robots, wheeled robots, legged robots, swimming robots, nano robots, miniaturized scale robots, measured robots, swarm robots, delicate versatile robots, flying robots and so forth. In any case, holding appropriate to our subjects robots can be chiefly isolated into three sorts by the prudence of movement, in particular fixed robots, wheeled robots and legged robots

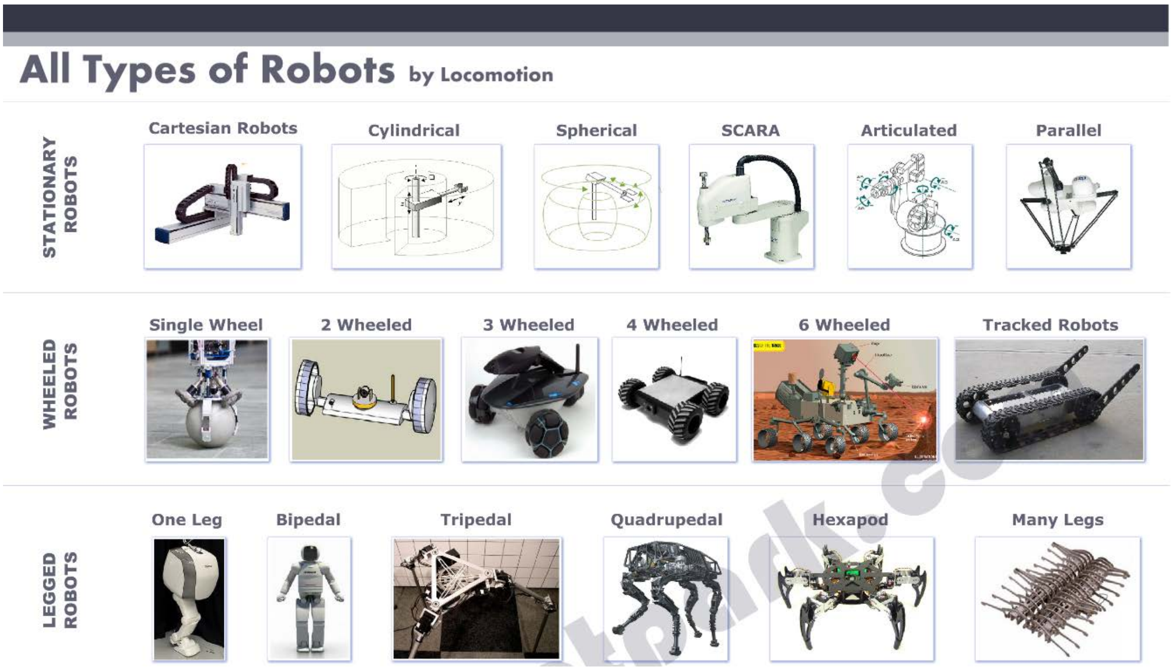

Figure 1.1 Various kinds of robots

Stationary or autonomous robots perform tasks which are iterative or require autonomy but as we are not working with one we shall exclude it from our discussion.

1.3 Disadvantages of Wheeled Robots

Wheeled robots will be robots that cross on a superficial level with the utilization of their wheels.Their design is much simpler as compared to legged robots. However they have a few disadvantages:[2]

They can’t navigate well over obstacles

They have a hard time traversing over uneven terrain, declined areas and areas with low friction

It is only easier for them to travel at straight paths with no pits or ditches

It needs to be 2 times taller to climb a vertical obstacle

1.4Tracked Robots

Although tracked robots have merits over the wheeled robots still there are issues that persist including:

These robots are fragile against pointed obstructions and are vulnerable to being punctured or torn at contact

If the tracks aren't protected by the sides there is a high chance of these robots being damaged

These robots have speed and agility issues as there is extremely high friction because of the ground contact surface

1.5 Legged Robots

Legged robots are the types which has legs like human or animal for traversal motions. With the passage of time mimicry of human limbs started to being used in robotics. Like the robotic arm and robotic leg. The main reason being high stability. Some of the advantages of legged robots include:

Jumping over obstacles could be achieved and undesirable footholds and pits could be avoided

Traversal over multiple terrains is possible due to versatility and agility

- These robots are considered more stable and resistant to opposing forces

Statistically stable (TITAN series, Wildcat, Cheetah, Big dog)

Working in complex environments is considered a hallmark of such robots

1.6 Selection criteria for number of legs

In legged robots stability is in direct relation with the number of limbs of the specimen but increasing the legs would also mean more cost and more complexity so while deciding the number of legs tradeoff has to be achieved between cost and performance. While talking about a robot following two terms are of utmost importance and can’t be neglected:

Static Stability: This term refers to the fact that the robot is well balanced and does not fall when standing. Also the center of gravity is within its ground contact base

Dynamic Stability: When a moving object is stable during its movement we say that the object is dynamically stable. Although achieving dynamic stability is a onerous task and is harder to control it gives more control, efficiency, stability and agility to the robot

Table 1.1: Stability comparison of Robots

No. of Legs	Static Stability	Dynamic Stability
1	Does not exist	Exists
2 (biped)	Hardly exists	Exists
3 (triped)	Exists	Can be possible
4 (quadrupled)	Exists	Can be possible
6 (hexapod)	Exists	Can be possible

From the above analysis it is extracted that triped, quadruped and hexapod robots could be of our choice. We choose a quadruped robot as its complexity and efficiency lies midway.

1.7 Advantages of a Quadruped Robot

Quadruped Robots profit by expanded security over different robots, and particularly during their development. At moderately moderate speeds, a quadrupedal robot may utilize just a single leg at once, in this manner it would guarantee a steady tripod. Four legged robots additionally have a bit of leeway of a lower focus of gravity which two-legged frameworks need

1.8 Applications of a Quadruped Robot

The most mentionable applications of a quadruped robot in all arenas of life include Exploration of dangerous and / or rough areas for humans. For instance searching for survivors after a nuclear disaster, exploring war zones, inspecting unstable buildings after natural disasters such as an earthquake, tsunami or a volcanic eruption. Equipped with sensors and weapons, such a robot could be used in a crisis or war to avoid risking human lives in battle. Defusing bombs such as land mines. Currently robots with tracks are used to do this but this robot spider would have the serious advantage in that it could be used in more impassable areas than the current robots and could be helpful in guarding property and

1.9 Literature Review

As it is decided that we will make material with which the robotic parts will be made ourselves there was a lot of research behind it and we read a plethora of research papers and online material that we read. [3],[4],[5]. Taking into account the control of four legged robots PabloGonzalez de Santos. presented a detailed account on that in his book Quadrupedal locomotion .Studying the locomotion of quadruped robots research papers by Than Trong khanh and Tran Thien Phuc [6] was very useful. For the designing aspect of the robot article by Dhurbajyoti Gupta. [7] proved helpful. Kinematic Analysis of a quadruped robot by Alain Segundo Potts. [8] and design of quadruped robot and its inverse kinematics PALIVELA ARUN KUMAR & YEOLE SHIVRAJ NARAYAN.[9] As the gait and stability analysis is a very important aspect Junming Wei. [10] presented a good approach on that.

1.10 Incentive of SpidoBot

There are various quadruped robots out in the market. What makes our robot one of its kind is that there are various havoc wreaked areas where human interaction is impossible and the areas become inaccessible to humans. In such circumstances we need someone to provide us with what's happening in that area. This is where our SpidoBot comes to rescue. SpidoBot would be sent into such an inaccessible area as we will be communicating with it via a Bluetooth Module. The SpidoBot will be equipped with a payload that will capture the live feed of the particular area and send it back to the base station. Back at the base station this information would be used to calculate the amount of destruction or loss in that area and plan ahead.

CHAPTER 2: MATERIALS

2.1 Background

When it came down to us deciding the material of our robot it was clear in our heads that we would be manufacturing our own material as the prevalent materials in the market for manufacturing robotic parts did not fit our requirements of the material we needed. The ultimate material of our choice had to be lightweight, cost effective and robust. Thus we set our foot on the journey of manufacturing our own material. Soon we realized that the material had to be a composite material as simpler materials alone did not fulfil our requirements

2.2 Composites

Talking about composites one should know what basically a composite is. “Composite is a word that is used to refer the solid materials that are out there and their composition consists of multiple components and those components are in different phases in that material”. Meaning one takes multiple materials and a process on them that could include applying temperature, pressure with respect to the product. This new material would be a synergy having properties of all the materials used and taking the cross section of the material one could make out all the materials that have been added

2.3 Comparison Metrics of common materials

The most common choices when it comes to making robotic parts are acrylic, steel, wood, Aluminum, Carbon Fiber etc. Following is a comparison metrics of theses materials that will help us with our selection

TABLE 2.1:: Comparison metrics of various materials

Material	Tensile Strength	Cost Effectiveness	Weight	Thermal Expansion
Acrylic	✓	✗	✓	✗
Wood	✗	✗	✗	✗
Aluminum	✓	✗	✓	✓
Steel	✓	✗	✗	✗
Fiber Glass	✓	✓	✓	✓

2.4 Matrix and Reinforcements

The two prime segments that are of most extreme significance while looking at assembling a composite are grid and support. These two individual components when joined structure a composite

a) Matrix: The lattice is fundamentally a homogeneous and solid material wherein a fiber arrangement of a composite is implanted. It is totally nonstop The lattice is the primary material in a composite. It ties the fiber fortification, gives the composite segment its shape and decides its surface quality. The fundamental elements of a grid incorporate halting or easing back the engendering of a break, supporting in the creation of net shape or close to net shape parts, disconnection of filaments with the goal that singular strands can act independently and so forth. The three primary classes of a lattice are polymer, artistic or metal.

b) Reinforcement Fibers: It gives quality and firmness to the composite material one way .It expands the mechanical properties of the composite, gives electrical conductivity or protection, auxiliary properties are likewise given by the fortification fiber. Fortifications can be nonstop, spasmodic or molecule type Contingent on these two boundaries we need a synergic yield

2.5 Factors that govern fibers

There are a number of properties that play a vital part in the robustness of a composite including the basic mechanical property of the fiber, surface interaction, orientation. When talking about the reinforcement fibers as a whole they depend on the type as well as the form we select.The forms include either 1D, 2D or 3D and the types include fiber glass, Carbon, organic or ceramic

2.6 Intricacies of our composite

For our matrix selection it came all down to us selecting Epoxy. Epoxy is the sub branch thermoset polymers. The main incentive behind selecting are as follows:

- Low viscosity
- Less temperature and Pressure requirements
- Unlimited storage life
- Low Shrinkage
- Excellent adhesion
- Very wide set of applications ranging from aerospace to space goods

As discussed above there are different forms and types of reinforcement fibers we selected the easily available 2D preform i.e. Chopped strand Mat (CSM) which is basically taking 2 1D's and weaving them. Short 1D fibers are taken, stacked and pressed together. The main reasons that led us to this selection are as follow:

- Highly economical
- Most widely used
- Chemically resistant
- Extreme hardness and stiffness

Fig: A sheet of CSM fiber

Although the best choice could have been using carbon fiber but it would tend to make our project less cost effective

2.7 Manufacturing Process

There are a great deal of decisions that could be ,made while choosing the assembling procedure and manufacturing the parts. The measures for choosing the procedure relies upon the

- Rate of creation,
- cost ,
- tensile quality,

- size
- shape necessities

The most well-known strategies for assembling of parts incorporate fiber winding, pultrusion, hand layup, shower up, wet layup, RTM, SRIM, pressure shaping, stepping, infusion trim and move wrapping. The procedure that we picked was hand lay-up process .hand layup process will be process where the materials are squeezed by hand so its increasingly reinforced and afterward the apportion of impetus and epoxy makes it worth the procedure. This decision was made on the grounds that for the size and state of our automated parts it invigorated us the due measure of that we required and it additionally was savvy.

