PROJECT THESIS ON:

Construction and Mechanism of a Humanoid Robot leg

PROJECT DONE BY:

Zubair Al-Azad (081436)
Swad-Us-Sami (081444)
Faiyaz Chowdhury (081445)
Muhammad Faisal Uddin Chowdhury (081447)

SUPERVISED BY:

PROF. DR. MD. NURUL ABSAR CHOWDHURY
DEPARTMENT OF MECHANICAL AND CHEMICAL ENGINEERING
ISLAMIC UNIVERSITY OF TECHNOLOGY
BOARD BAZAR, DHAKA, BANGLADESH.

Abstract

This thesis describes designs of the robot-leg mechanisms, hardware and the leg control methods for walking machines. The body of knowledge that applies to mobile wheeled robots is quite well developed. However, autonomous walking vehicles are still relatively new, and the body of knowledge concerning their development is not as well defined. The difficulty factor in building a legged robot is also considerably higher than that for a wheeled robot.

Physical-based control using center of mass, center of pressure, and foot placement is used to enable a simulated twelve -degree of freedom, seven-link, three-dimensional bipedal robot to lean sideways, pick up its foot and start walking on a flat surface.

Energy analysis is used to compel the same simulated robot to do a side-to-side rocking motion and eventually come to a stop. If the robot is pushed hard enough, it will raise its leg that is in the air in the frontal plane to prevent itself from falling.

Center of mass and center of pressure analysis is used to enable the same robot to balance on one foot and stand.

Acknowledgement

We would like to express deep gratitude to our supervisor PROF**. DR. MD. NURUL ABSAR CHOWDHURY** of Mechanical and Chemical Engineering Department of IUT, whose guidance and suggestions enabled us to do this project and attain a practical result.

We also extend our heartiest thanks to all the **mechanical workshop assistants** for providing the technical support in different stages of this project.

Table of Contents

Chapter 1

Introduction

Traditionally, most mobile robots have been equipped with wheels. The wheel is easy to control and direct. It provides a stable base on which a robot can maneuver and is easy to build. One of the major drawbacks of the wheel, however, is the limitation it imposes on the terrain that can be successfully navigated. A wheel requires a relatively flat surface on which to operate. Rocky or hilly terrain, which might be found in many applications as forestry, waste clean-up and planetary exploration, imposes high demands on a robot and precludes the use of wheels. A second approach to this problem would be to use tracked wheel robots. For many applications this is acceptable, especially in very controlled environments. However, in other instances the environment cannot be controlled or predicted and a robot must be able to adapt to its surroundings. Such a surrounding can be placed where robots would have to step over the obstacles such as a surface where pipes are running and where they have to move on discontinuous terrain like steps.

Research into legged robotics promises to overcome these difficulties.

The complexity of control required for a legged robot to navigate autonomously over unfamiliar terrain has made them difficult to build. Recent developments in embedded controller technology have yielded very sophisticated computing devices in relatively small, easily programmed modules. With these advanced components, it is now possible to control relatively complex and sophisticated devices.

Chapter 2

Motivation

Robots possess the ability to relieve us of repetitive, boring and dangerous tasks. They have already found extensive use in the manufacturing industries. Most of these industry robots are large and fixed in position. There is, however, a multitude of tasks that could be accomplished if the robot was mobile and could navigate through its surroundings.

For robots to become mobile requires that they become self-contained. This means that anything they need, like power supply, control hardware etc must be carried with them. This requires mobile robots to be greatly limited in size and weight. Reliable navigation and route planning are incredibly difficult tasks and continue to be the most challenging component of producing autonomous mobile robots.

Our environment is so specialized for our body shape that major difficulties arise to accommodate for non-able-bodied people. When you look at the problems facing wheel chair bound people you can see the constraints placed on them by our environment. Accessibility both inside and outside buildings, in vehicles etc can be severely restricted unless major design changes are implemented specifically for these problems.

Moreover in our country, study of biped is not that familiar.