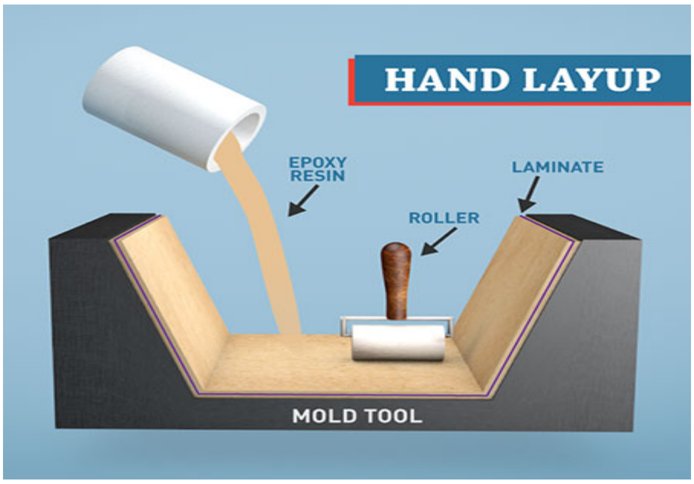

Fig: The hand layup process

2.8 Catalyst Selection

It is to be noted here that the catalyst selection for the manufacturing of a material plays a very vital role as it is needed for curing purposes.curing is a physical process to make a hard and strong material. We needed a catalyst for curing our epoxy resin. The resin that we came down to selecting was MEKP (Methyl Ethyl Ketone Peroxide) is the catalyst added to polyester resins and vinyl ester resins. The other catalyst that we used in addition to MEKP was cobalt

2.9 The actual process

For making sheets of fiberglass we took almost 2 liters of Epoxy Resin. For this we added 50 gram of MEKP and Cobalt catalyst each to cure our resin. We used 10 oz plain weave cloth. On a surface to which glass does not stick the CSM fiber glass was placed. With the help of a brush we coated it with the epoxy and catalyst mixture until we got the desired thickness. The strength of the composite material depends on the number of sheets. The more the sheets the more strong our output would be. After this the sheets were set aside to dry. It took almost 2 hours to dry. The tensile strength of this fiber glass was almost equal to that of steel and it was very lightweight. After the sheets were made the next task was to cut them into our desired robot parts. For cutting there are a number of ways that are used including heavy duty sheers, sawing, laser, water jet and ultrasonic machining. Initially we did use the saw and shears but it was a laborious task which did not give us precision and also it damaged the sheets. So ultimately we resided to laser cutting and got all our robotic parts laser cut.

Concluding the entire composite material manufacturing journey it was a hectic trajectory that we had to follow and we definitely had to go an extra mile to achieve our desired output but it was definitely worth the effort

CHAPTER 3: KINEMATICS

3.1 Robot anatomy

A robot comprises of arrangement of kinematic chains of connections and joints. Essential development incorporates associating unbending bodies (joins) together by methods for joints so relative movement between joins gets conceivable. Incitation as far as mechanical autonomy is the activity of making the robot work and activation of joints is done ordinarily by engines. While numbering the connections in a robot it is typically begun from 0 and goes till n. While that of joints is done from 1 till n.

3.2 Types of links

As described above a link is a rigid member connecting the joints. Generally there are two types of links in a robot:

1. Input link: It is the link that is near the base of the robot
2. Output link: It is the link that is near to the end-effector of the robot which is the load bearing end

3.3 Types of joints

A joint (likewise called hatchet) is the versatile part of the robot that causes relative movement between contiguous connections. There are commonly two fundamental sorts of robot joints considered while building a robot:

1. Prismatic joint: kaleidoscopic joints have one DoF and are utilized to depict translational developments between objects. Their arrangement is characterized by one worth that speaks to the measure of interpretation along their first reference casing's z-hub. They can be utilized as latent joints, or as dynamic joints (engines).

3. Revolute joint: revolute joints have one DoF and are utilized to portray rotational developments (with 1 DoF) between objects. Their arrangement is characterized by one worth that speaks to the measure of revolution about their first reference edge's z-pivot. They can be utilized as inactive joints, or as dynamic joints (engines).

3.4 Subsystems of a robot

There could be a few subsystems of a robot relying upon the specific sort of robot under development. The primary subsystems that we ought to have a skill about are as per the following:

1.Base: A base is a steady stage to which the arms or legs of a robot are appended. The base could be fixed or versatile

2.Manipulator: A robot controller is an electronically controlled instrument, wherein errands are performed by interfacing with nature. There are body, arm and rest controllers utilized by the need

3.End-effector: an end effector is a gadget or apparatus that is associated with the finish of a robot controller. The end effector could be mechanical, pneumatic or attractive

3.5 Kinematics of Robot’s Leg

Our robot comprises of four legs. Each has two degrees of opportunity. In this manner, we have to control a sum of eight degrees of opportunity all the while with the end goal that it results into a steady stride. For accomplishing a steady stride we need to investigate the direction of every leg and ensure that it does not conflict with zero second point dependability rules.

3.6 Forward kinematics methods

In forward kinematics given the joint variables we find the end-effector position, in other words the joint variables are the input and the end-effector position is the output. Basically there are 2 ways to find forward kinematics:

1. We draw coordinate frames on each of the joints and then we find the rotation matrixes and displacement vectors that transform one set of coordinates into the next. We then combine the rotation and displacement matrices into the homogeneous transformation matrices.
2. The second method is the Denavit-Hartenberg representation method. It also helps to find the homogeneous transformation matrix. It works well in the case where we have manipulators with different kinds of offsets. The previous method which we use for manipulators that do not have offsets would still work. D-H is easier and standard. In this particular method we do not have to be particular about how we assign coordinate axis but after we have assigned the coordinate axis the process of finding he homogeneous transformation matrix is complex with this method.

3.7 Homogeneous transformation matrices

Have you ever wondered how a robot makes sense of the 3D world we live in? One thinks of some type of sensor like camera, GPS or system coupled with doppler velocity log. But still what does the robot do with all this information, how can it take in sensor data from the world and do something meaningful with it? All the answers lie in a transformation matrix which can mathematically represent movement in 3D space. Take a robotic arm for example, T.M can be used to tell at what angle the servos need to be to reach the desired position in space or maybe an autonomous underwater vehicle needs to reach to align itself with several different objects in the water. T.M could be used to tell how all the obstacles are rotated and translated with respect o the underwater robot or even how the obstacles are rotated and translated between each other. Between any two frames, two things have to be taken care of

1. Rotation

2. Translation

$$ {}_{1}^{0}T = \begin{bmatrix} r11 & r12 & r13 & d1 \\ r21 & r22 & r23 & d2 \\ r31 & r32 & r33 & d3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1) $$

3.8 Coordinate frame assignment

Let’s first start by taking a deeper look into assigning coordinate frames to an object. In order to fully describe an object in 3D space one needs its rotation and translation w.r.t some reference frame and in order to do that all the objects of interest are given reference frames because it gives a more defined description when rotating or translating an object. For example if in fig 3.1 it was asked to rotate the object by 30 degrees there could be several different configurations but if asked to rotate by positive 30 degrees about its x-axis then it would be exactly known what the object would look after rotation. In this case the reference frame is labelled 0 and object frame as 1. The difference between these two frames is the T.M. So ${}^{0}_{1}T$ is the notation that could be read as the transformation of frame 1 w.r.t frame 0.

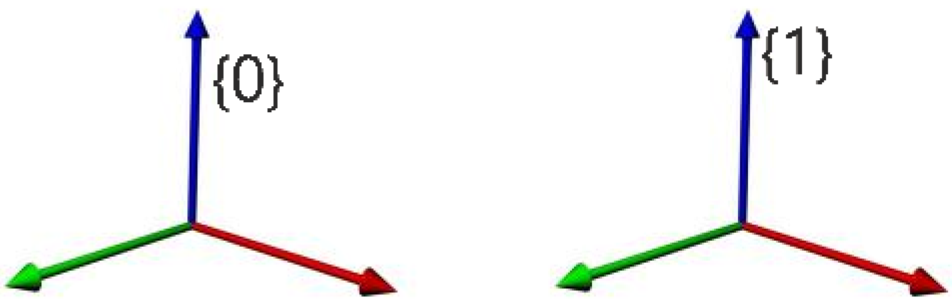

Figure 3.1: Coordinate frame assignment

To understand the trajectory of each joint we will make use of forward and inverse kinematics.

Coordinate frames are assigned for the use of forward and inverse kinematics.these are shown in Figure below.

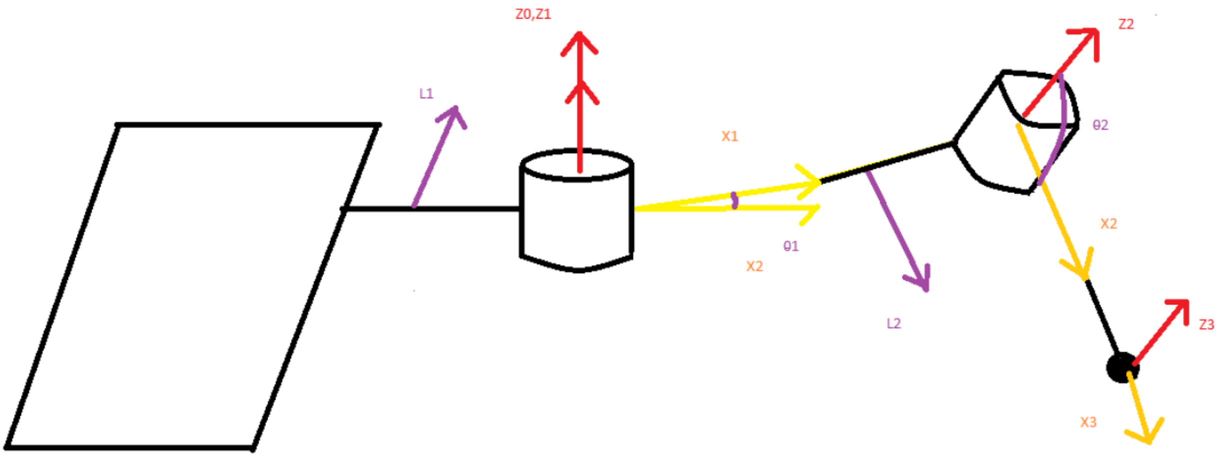

Figure 3.2: Coordinate frame assignment on leg of a spider

3.9 DH parameters

The context of these parameters is that we are trying to find homogeneous transformation matrix from one frame which we will call n-1 to the frame immediately after it which we will call frame n. In DH parameter w use the same naming convention as used above. Following are the 4 DH parameters and their explanation:

θ_i	It is a rotation parameter defined angle between $X_{i+1}$and $X_i$aboutZ_i. It is also called as joint angle
α_i	It is also a rotation parameter and is the angle between Z_i. AndZ_{i+1} aboutX_i . It is also called as link twist
a_i	It is a displacement parameter and is the distance between Z_i. AndZ_{i+1} along Xi. .It is called as link length
d_i	It is a displacement parameter and is the distance between$X_{i-1}$and $X_i$along Z_i. It is called as link offset

The accomplishment of DH depends on accurately characterizing the facilitate outlines at each purpose of the controller. Adhering to rules must be followed while doing this:

1. Z-pivot is toward joint hub
2. X-hub must be symmetrical to both Zn (current casing for which we are as of now drawing) and its past Z outline
3. Y-pivot keeps right hand rule
4. The current X-pivot must converge the past Z hub

Table 3.1: Denavit Hartenberg parameters

I	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	L1	0	θ_2
3	0	L2	0	0

3.10 Coordinate frame translation

Transformation matrices are used to evaluate vectors from base of robot to foot of its leg

$$ {}_1^0T = \begin{bmatrix} r11 & r12 & r13 & d1 \\ r21 & r22 & r23 & d2 \\ r31 & r32 & r33 & d3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1} $$

Now we will be calculating the transformation matrices for the robot’s leg using this generalize

formula which makes use of Denavit Hartenberg Parameters

$$ {}^{i-1}_{i}T = \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & a_{i-1} \\ sin\theta_i cos\alpha_{i-1} & cos\theta_i cos\alpha_{i-1} & -sin\alpha_{i-1} & -sin\alpha_{i-1} d_i \\ sin\theta_i cos\alpha_{i-1} & cos\theta_i sin\alpha_{i-1} & cos\alpha_{i-1} & cos\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

(3.2)

Putting the DH parameter values from the table into above equation

$$ {}^{0}_{1}T = \begin{bmatrix} cos\theta_1 & -sin\theta_1 & 0 & 0 \\ sin\theta_1 & cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3) $$

$$ {}^{1}_{2}T = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ -sin\theta_2 & -cos\theta_2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4) $$

$$ {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5) $$

As ${}^{a}_{b}T * {}^{b}_{c}T * {}^{c}_{d}T = {}^{a}_{d}T$

$$ {}^{0}_{3}T = \begin{bmatrix} cos\theta_1 cos\theta_2 & -cos\theta_1 sin\theta_2 & -sin\theta_1 & L_2 cos\theta_1 cos\theta_2 + L_1 cos\theta_1 \\ cos\theta_2 sin\theta_1 & -sin\theta_1 sin\theta_2 & cos\theta_1 & L_2 cos\theta_2 sin\theta_1 + L_1 sin\theta_1 \\ -sin\theta_2 & -cos\theta_2 & 0 & -L_2 sin\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6) $$

3.11 Forward Kinematics

Forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters. Forward kinematics model for our robot can be extracted from the fourth column of ${}^{0}_{3}T$ matrix.

$x = L_2 cos\theta_1 cos\theta_2 + L_1 cos\theta_1$ (3.7)

$y = L_2 cos\theta_2 sin\theta_1 + L_1 sin\theta_1$ (3.8)

$z = -L_2 sin\theta_2$ (3.9)

Following is the MATLAB code for forward kinematics

```
k1 means theeta1

% k2 means theeta2

syms k1 k2 L1 L2;
r=pi/180;


L1= input('Length of Link 1= '); L2=
input('Length  of  Link  2=  ');  k1=
input('Theta 1= ');

k2= input('Theta 2= ');


z1= [cos(k1*r) -sin(k1*r) 0 0; sin(k1*r) cos(k1*r) 0 0; 0 0 1

0; 0 0 0 1]

Z2= [cos(k2*r) -sin(k2*r) 0 L1; 0 0 1 0; -sin(k2*r) -cos(k2*ra)

0 0; 0 0 0 1]

z3= [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1] z1

z12= z1*z2

= z2*z12

Z3(:,4)
```

3.12 Inverse Kinematics

Inverse kinematics refers to the use of kinematic equations of a robot to determine the joined parameters that provide the desired movement of end-effector. Inverse kinematics takes as input the Cartesian end effector position and orientation, and calculates joint angles. Inverse kinematics is used for trajectory planning.Solving set of forward kinematics Equations analytically, we will be able to form inverse kinematics model for the robot.

$\theta_2 = sin^{-1}(\frac{-z}{L_2})$

(3.10)$\cos\theta_1 = \frac{x}{L_1 + L_2 \sin(\frac{-z}{L_2})}$

(3.11)

$\theta_1 = cos^{-1}\left[\frac{x}{L_1 + L_2 \sin(\frac{-z}{L_2})}\right]$(3.12)

3.12 Trajectory of motors

Since we have converse kinematic model of every leg, we can make an interpretation of direction of foot into direction of the considerable number of joints. This will permit us to program actuators such that they will at the same time execute the ideal movement. Utilizing forward and opposite kinematics models we assessed points of engines and break down directions by plotting them in MATLAB.

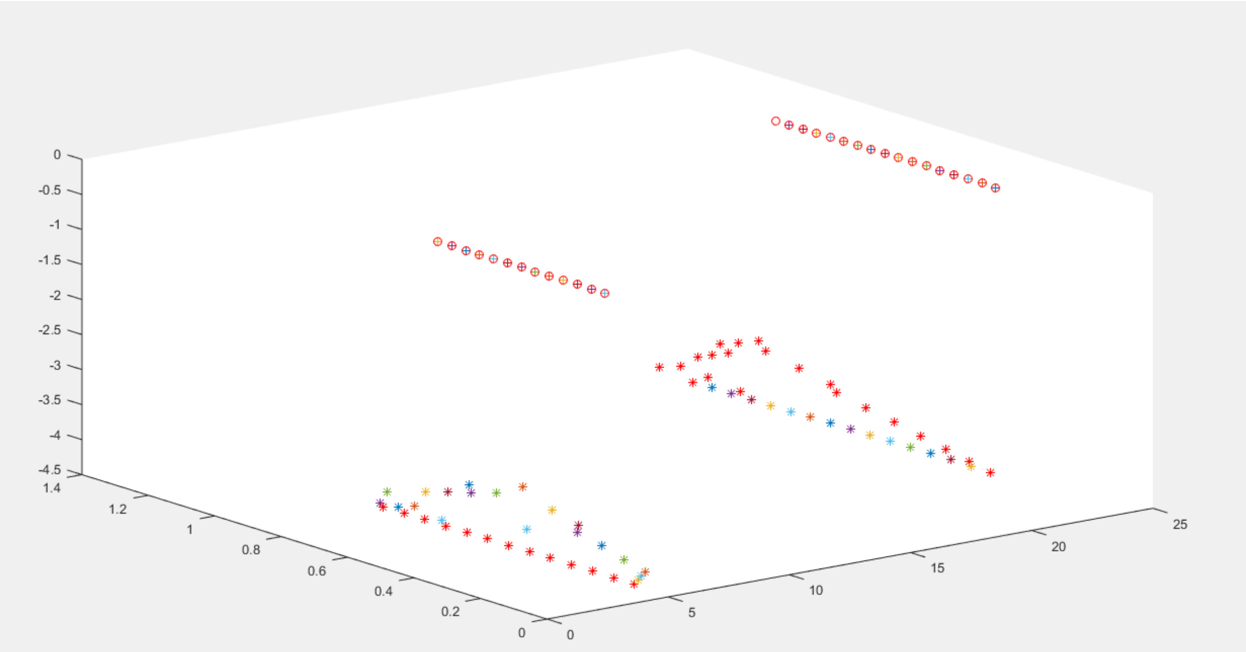

Fig 3.3:trajectory of front legs of a spiobot

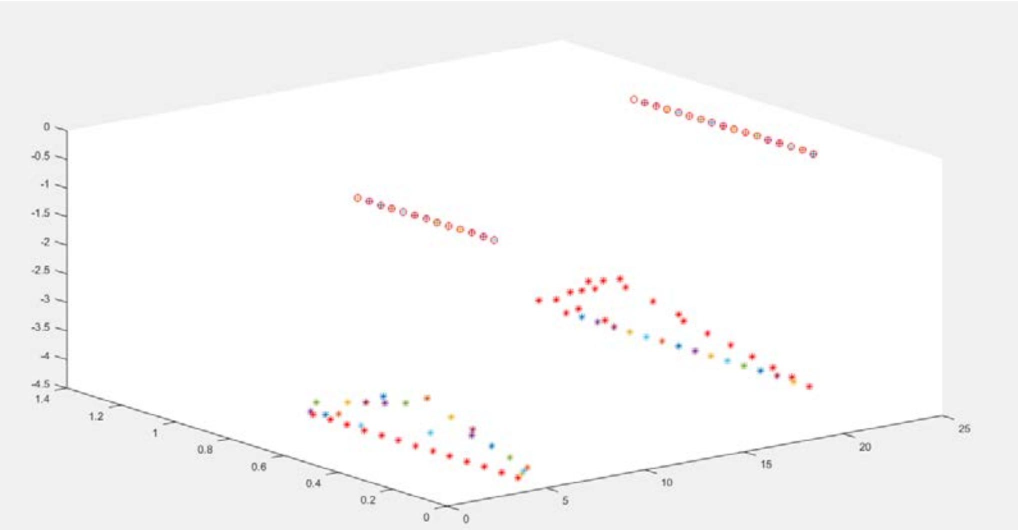

Figure 3.4: Trajectory of rear legs of a spidobot

Following is the MATLAB code for front leg trajectories:

```
k1=[5,6,6,7,8,8,9,10,10,11,12,12,13,14,14,15,16,17,18,19,20,20,21,19,1
0,17,17,16,15,15,14,13,13,12,11,11,10,9,8,7,6,5];
k2=[76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,
74,71,72,74,66,64,62,60,58,56,58,60,62,64,66,68,70,72,74,76];
k3=[12,11,11,10,9,9,8,7,7,6,5,5,4,3,3,2,1,0,0,0,0,0,0,0,0,0,0,1,2,2,3,
4,4,5,6,6,7,8,9,10,11,12];
k4=[73,72,68,70,72,64,62,60,58,58,54,66,58,60,62,64,66,68,70,72,74,74,
74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74];
syms L1 L2;

L1=2.4; L2=4.4;
for i=1:42;

xa= 19+L1*cosd(k1(i));
ya= L1*sind(k1(i));
za=0;
display (xa);
display (ya);
display (za);
xb= 19+L1*cosd(k1(i)) + L2*cosd(k1(i))*cosd(k2(i));
display (xb);
yb= L1*sind(k1(i)) + L2*cosd(k2(i))*sind(k1(i));
zb= -L2*sind(k2(i));

xc= L1*cosd(k3(i));
yc= L1*sind(k3(i));
zc= 0;

xd= L1*cosd(k3(i)) + L2*cosd(k3(i))*cosd(k4(i));
yd= L1*sind(k3(i)) + L2*cosd(k4(i))*sind(k3(i));
zd= -L2*sind(k4(i));
```

```
if i<=21
plot3(xa,ya,za,'p','Marker','+') ;hold on;
elseif i>21
plot3(xa,ya,za,'r','Marker','o') ;hold on;
end
if i<=21
plot3(xb,yb,zb,'p','Marker','*') ;hold on;
elseif i>21
plot3(xb,yb,zb,'r','Marker','*') ;hold on;
end

if i<=21
plot3(xc,yc,zc,'p','Marker','+') ; hold on;
elseif i>21
plot3(xc,yc,zc,'r','Marker','o'); hold on;
end

if i<=21
plot3(xd,yd,zd,'p','Marker','*'); hold on;
elseif i>21
plot3(xd,yd,zd,'r','Marker','*'); hold on;
end
pause (0.5);
end
```

Following is the MATLAB code for rear leg trajectories

```
k1=[5,6,6,7,8,8,9,10,10,11,12,12,13,14,14,15,16,17,18,19,20,20,2
1,19,18,17,17,16,15,15,14,13,13,12,11,11,10,9,8,7,6,5];
k2=[76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,76,
76,76,74,71,72,74,66,64,62,60,58,56,58,60,62,64,66,68,70,72,74,7
6];
k3=[12,11,11,10,9,9,8,7,7,6,5,5,4,3,3,2,1,0,0,0,0,0,0,0,0,0,0,1,
2,2,3,4,4,5,6,6,7,8,9,10,11,12];
k4=[73,72,68,70,72,64,62,60,58,58,54,66,58,60,62,64,66,68,70,72,
74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,7
4];
syms L1 L2;

L1=2.4; L2=4.4;

for i=1:42;

xa= L1*cosd(k1(i)); ya= L1*sind(k1(i)); za=0;

xb= 1.5*L1*cosd(k1(i)) + 1.5*L2*cosd(k1(i))*cosd(k2(i)); yb=
1.5*L1*sind(k1(i)) + 1.5*L2*cosd(k2(i))*sind(k1(i)); zb= -
1.5*L2*sind(k2(i));
```

```
xc= 19+L1*cosd(k3(i)); yc=  L1*sind(k3(i)); zc= 0;

xd= 1.2*19+L1*cosd(k3(i)) +1.2*L2*cosd(k3(i))*cosd(k4(i)); yd=
1.2*L1*sind(k3(i)) + 1.2*L2*cosd(k4(i))*sind(k3(i)); zd= -
1.2*L2*sind(k4(i));


if i<=21
plot3(xa,ya,za,'p','Marker','+');hold on;
elseif i>21
plot3(xa,ya,za,'r','Marker','o'); hold on;
end
if i<=21
plot3(xb,yb,zb,'p','Marker','*') ;hold on;
elseif i>21
plot3(xb,yb,zb,'r','Marker','*') ;hold on;
end

if i<=21
plot3(xc,yc,zc,'p','Marker','+') ;hold on;
elseif i>21
plot3(xc,yc,zc,'r','Marker','o') ;hold on;
end

if i<=21
plot3(xd,yd,zd,'p','Marker','*') ;hold on;
elseif i>21
plot3(xd,yd,zd,'r','Marker','*') ;hold on;
end
pause (0.5);
end
```

3.13 Stability Margins

Since the commencement of strolling robots a few static and dynamic security models have been characterized. By the by, various applications may require distinctive dependability standards and, up to the creators' best information, there is no subjective arrangement of such steadiness estimations. Controlling a robot stride by methods for utilizing an inappropriate solidness standard may keep the errand from succeeding. By the other hand, if the ideal measure is discovered the robot walk can likewise be improved. Two of the most well-known steadiness measures utilized on the planet are examined underneath.

1. Static Stability: Static strolling accept that the robot is statically steady. Implying that our robot will remain in a steady position regardless of whether anytime the robot has halted. It is vital that the projection of the focal point of gravity of the robot on the ground must be contained inside the foot bolster zone. In the event that it is the situation of one supporting leg we would characterize the help territory as the outside of the foot or the base curved zone containing both foot surfaces on the off chance that we are discussing both the feet. This sort of strolling isn't extremely successful as the measure of necessities are more than the outcome gave. Most analysts have rather gone for the second sort of strolling that is dynamic strolling, on the grounds that its aftereffects of development are progressively practical and spry developments.

2. Dynamic Stability: Biped powerful strolling permits the focal point of gravity to be outside of the help district for constrained measures of time. The security gave by powerful strolling isn't a flat out models yet the robot could be planned such that it can recuperate from the insecurities it faces. Nonetheless, if the robot has dynamic lower leg joints and consistently keeps at any rate one foot on the ground then the Zero Moment Point (ZMP) can be utilized as a dependability rules. The ZMP is where the robot's all out second at the ground is zero. As long the ZMP is inside the help locale the strolling is viewed as powerfully stable in light of the fact that is the main situation where the foot can control the robot's stance. Obviously for robots that don't consistently keep at any rate one foot on the ground or that don't have dynamic lower leg joints

(strolling on braces), the thought of help zone doesn't exist, in this manner the ZMP rule can't applied. Dynamic strolling is accomplished by guaranteeing that the robot is continually pivoting around a point in the help district. On the off chance that the robot turns around a point outside the help locale, at that point this implies the supporting foot will in general get off the ground or get presses against the ground. The two cases lead to flimsiness. To draw a similarity with static strolling, on the off chance that all movement is halted, at that point the robot will in general pivot around the ZMP.