Chapter 3

History of the Project

The Biped Project is an ongoing project with the ultimate aim of producing a biped robot that can successfully negotiate its way around a human environment. The project has previously been worked on by three students, in the years 1992, 1994, and 1997. Robert Newton [1992] produced the original design and provided the impetus to get the project underway. Newton construction preliminary designs from Lego, and his original prototype can be seen in Figure 1. The robot is designed with twelve degrees of freedom, which corresponds to 12 independently moving joints. There is no torso in
the design and thus each leg contains six joints. Based on the human form, the joints are modeled on the hip, the knee, and the ankle. The hip is free to move in two directions, inwards and outwards in the transverse plane, as well as forwards and backwards in the sagittal plane. The knee joint has only one degree of freedom, that being forwards and backwards in the sagittal plane. The ankle contains three degrees of freedom, forward and back in the sagittal
plane, tilting in the transverse plane, and also horizontal rotation allowing the foot to change direction.

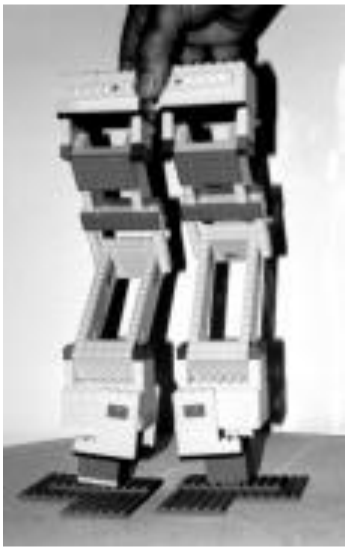


Figure 1. **Original Lego Prototype**

The joints themselves were implemented with DC servo motors, and moments and torque analysis provided the specifications in the selection of these motors. Machined lightweight aluminum was constructed to provide the robot with a skeletal frame. The control hardware designed by Newton, consisted of a token ring of three 80C196KC processors plus I/O boards used to pass control signals to the motors. Figure 2 shows the appearance of the robot at the end of 1992. The project was continued by Conomos [1994]. He found that the MS-747WB motors that were used in all joints except the bottom rotating ankle motors, were prone to slipping due to warn plastic gears. He replaced six of the ten motors with new, metal geared HS-750MG motors. Four motors were not replaced as the new motors were slightly larger and did not fit into the skeletal frame.

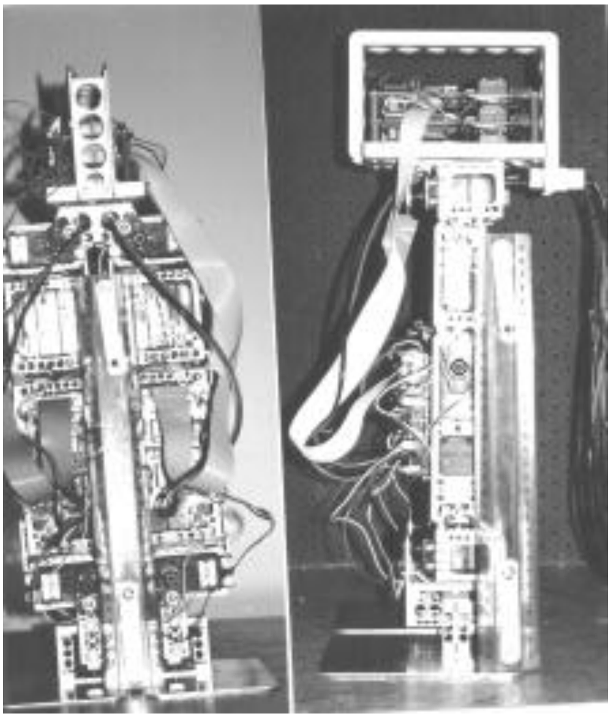


Figure 2. **The biped robot in 1992**

Conomos used the same processor configuration as Newton in his design. He also replaced some of the I/O circuitry, resulting in a more reliable design. However as the token ring control structure communicates serially, the speed of servo control was not going to be fast enough for the accurate control of position.