Since dynamic strolling is progressively lithe, quick, devours not so much vitality but rather more reasonable, so we present powerful strolling in our robot and picked zero second point as our dependability rules.

3.14 Gait Pattern

The spidobot can essentially utilize two distinct sorts of walks to move from one highlight the other. It can walk straight and it can turn about its own focal vertical hub. Both of these walks are clarified beneath.

1.Straight Walking: For straight strolling robot move its four legs so that any two of its legs perform indistinguishable movement at any second in time. As the two legs of the robot are in contact with the ground the other two stays in air. The legs in contact with the ground push it in reverse which empower the arachnid to push ahead as a response. Simultaneously the other two legs are playing out a bended movement noticeable all around to arrive at the beginning stage of the cycle and prepare for contract with the ground. In the event that 'T' speaks to the ideal opportunity for one complete pattern of every leg in fig

2.For turning of robot about its own focal vertical pivot, one of the front legs of the robot pushes while different pulls. Considering fig 3.5(a), If R1 is in contact with the ground and pushing the robot toward turning then L2 will likewise be in contact with the ground moving in such a manner to forestall any slipping. Simultaneously the other front leg L1 will be moving noticeable all around so that it arrives at the point from where it can pull the robot toward turning. For whatever length of time that L1 stays in air R2 will likewise stay in air. In this manner at any second in time two legs are in contact with the ground while the other two are in air. Fig 3.5(a) and fig 3.5(b) are likewise legitimate for this step.

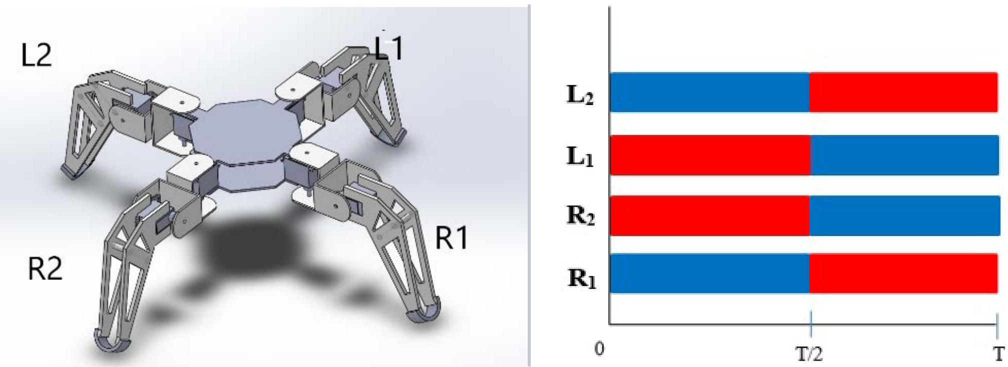

Figure 3.5: (a) Leg convention **(b)** Schematic representing the position of foot

represents foot in contact with the ground represents foot in the air

Chapter 4: MECHANICAL DESIGN

4.1 Overview

For the design bit of our robot we knew a few things beforehand. It was an 8 degree of freedom giving each leg a 2 degree of freedom. Each of the legs would further have sub parts viz upper leg and lower leg. These legs would be attached to the main body or base plate of the robot which would be carrying the payload for our design. Although these are the main parts of our robot, numerous fasteners and parts were used to affix the main parts.

As described in the materials section we did not buy the robotic parts although there is a plethora of such parts that are available online. Instead we went with making the parts ourselves. This made it possible for us to carve out the parts as we desired and which were most feasible for our project.

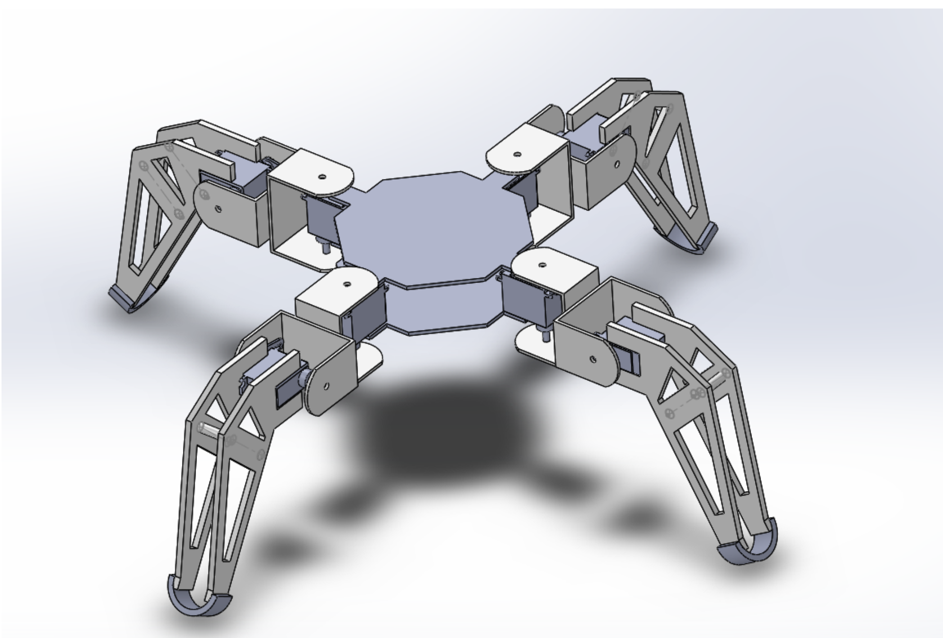

Figure 4.1: SolidWorks model of Spidobot

4.2 Base plate (Main body)

The base plate was the main part of the robot as it was supposed to carry all of the electrical equipment i.e Arduino, battery etc. The base plate had to provide the robustness and sturdiness to the robot. The size of the base plate was determined keeping in mind the following 2 factors:

1. It should be large enough to carry the payload
2. It should not add any unnecessary weight to the robot

Considering the above aspects in mind there were a few designs of the base plate we came up with. Also we examined quite a few designs of base plates online and most of them were designed on the same theme as described above.

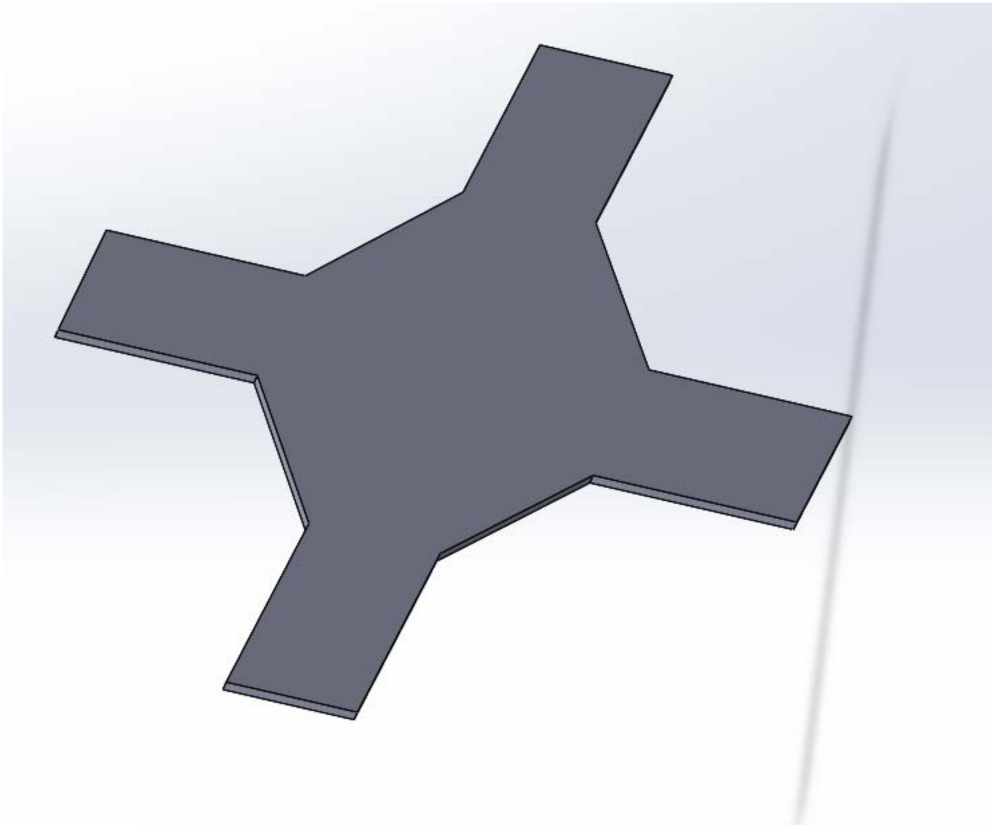

Figure 4.2: Base plate

As seen above the length and width was selected considering the payload. The thickness of the base plate was achieved using two plies of CSM fiber glass. The servos were placed between the two base plates and so the height was adjusted according to the servo motor MG996R (26.6mm).

4.3 Upper leg

The upper leg or femur is the middle part of our robot which is connected on one side to the base plate and to the lower leg on other side. It also holds the servo motors on either side. So its design had to be one which was sturdy enough to bear the weight. As the servo motors were placed inside the femur it had to be designed so that the motors fit perfectly and did not budge. The design also had to be one which did not provide any hinderance to the movement of the motors. The servo motor MG996R has two sides including a flat side and the other one is the power side.

For the size of the femur its height and length was according to the servo motor and its attachments. As the servo motor has to be fit in the femur there is a attachment mechanism that was built. The holes in the walls of the femur is for that attachment

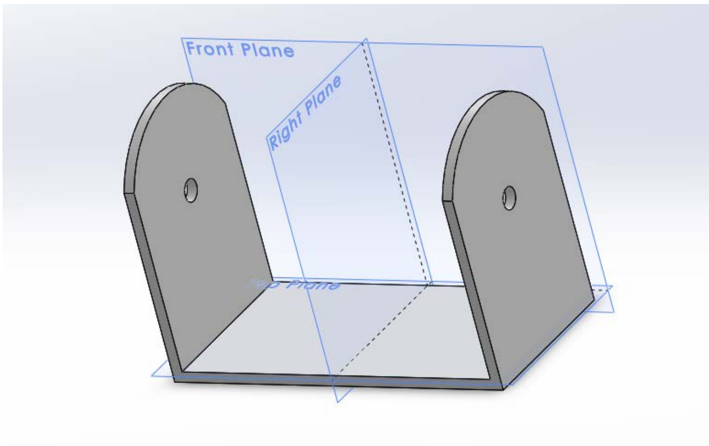

Figure 4.3: Upper Leg

4.4 Lower leg

The lower leg or tibia is the part of the robot that comes in contact with the ground or any other surface that the robot traverses on. There were several design options for this part but we designed one which was as light as possible and also provided us with the strength we needed. We cut out a major chunk from the center of the tibia to make it light weight. The second servo motor was attached between the femur and tibia so its height was adjusted according to that. The thickness of the sheet from which the tibia was made was the same from which we had made the other parts.

The basic construction included cutting out two flat sides of the lower leg and making them hollow from within so that it was as lightweight as possible. Ultimately, these two sides were joined together.

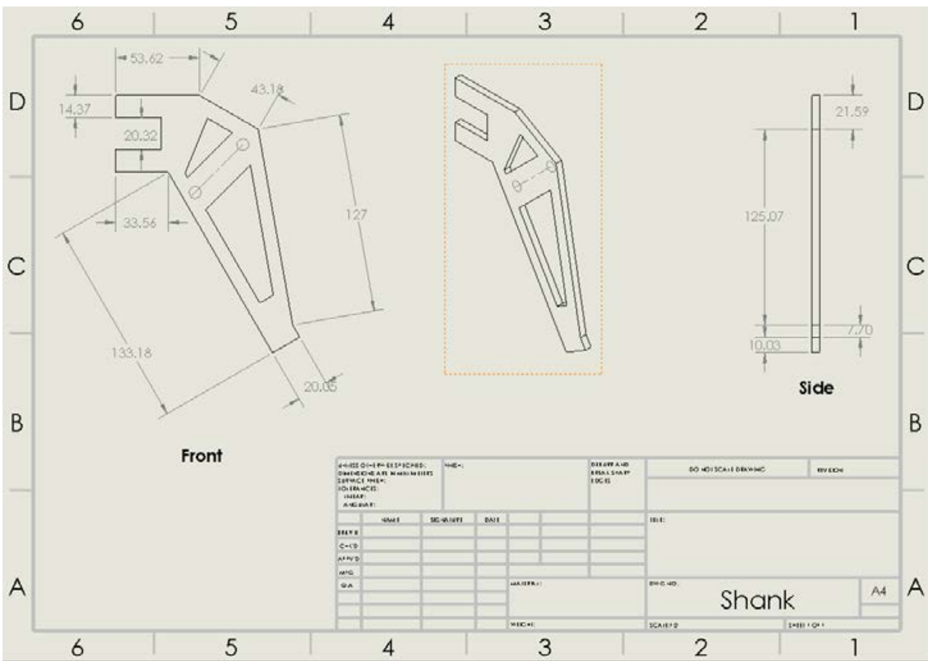

Figure 4.4: Dimensions of lower leg in SolidWorks

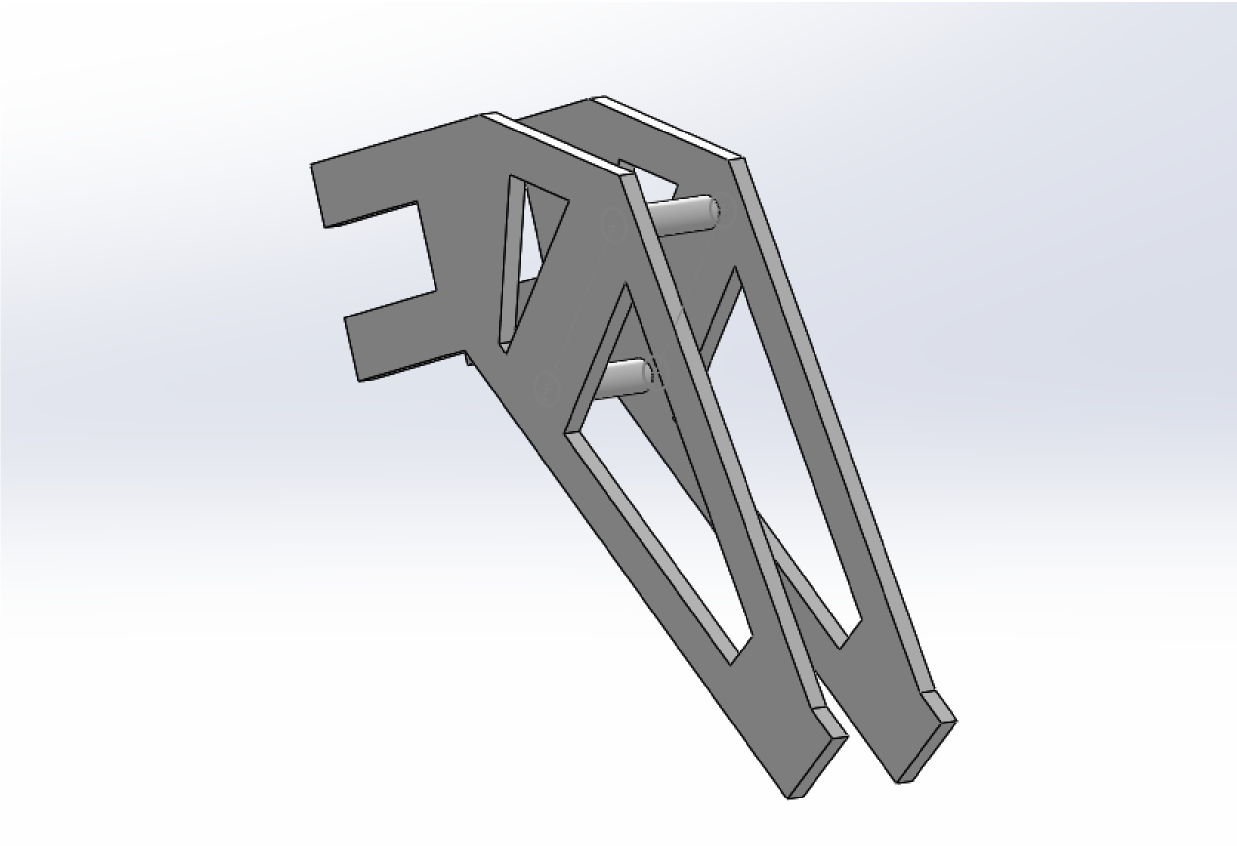

Figure 4.5: Lower leg

4.5 Servo motor attachments

The servo motor was used to transfer the power and as a result the parts moved. There are two sides of a servo motor:

1. Flat side
2. Power side

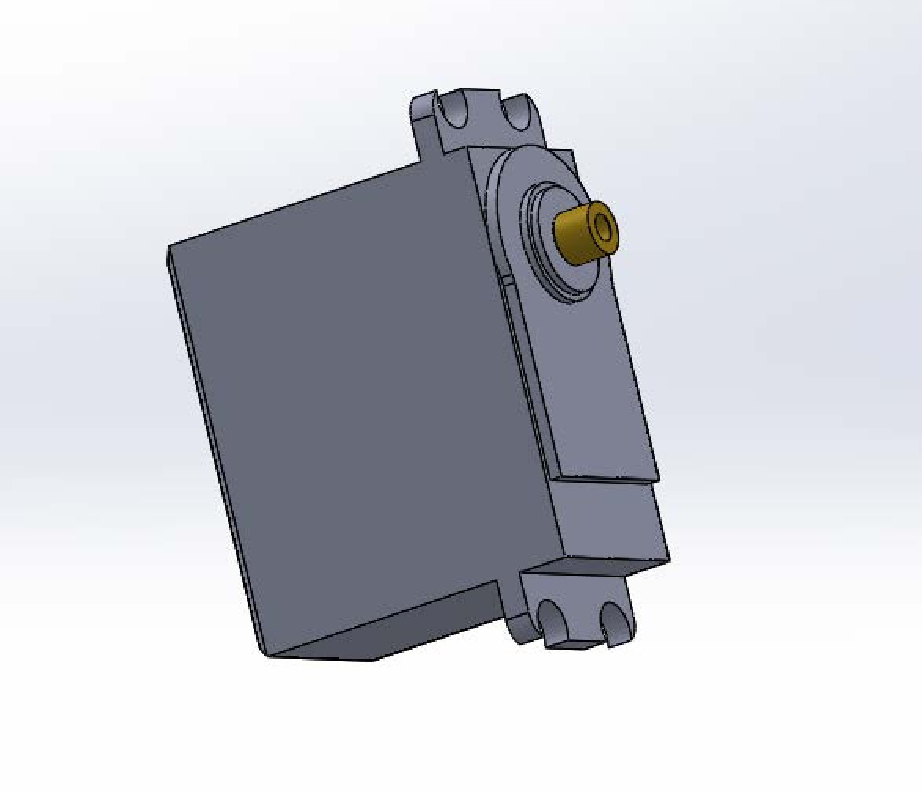

Figure 4.6: Servo Motor

Figure 4.7: Servo motor attachments

At the point when the servo engines were purchased they had accompanied worked in connectors of different shapes and a couple of connections through which the force side of the servo engine was joined to the femur. For the level side a metal ball was to make an association of servo engine with the femur. The metal roller was connected to a heading shaft and was made to fit a material bearing seat.

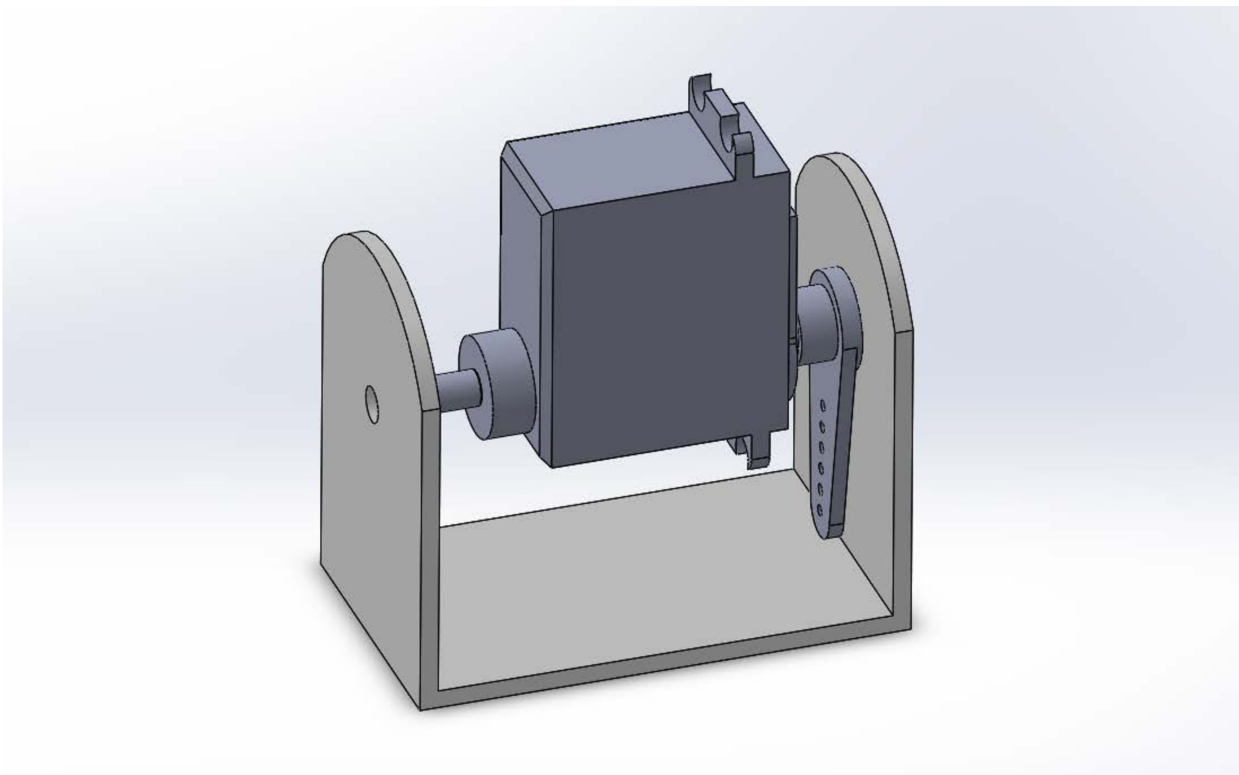

Figure 4.8: Servo motor attached to the upper leg via attachments

CHAPTER 5: ELECTRICAL DESIGN

5.1 Servo Motor

We as a whole realize that an engine changes over electrical vitality into mechanical. As this is the fundamental necessity of our entire automated structure we set foot to locate an engine that most appropriate our venture. Presently the three primary sorts of DC motors are:

1. Standard DC motor
2. Servo motor
3. Stepper motor

Despite the fact that the standard DC engine is simpler to utilize and is less expensive when contrasted with different engines we were unable to work with it as its yield is a ceaseless revolution and we didn't need that. The issue with stepper engine was that in spite of the fact that it furnished us with both consistent and fixed edge turn it required an additional driver IC and a microcontroller and we didn't need an additional problem. Servo engine as we probably am aware gives a fixed edge revolution and is programmable so we chose to settle on it our engine of decision.

Essentially, a servo engine is utilized to control exact development that could be direct or rakishposition, speed or increasing speed. Here is an outline that clarifies the essential working of a servo engine.

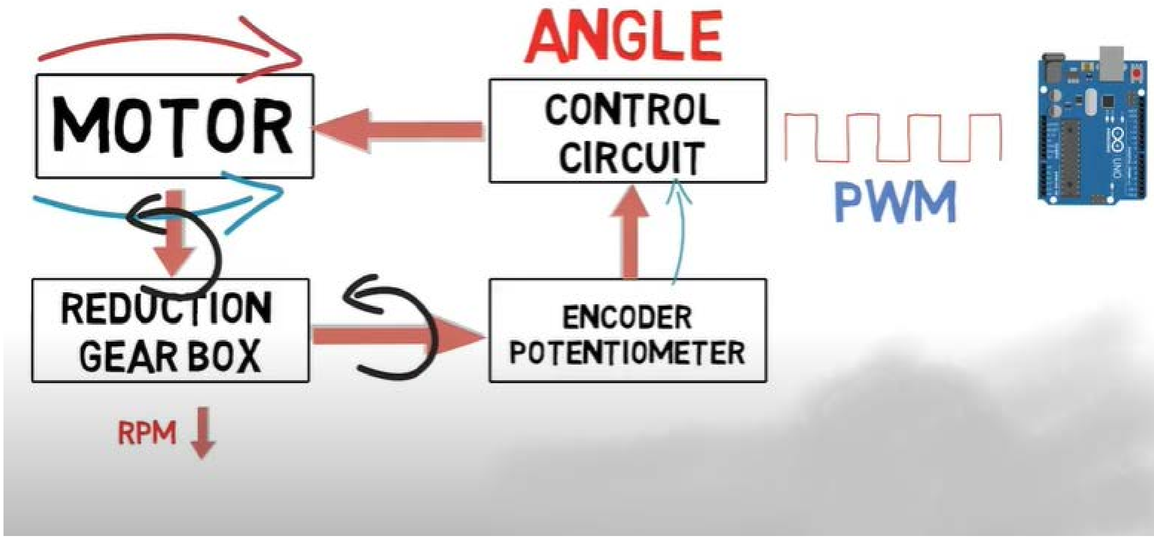

Figure 5.1: Internal mechanism of a servo motor

The microcontroller gives the control circuit a PWM signal. The control circuit decodes the PWM signal and gets the degrees to which it has to rotate. The motor and controlcircuit are connected so the motor moves clockwise or counter clockwise and so the output shaft moves. The encoder potentiometer gives feedback signal to the control circuit and thus the motor stops.