McManus resumed work on the biped in 1997. His assessment of the situation was as follows:

- The servo motors were still inadequate for the task
- The control software was still too slow
- There was little room for the addition of new sensors
- The theoretical algorithm for walking did not provide an effective practical implementation basis

In order to deliver more power to the motors, McManus constructed new motor driver boards. He based his driver on the H-bridge design which used 2 N-channel and 2 P-channel power mosfets. The design can be seen in
Figure 3.

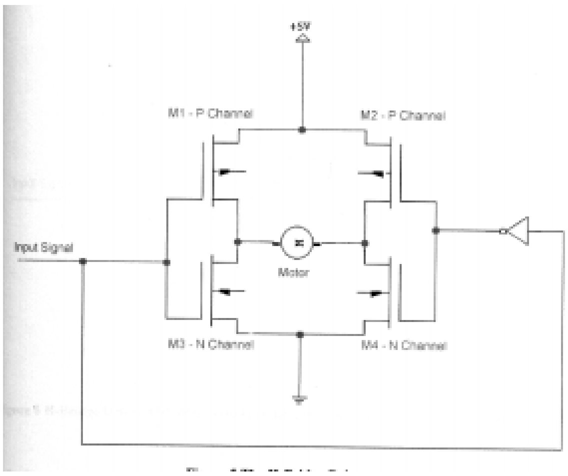


Figure 3. H-bridge motor driver

McManus based his design around the Motorola 68332 processor. This processor can provide a lot of functionality, and includes 16 independent, programmable timer channels. These can be used to easily generate pulse width modulated signals necessary to control the motors. This unit runs separately to the CPU and therefore requires minimal CPU intervention.

McManus also constructed a feedback board, which filtered and amplified the position feedback and torque feedback signals. The position feedback results from measuring the voltage across the servo potentiometer, while the torque feedback is achieved by measuring the current being supplied to the motor. This is possible since the torque of the joint is directly dependent on the current flowing through the motor.

While McManus redesigned and constructed all the control hardware on the robot, he was unable to provide a working demonstration of his design. He found that when he powered up his hardware, the processor would not come out of reset. There are a number of reasons that are responsible for this lack of reliability .

Firstly the soldering of components on the boards was not at a necessary high standard. Also, the supply and distribution of power to the robot appears to be inadequate. When all motors are running, the biped could be drawing current in the vicinity of 10 Amps. This current was being distributed to between boards through very thin wires and tracks.

Furthermore there was no voltage regulation between the power supply and any of the digital components, including the processor.

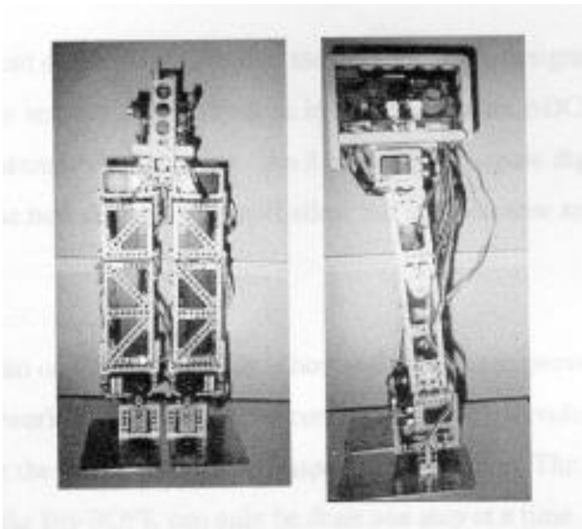


Figure 4. Biped Robot in 1997

Chapter 4

Literature Review

This chapter reviews some of the developments that are currently being made in the field of robotics, with emphasis being placed on topics that relate directly to bipedal robotics. It is important to examine the work of others in order to gain an increased appreciation of the some of the problems that will require solving, as well as some of the ideas that have been proposed to solve them. This chapter firstly covers the selection of robot actuators followed by methods of servomotor control. A number of methods for generating a walk and producing adaptive walking are then examined.