The most extreme Torque necessities for the automated structure as broke down in SolidWorksMotion study came out to be 0.5 Nm or 5 kg/cm. The most widely recognized servo engines for automated controllers are SG-90, NRs-995, MG-90s and a couple of others. The reason of us not choosing any of these motors was the fact that none of these fulfilled our Torque requirements. Although a few motors did provide us with the Torque but they weren’t much cost effective. So ultimately to get the best of both worlds we chose servo motor MG996R which made our project cost effective and also fulfilled the Torque requirements giving us a total of 11kg/cm stall Torque.

The MG996R servo motor is a high torque servo motor which has a metal gearing so it gives exceptionally high value of stalling torque although the size is not that huge. It is an upgrade of the already existing MG995 servo motor, the issue with which was that it was not shock proof and also the system of PCB and IC control was not that great. The upgradation of all this makes MG996R a much better version. MG996R comes with multiple attachments and fasteners which makes it really practical to use

Figure 5.2: Servo motor MG996R

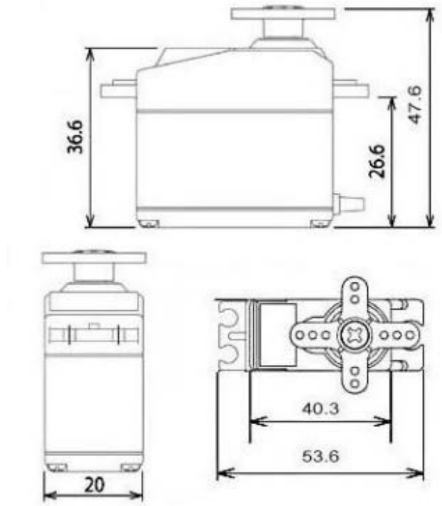

Figure 5.3: Dimensions of MG996R

MG996R Servo Motor Features

- Operating Voltage is +5V typically
- Current: 2.5A (6V)
- Stall Torque: 9.4 kg/cm (at 4.8V)
- Maximum Stall Torque: 11 kg/cm (6V)
- Operating speed is 0.17 s/60°
- Gear Type: Metal
- Rotation : 0°-180°
- Weight of motor : 55gm
- Package includes gear horns and screws

5.2 Arduino

- As we as a whole know Arduino is a stage that has its utilization of building electronic undertakings, likewise it is an open-source stage. The plan has changed totally as the years progressed, and a few varieties incorporate different parts too. Be that as it may, on fundamental board, you will locate the accompanying pieces:
- A number of pins, which are utilized to interface with different segments you should use with the Arduino. Arduinos have computerized and simple pins to enter and speak to information. Computerized pins peruse or compose, ON or OFF a solitary state while simple pins can understand values. These pins are masterminded in a particular example, so that in the event that you purchase an extra board intended to fit into them, commonly called a "shield," it should fit into most Arduino-good gadgets without any problem.
- A power connector, which gives capacity to both the gadget itself, and gives a low voltage which can control associated parts like LEDs and different sensors, gave their capacity needs are sensibly low. The force connector can interface with either an AC connector or a little battery.
- A microcontroller, which is the fundamental chip in the Arduino that is customized and the projects are executed and handled. In spite of the fact that with each extraordinary Arduino these chips shift yet for the most part these chips have a place with the Atmel family including Atmel controllers, typically an ATmega8, ATmega168, ATmega328, ATmega1280, or ATmega2560

- A sequential connector, which on most more current sheets is executed through a standard USB port. This connector permits you to impart to the board from your PC, just as burden new projects onto the gadget. As a rule Arduinos can likewise be fueled through the USB port, evacuating the requirement for a different force association. A variety of other small components like an oscillator and/or a voltage regulator, which provide important capabilities to the board, although you typically don't interact with these directly; just know that they are there

  To program an Arduino and make it fit for functionality there is an open-source application that is used to program the Arduinos

As we have seen that the servo motors work by getting a PWM signal from the microcontroller so we need PWM channels to control all the servos. This is basic criteria on which we will select our Arduino. The most common Arduinos Arduino UNO and Arduino NANO do not fulfil this requirement of ours so instead we used Arduino Mega 2500 as it has 15 PWM channels that fit perfectly with our requirement.

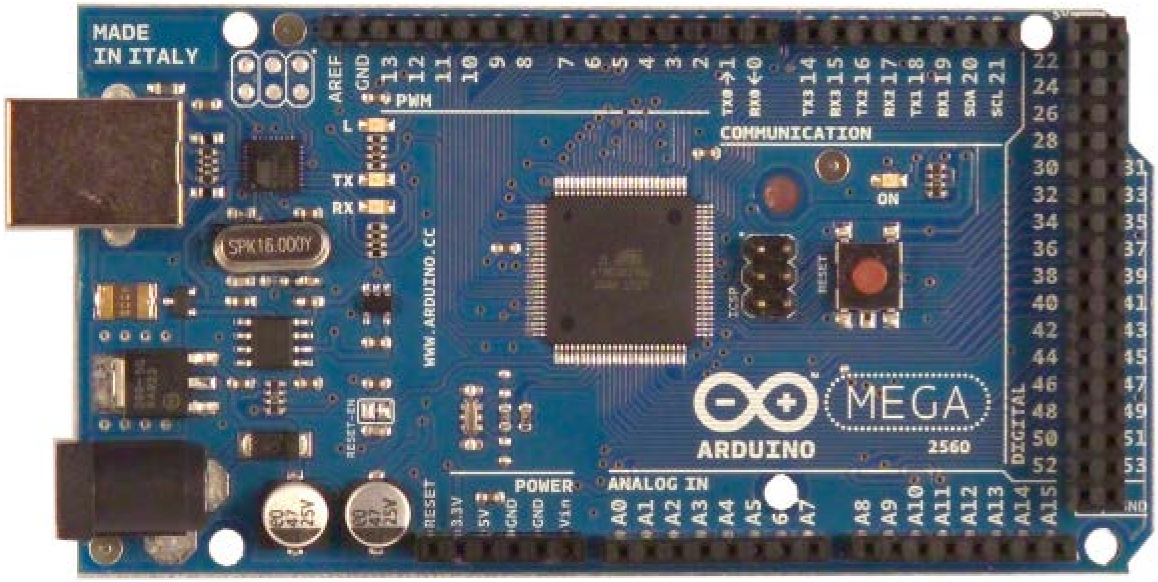

Figure 5.4: Arduino Mega 2500

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Table 5.1: Technical specifications of Arduino Mega 2500

5.3 Battery

The most common batteries that are available in the market having their own sets of pros and cons are Li-Po battery, Li-Fe battery, lead acid battery and Ni-MH battery. These batteries come with individual cells of different voltages and can be attached in series or parallel as per desire to get the wanted output.

The most important factors that were to decide the battery of our choice were most importantly the operating voltage of the servo motors and the discharge time of the batteries Beginning with the working voltage of the servo engines, as referenced over the working voltage of our servo engines ranges from 5-7 V. So we chose the LI-Po battery which is a lightweight battery with great vitality thickness and has a solitary cell of 1.2 V. We utilized 6 such cells causing the all out voltage to up to 7.2 V. Presently this voltage was a piece on the higher side so we utilized a buck converter to restrict to 6.6 V which is a sheltered range for our servo engines.

Presently another significant factor for a battery as expressed before is the release time as it informs a ton regarding the proficiency of the engine, which is essentially the electric charge partitioned by the complete flow.

$$dischargetime = \frac{electriccharge}{current}$$

The battery we chose had a watt rating of 1700 mAh basically meaning it could drive a servo motor consuming 1.7 Amperes for 1 hour. Furthermore, a single servo motor consumed 0.9 Amperes so the total current became 8*0.9= 7.2 A

$$t = \frac{1.7}{7.2} = 14.16\, mins$$

Hence as mentioned above the selection of our battery Li-po, 6.6V, 1700mAh is justified as it will efficiently run our motors as well as give us a good discharge time

Figure 5.5: Li-Po Battery

5.4 Bluetooth Module

When it came down to giving instructions to the robot we had the options of making it wired or wireless. Wired communication was a much easier option as it only demanded connection the Arduino with the laptop but this is not the most practical decision. Going with the wireless option demanded some extra effort but we still went with it as it increased the feasibility of our robot.

For this reason we utilized a Bluetooth module which included two-way (full-duplex) remote usefulness to our robot. Basically power the module with +5V and associate the Rx pin of the module to the Tx of MCU and Tx pin of module to Rx of MCU as appeared in the figure underneath

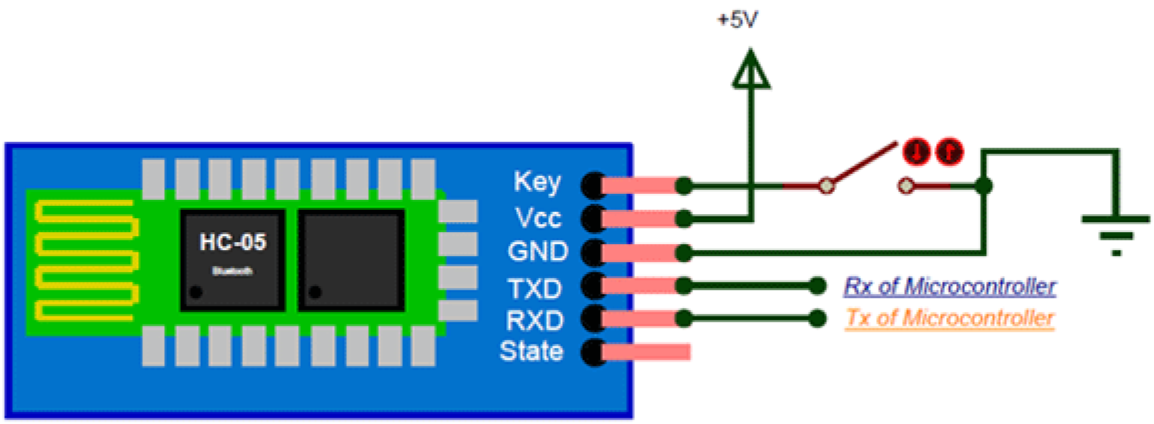

Figure 5.6: Bluetooth Module HC-05 pin description

HC-05 Default Settings

Default Bluetooth Name: “HC-05”

Default Password: 1234 or 0000

Default Communication: Slave

Default Mode: Data Mode

Data Mode Baud Rate: 9600, 8, N, 1

Command Mode Baud Rate: 38400, 8, N, 1

Default firmware: LINVOR

HC-05 Technical Specifications

Serial Bluetooth module for Arduino and other microcontrollers

Working Voltage: 4V to 6V (Typically +5V)

Working Current: 30mA

Range: <100m

Works with Serial correspondence (USART) and TTL perfect

Follows IEEE 802.15.1 normalized convention

Utilizations Frequency-Hopping Spread range (FHSS)

Can work in Master, Slave or Master/Slave mode

Can be effectively interfaced with Laptop or Mobile telephones with Bluetooth

Bolstered baud rate: 9600,19200,38400,57600,115200,230400,460800.

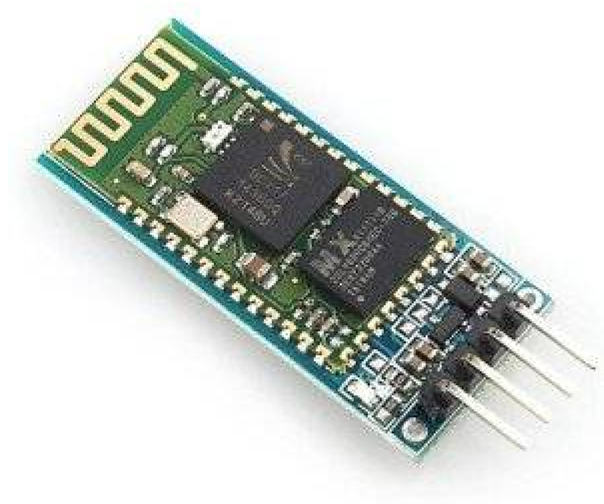

Figure 5.7: Bluetooth module HC-05

5.5 Buck Converter

We used buck converter to step our voltage down from 7.2 to 6.6 V. A buck converter is a type of converter that performs DC to DC conversions and is very simple in construction and working. What it does is that it delivers a yield estimation of voltage that is less that the worth given at the information. It is likewise given its name in light of the fact that the inductors demonstration against or buck the estimation of info voltage. As we realize that V and I are in backwards connection the Buck converter is a DC to DC converter that means up the current from contribution to yield or from gracefully to load and steps down the voltage from contribution to yield or from flexibly to stack.

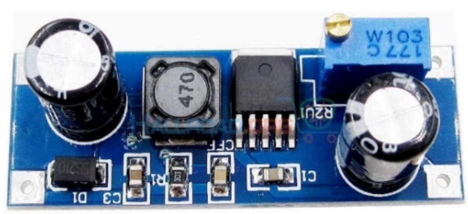

Figure 5.8: Buck Converter

5.6 Camera

As we know, the main incentive of our project was the communication with the base station via a payload, which would send audio and audio signals back to the base station from a remote destination. For the image, video and audio communication the device that we used was “ Mini Spy IP Camera ”. The camera came with a user manual, a USB charger with micro USB port, the camera itself and a camera stand.

The camera itself was equipped with an OFF/ON button, power indicator, mic, reset password, SD card slot, charging port, QR code and a magnetic installing for attaching the stand.

The main attractions in the camera were:

- Wireless communication
- HD display (1080 P)
- 25 TF card storage
- USB interface
- QR scanning

The camera provided us with high quality image, video and motion detection results. The application that we used for the camera was HDSP cam which was connected to the camera using Wi-Fi.

Figure 5.9: Mini spy IP camera

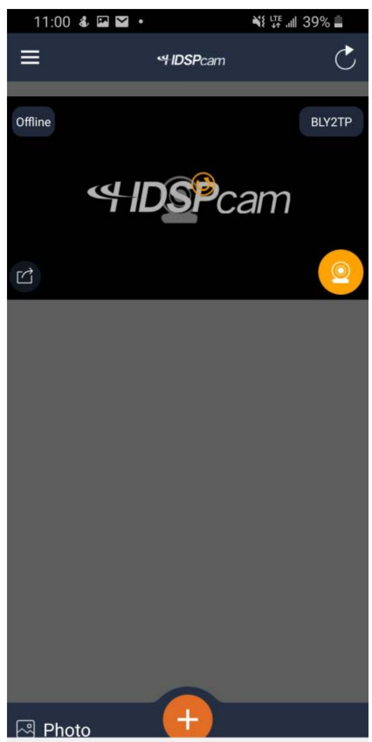

Figure 5.10: User Interface for HDSP cam application

CHAPTER 6: EVALUATION AND CONTROL

6.1 Overview

The task in hand of designing and fabricating a four-legged robot with spider analogy hence the name SpidoBot is completed successfully. The incentive of our robot which is to be useful in areas where human reach is inaccessible is in full functionality. As the robot will be controlled via Bluetooth module through an application the robot is able to traverse in farfetched areas and provide the information of the surroundings to the base station via the payload mounted on it. The camera would take a view of the vicinity and we will be able to take a view of the live feed of that area. The spidoBot is an agile robot capable of performing the two phenomenon i.e. walking and turning efficiently. So it is capable of reaching a particular destination. It traverses in a way that at any point two out of the four legs perform an identical motion be it in contact with the ground or in the air. If we talk about the two legs in contact with the ground they push the ground in a backward motion and as a result of that action the robot moves forward in reaction. If we talk about the two legs in the air, they complete a distance in the air and when the time of their contact with the ground comes they reach the starting point of the circle. For turning motion of the robot one of the two legs pushes while the other pulls and as a result the robot turns along its own vertical central axis.

6.2 Block Diagram

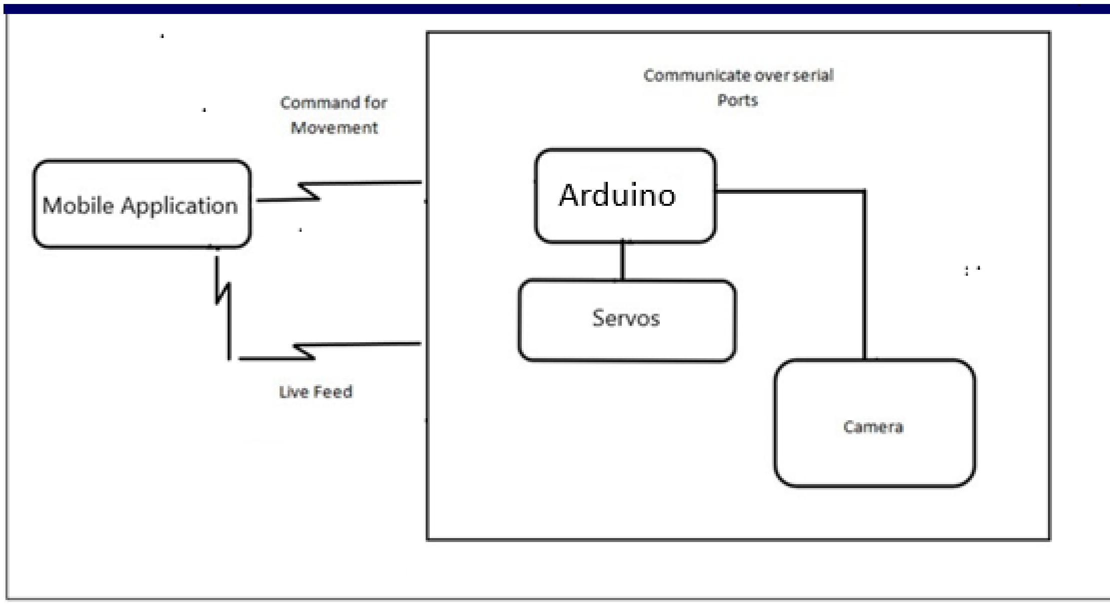

Figure 6.1: Block Diagram of SpidoBot

6.3 Control of the Robot

The main component that plays a vital role in controlling SpidoBot is the Arduino Mega 2560. The main purpose of the Arduino is the simultaneous control of the 8 servo motors. Figure 6.2 shows a schematic diagram of the Arduino, servo motors and the battery. The specific battery is providing the required power that the servo motors need to function. In the Schematic Diagram shown in the figure 6.1 the yellow wires are taking signals and sending them to the servos from the PWM channels of Arduino hence called the signal wires. As discussed in the earlier chapters we know that the PWM signals control the servo motors. The red wires are the live wires whereas the black ones are showing either the negative side of the battery or the ground of the servos. The control of the servos was done using Arduino programming