4.1 Servo Control

Servo loops are widely used in robot control programs. The servomotor contains a potentiometer, which is directly connected to the gear train. This allows the position of the servo to be measured and hence the joint can be controlled. Clark [1989] proposes using a Hierarchical Control System (HIC) developed to manage a hierarchy of servo loops. One such layer of the structure can be seen in Figure 6. The input routine obtains readings from the joint potentiometer and converts them (via an A/D converter) to position values that are placed in the Actual Positions buffer. The Planner Routine is then invoked and obtains the Actual Position data and the Target Position data. Applying appropriate control laws to the joint it produces the trajectory data which is stored on the Trajectory buffer. Finally the Output Routine is called which takes the trajectory data from the buffer and produces the necessary control signals to the motor.

The control system could be extended by the addition of higher layers of control. Each layer would be structured the same, with Input, Planner and Output Routines. To apply this system to the biped project, with its twelve servos, one could imagine a bottom layer comprising of twelve

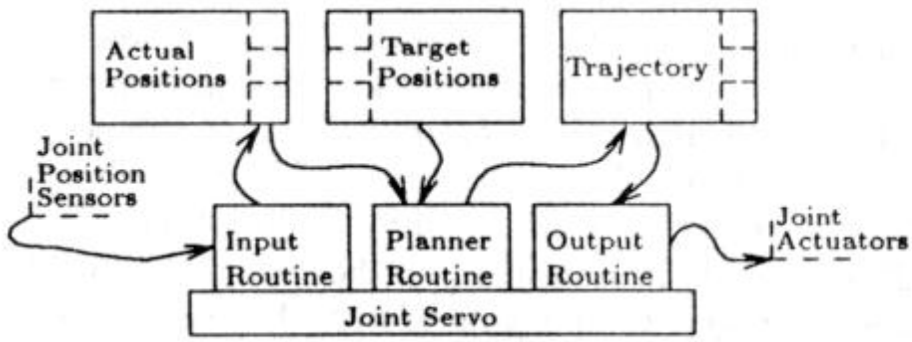


Figure 5. Simple Joint Position Servo Taken from Clark [1989]

structures such as the one shown in Figure 5. Each structure would provide the control for a single servo. Sitting above this layer could be two further structures, one for controlling the six servos in the left leg, and the other for controlling the six servos in the right leg. The Actual Positions data for the six servos in each leg would be fed into the Input Routine of this higher layer. The higher level structure would plan and generate the Target Positions data to be used by the bottom layer. A third level in the hierarchical structure could sit at the top and provide control information to the two leg control structures.

The co-ordination of multiple degrees of freedom to produce a stable walk is complex task by anyone's standard. Stewart and Cremer [1989] suggest that there are two components required in most control schemes: an algorithmic component, which takes a high level goal and generates joint trajectories, and a dynamic component, which takes these joint trajectories and generates the required joint torques. Their paper looks at simplifying the algorithm component by, instead of supplying a complete set of joint trajectories, their algorithm controls other, more intuitive, degrees of freedom. For example, they suggest that it may be desirable to control the overall acceleration of the biped without explicitly controlling each component of the biped.

4.2 Generating a Walk

Stewart and Cremer [1989] use an algorithm that cycles through six states. These states are:

1. Swing the right leg
2. Land the right foot
3. Lift the left leg
4. Swing the left leg
5. Land the left foot
6. Lift the right foot

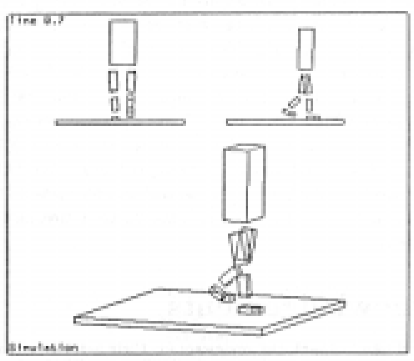


Figure 6. **Simulated Biped Model**
Taken from Stewart and Cremer [1989]

They used a simulator to test their model. Pictures taken from their simulator can be seen in Figures 5 and 6.They have observed that the combination of many small accelerations yields more stable motion than large local accelerations. They believe that the algorithm should greatly under constrain the motion by relying on “passive elements”, such as springs, dampers etc. The algorithm then has the job of guiding rather than forcing the motion of the robot.