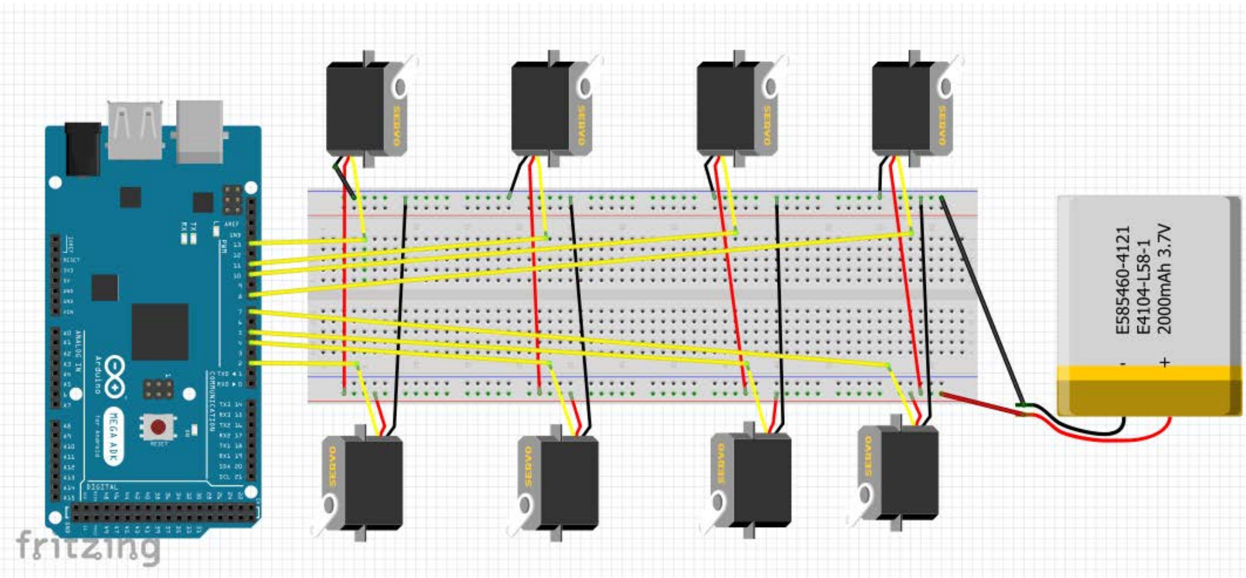

Figure 6.2: Schematic Diagram of SpidoBot

6.4 Arduino Code

```
char letter;
#include<SoftwareSerial.h>
SoftwareSerial bt(0,1); // assigning 0 as RX ans 1 as TX
#include <Servo.h>
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;
Servo myservo5;
Servo myservo6;
Servo myservo7;
Servo myservo8;
void setup()
{
 Serial.begin(9600);
myservo1.attach(10); //Pwm Channel 10
myservo2.attach(8); // Pwn channel 8
myservo3.attach(7); // Pwm channel 7
myservo4.attach(5); // Pwm channel 5
myservo5.attach(2); // Pwm channel 2
myservo6.attach(4); // Pwm channel 4
myservo7.attach(11); //Pwm channel 11
myservo8.attach(13); //Pwm channel 13
}
void loop()
{
 if(Serial.available())
 Serial.println("Start walking");
 letter= Serial.read();
 Serial.println(letter);
 if(letter =='F') // setting alphabet F for straight walking mode
```

```
{
Serial.println("Straight"); // print straight on serial monitor
for (int i=0; i<21; i++)
{
// For both motors of leg 1
int a[21]= {99, 98, 98, 88, 97, 97, 95, 94, 94, 92, 91, 91, 90,89, 89, 88, 87, 86, 85, 84, 83};
int b[21]= {132, 132, 132, 132, 132, 132, 132, 132, 132, 132,132, 132, 132, 132, 132, 132, 132,
132, 132, 132, 132};
// For both motors of leg 2
int c[21]= {99, 100, 100, 101, 102, 102, 103, 104, 104, 105, 106, 106, 107,108, 108, 109, 110,
111, 112, 113, 114};
int d[21]= {121, 121, 121, 121, 121, 121, 121, 121, 121, 121,121, 121, 121, 121, 121, 121, 121,
121, 121, 121, 121};
//For both motors of leg 3
int e[21]= {83, 84, 84, 85, 86, 86, 87, 88, 89, 90, 91, 91,92, 93, 93, 94, 95, 96, 97, 98, 99};
int f[21]= {66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 84,82, 80, 78, 76, 74, 72, 70, 68, 66};
//For both motors of leg 4
int g[21]= {104, 103, 103, 102, 101, 101, 100, 99, 99, 98, 97, 96,95, 94, 93, 92, 91, 90, 89, 88,
87};
int h[21]= {121, 119, 117, 115, 113, 111, 109, 107, 105, 103,101,103, 105, 107, 109, 111, 113,
115, 117, 119, 121};


myservo2.write (a[i]);
delay (2);
myservo1.write (b[i]);
delay (2);

myservo3.write (d[i]);
delay (2);
myservo4.write (c[i]);
delay (2);
```

```
 myservo5.write (e[i]);
 delay (2);
 myservo6.write (f[i]);
 delay (2);

 myservo7.write (g[i]);
 delay (2);
 myservo8.write (h[i]);
 delay (2);

 }
for (int j = 0; j < 21; j++)
 {
 // For both motors of leg 1
 int k[21]= {83, 84, 84, 85, 86, 86, 87, 88, 89, 90, 91, 91,92, 93, 93, 94, 95, 96, 97, 98, 99};
 int m[21]= {132, 130, 128, 126, 124, 122, 120, 118, 116, 114, 112,114, 116, 118, 120, 122, 124,
126, 128, 130, 132};
 // For both motors of leg 2
 int n[21]= {116, 115, 115, 114, 113, 113, 112, 111, 111, 110,109, 109, 108, 107, 106, 105, 104,
103, 102, 101, 100};
 int o[21]= {121, 119, 117, 115, 113, 111, 109, 107, 105, 103,101,103, 105, 107, 109, 111, 113,
115, 117, 119, 121};
 // For both motors of leg 3
 int p[21]= {99, 98, 98, 88, 97, 97, 95, 94, 94, 92, 91, 91, 90,89, 89, 88, 87, 86, 85, 84, 83};
 int q[21]= {66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66,66, 66, 66, 66, 66, 66, 66, 66, 66};
 // For both motors of leg 4
 int r[21]= {88, 89, 89, 90, 91, 91, 92, 93, 93, 94, 95, 96,97, 98, 99, 100, 101, 102, 103, 104,
105};
 int s[21]= {121, 121, 121, 121, 121, 121, 121, 121, 121, 121,121, 121, 121, 121, 121, 121, 121,
121, 121, 121, 121};

 myservo2.write (k[j]);
```

```
myservo1.write(m[j]);
delay(2);

myservo4.write (n[j]);
delay (2);
myservo3.write(o[j]);
delay(2);

myservo5.write (p[j]);
delay (2);
myservo6.write(q[j]);
delay(2);

myservo7.write (r[j]);
delay (2);
myservo8.write(s[j]);
delay(2);

}

}
else if(letter == 'T') // setting letter T for turning
{
Serial.println("Turningg");
for (int i=0; i<21; i++)
{
// For both motors of leg 1
int aa[21]= {67, 68, 70, 72, 74, 76, 77, 78, 79, 80, 81, 82,83,84, 85, 86, 88, 90, 92, 94, 96};
int bb[21]= {132, 130, 128, 126, 124, 122, 120, 118, 116, 114, 112,114, 116, 118, 120, 122,
124, 126, 128, 130, 132};
// For both motors of leg 2
int cc[21]= {67, 68, 70, 72, 74, 76, 77, 78, 79, 80, 81, 82,83,84, 85, 86, 88, 90, 92, 94, 96};
```

```
int dd[21]= {121, 119, 117, 115, 113, 111, 109, 107, 105, 103,101,103, 105, 107, 109, 111, 113,
115, 117, 119, 121};
 // For both motors of leg 3
 int ee[21]= {142, 140, 138, 137, 136, 134, 133, 132, 130, 129,128, 126, 125, 124, 122, 120, 118,
117, 116, 114, 112};
 int ff[21]= {66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66, 66,66, 66, 66, 66, 66, 66, 66, 66, 66};
 // For both motors of leg 4
 int gg[21]= {121, 123, 124, 125, 126, 127, 129, 130, 132, 134,136,137,138, 139, 140, 141, 142,
144, 146, 148, 150};
 int hh[21]= {132, 132, 132, 132, 132, 132, 132, 132, 132, 132,132, 132, 132, 132, 132, 132,
132, 132, 132, 132, 132};
 myservo2.write (aa[i]);
 delay (2);
 myservo1.write (bb[i]);
 delay (2);

 myservo3.write (dd[i]);
 delay (2);
 myservo4.write (cc[i]);
 delay (2);

 myservo5.write (ee[i]);
 delay (2);
 myservo6.write (ff[i]);
 delay (2);

 myservo7.write (gg[i]);
 delay (2);
 myservo8.write (hh[i]);
 delay (2);
}
for (int i=0; i<21; i++)
```

```
 // For both motors of leg 3
 int kk[21]= {112, 114, 116, 117, 118, 120, 122, 124, 125, 126,128, 129, 130, 132, 133, 134,
136, 137, 138, 140, 142};
 int mm[21]= {66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 84,82, 80, 78, 76, 74, 72, 70, 68, 66};
 // For both motors of leg 1
 int nn[21]= {96, 94, 92, 90, 88, 86, 85, 84, 83, 82, 81, 80,79, 78, 77, 76, 74, 72, 70, 68, 67};
 int oo[21]= {132, 132, 132, 132, 132, 132, 132, 132, 132, 132,132, 132, 132, 132, 132, 132,
132, 132, 132, 132, 132};
 // For both motors of leg 2
 int pp[21]= {96, 94, 92, 90, 88, 86, 85, 84, 83, 82, 81, 80,79, 78, 77, 76, 74, 72, 70, 68, 67};
 int qq[21]= {121, 121, 121, 121, 121, 121, 121, 121, 121, 121,121, 121, 121, 121, 121, 121,
121, 121, 121, 121, 121};
 // For both motors of leg 4
 int rr[21]= {150, 148, 146, 144, 142, 141, 140, 139, 138, 137,136, 134, 132, 130, 129, 127, 126,
125, 124, 123, 121};
 int ss[21]= {132, 130, 128, 126, 124, 122, 120, 118, 116, 114, 112,114, 116, 118, 120, 122, 124,
126, 128, 130, 132};

myservo2.write (kk[j]);
 delay (2);

myservo1.write (mm[j]);
 delay (2);

myservo3.write (nn[j]);
 delay (2);

myservo4.write (oo[j]);
 delay (2);

myservo5.write (pp[j]);
 delay (2);
```

```
myservo6.write (qq[j]);
 delay (2);

myservo7.write (rr[j]);
 delay (2);

myservo8.write (ss[j]);
 delay (2);
 }

 }
 else if(letter == 'B')
 {
 }
}
```

6.5 The final SpidoBot

Figure 6.3: SpidoBot

CHAPTER 7: FUTURE WORK

SpidoBot is designed in such a fashion that it is able to traverse over multiple terrains including terrains which are inaccessible to humans. This is possible due to its eight degree freedom of movement, its structure and material. Although it can work very well in the areas it is designed for yet there are a couple of domains where improvements can be made to overall better its performance

7.1 Electrical Design

1) Addition of multiple sensors can increase the practicality of SpidoBot as it will make it more adjustable in even worse environments
2) Even stronger servo motors with higher value of Torque can be used to better the motion of SpidoBot
3) A specific mobile application can be built for The SpidoBot for marketingpurposes

7.2 Mechanical Design

1) The degrees of freedom could be increased from 8 to 12 to make it even more stable and agile
2) The mechanical design could be made even better by making it even lighter
3) Carbon fiber instead of Fiber glass can be used to make the body stronger and harm proof
4) The gait size of the robot could be increased to make it more efficient

References:

[1] https://www.sciencefriday.com/segments/the-origin-of-the-word-robot/

[2] Robot Platform | Knowledge | Wheeled Robotshttp://www.robotplatform.com/knowledge/Classification_of_Robots/legged_robots.html

[3] https://www.asminternational.org/documents/10192/1849770/05287G_Sample_Chapter.pdf

[4] https://www.intechopen.com/online-first/introduction-to-composite-materials

[5] Fabrication methods : CompositesWorld

https://www.compositesworld.com/articles/fabrication-methods

[6] A study on locomotion of quadruped robots| https://www.researchgate.net/publication/292850572_A_Study_on_Locomotions_of_Quadruped_Robot

[7] Designing, 3D printing of a quadruped robot and choice of materials for fabrication| Akash Maity, Koustav Roy, Dhrubajyoti Gupta

https://www.researchgate.net/publication/333842118_Designing_3D_Printing_of_a_Quadruped_Robot_and_Choice_of_Materials_for_Fabrication

[8] Kinematics Analysis of a Quadruped Robot| Alain SegundoPotts*Jose Jaimeda Cruz**

https://www.sciencedirect.com/science/article/pii/S1474667015374917

[9] DESIGN OF A QUADRUPED ROBOT AND ITS INVERSE KINEMATICS |PALIVELA ARUN KUMAR & YEOLE SHIVRAJ NARAYAN

https://www.researchgate.net/publication/318761363_DESIGN_OF_A_QUADRUPED_ROBOT_AND_ITS_INVERSE_KINEMATICS

[10] Gait and Stability Analysis of a Quadruped Robot| Junming Wei, Zhiguo Shi

https://link.springer.com/chapter/10.1007/978-3-642-21402-8_56

[11]. Akbas, T., Eskimez, S.E., Ozel, S., Adak, O.K., Fidan, K.C., and Erbatur, K. Zero Moment Point Based Pace Reference Generation for Quadruped Robots via Preview Control.

[12]. Gao, J., Zhang, S., Duan, X., Li, H., Yu, Z., Chen, X. Li, J., Liu, H. LI, X., Liu, Y., and

Xu, Z. Trot Pattern Generation for Quadruped Robot Based on the ZMP Stability Margin.

International Conference on Complex Medical Engineering. Beijing, China.

Bibliography

Current ROBOTICS MECHANICS, PLANNING, AND CONTROL by Kevin M. Lynch and Frank C. Park

An Introduction to Composite Materials (second ed.). NY: Cambridge University Press Hull, D., and Clyne, T.

PLAGIARISM REPORT

- Processed on: 17-Jul-2020 19:40 PKT
- ID: 1358637760
- Word Count: 8600
- Submitted: 2

222 By Hassan Nawaz

Similarity Index

15%

Similarity by Source

Internet Sources:
7%
Publications:
0%
Student Papers:
14%

3% match (student papers from 02-May-2017)
Submitted to University of Wolverhampton on 2017-05-02
2% match (student papers from 29-Nov-2018)
Submitted to Higher Education Commission Pakistan on 2018-11-29
1% match (student papers from 20-Apr-2020)
Submitted to Institute of Technology Carlow on 2020-04-20
1% match (Internet from 14-Nov-2018)
http://www.megahard.pro/design-portfolio/spider-robot
1% match (student papers from 13-Jul-2020)
Submitted to Higher Education Commission Pakistan on 2020-07-13
1% match (student papers from 03-Dec-2017)
Submitted to University of Lancaster on 2017-12-03
1% match (Internet from 16-Jul-2020)
https://components101.com/motors/mg996r-servo-motor-datasheet
1% match (Internet from 24-Jun-2020)
https://vtcmagblog.com/tag/a-i-aka-ai-artificial-intelligence/
< 1% match (Internet from 26-Sep-2019)
https://opensource.com/resources/what-arduino
< 1% match (student papers from 14-Jun-2012)
Submitted to Higher Education Commission Pakistan on 2012-06-14
< 1% match (student papers from 01-Jul-2019)
Submitted to PSB Academy (ACP eSolutions) on 2019-07-01
< 1% match (student papers from 31-May-2019)
Submitted to University College London on 2019-05-31
< 1% match (student papers from 16-Nov-2012)
Submitted to University of Leeds on 2012-11-16
< 1% match (Internet from 14-Feb-2020)
https://issuu.com/ijitce/docs/april2019
< 1% match (student papers from 18-Jun-2018)
Submitted to RMIT University on 2018-06-18
< 1% match (Internet from 24-Apr-2020)

https://hybridatelier.uta.edu/equipment/53-mekp-liquid-catalyst
< 1% match (student papers from 21-Apr-2014)
Submitted to CSU Northridge on 2014-04-21
< 1% match (student papers from 21-May-2010)
Submitted to University of Central Lancashire on 2010-05-21
< 1% match (Internet from 08-Feb-2020)
http://lukaswoodtli.github.io/rotation_matrix.html
< 1% match (student papers from 02-May-2014)
Submitted to VIT University on 2014-05-02
< 1% match (student papers from 11-Nov-2018)
Submitted to Universiti Teknologi MARA on 2018-11-11
< 1% match (student papers from 01-Nov-2012)
Submitted to King Mongkut's University of Technology Thonburi on 2012-11-01
< 1% match (student papers from 29-Mar-2020)
Submitted to The University of Manchester on 2020-03-29
< 1% match (student papers from 27-Jan-2020)
Submitted to University of Westminster on 2020-01-27
< 1% match (student papers from 25-Oct-2011)
Submitted to Higher Education Commission Pakistan on 2011-10-25
< 1% match (student papers from 18-Jul-2014)
Submitted to The University of the South Pacific on 2014-07-18
< 1% match (student papers from 27-Apr-2013)
Submitted to Coventry University on 2013-04-27
< 1% match (student papers from 07-Mar-2014)
Submitted to Universiti Selangor on 2014-03-07
< 1% match (student papers from 12-Jun-2017)
Submitted to Bahrain Polytechnic on 2017-06-12
< 1% match (student papers from 02-Jul-2020)
Submitted to University of Northampton on 2020-07-02
< 1% match (student papers from 14-Feb-2014)
Submitted to London School of Marketing on 2014-02-14
< 1% match (student papers from 10-May-2017)
Submitted to Heriot-Watt University on 2017-05-10
< 1% match (student papers from 24-May-2019)
Submitted to University College London on 2019-05-24
< 1% match (student papers from 09-Jan-2017)
Submitted to University of Philadelphia - Jordan on 2017-01-09
< 1% match (student papers from 28-May-2014)
Submitted to Higher Education Commission Pakistan on 2014-05-28
< 1% match (student papers from 27-Mar-2012)
Submitted to Royal Melbourne Institute of Technology on 2012-03-27
< 1% match (student papers from 30-Apr-2009)
Submitted to University of Petroleum and Energy Studies on 2009-04-30
< 1% match (publications)
Mohammadhossein Saadatzi, David C. Long, Ozkan Celik. "Comparison of Human-Robot Interaction Torque Estimation Methods in a Wrist Rehabilitation Exoskeleton", Journal of Intelligent & Robotic Systems, 2018
< 1% match (student papers from 03-May-2019)
Submitted to University of Southampton on 2019-05-03
< 1% match (student papers from 05-Jun-2020)
Submitted to American University of the Middle East on 2020-06-05
< 1% match (student papers from 12-Jan-2015)
Submitted to University of Leeds on 2015-01-12