4.3 Adaptive Walking

Yamaguchi, Kinoshita, Takanishi and Kato [1996], have produced a dynamic biped walking robot that adapts to the contours of the floor. Utilizing a special foot system they obtain the position relative to the landing surface and its gradient. They employ an adaptive walking control system which takes this information of the surface profile gathered by the foot mechanism and modifies the walk of the robot. The constructional assembly and the link structure of the robot

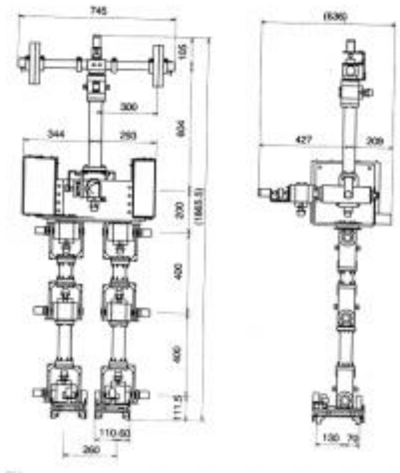


Figure 7. Assembly Drawing of WL-12RVII

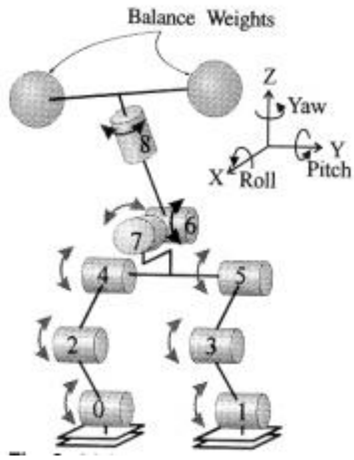


Figure 8. Link Structure of WL-12RVII

Chapter 5

Theoretical Approach

5.1 Model background

The model of this robot has a total of six degrees of freedom, one joint at each hip, one at each
knee, and one at each ankle. We can see from the figure.

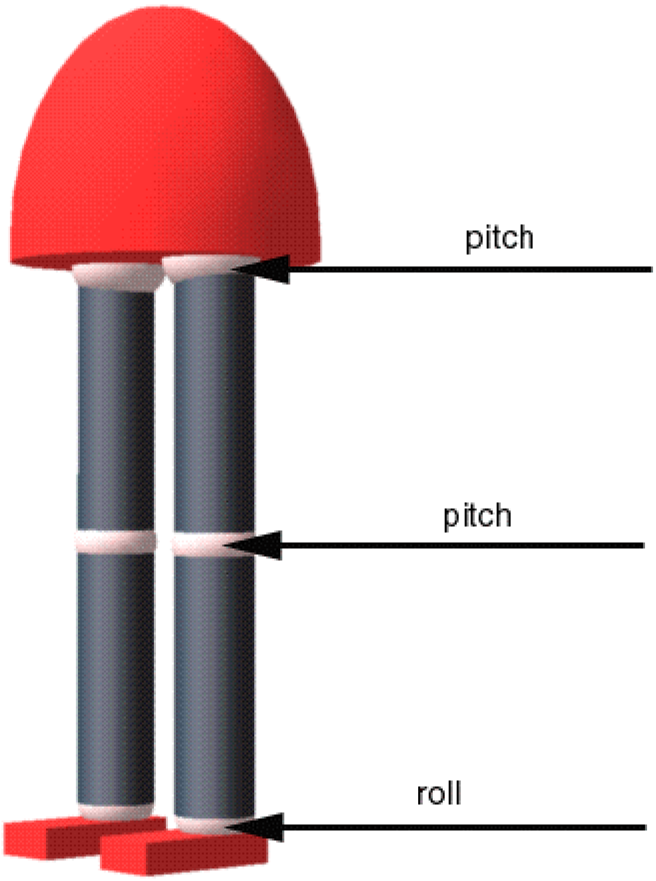


Figure 9: The three-dimensional bipedal robot has one degree of freedom at each hip, one degree of freedom at each knee, and one degree of freedom at each ankle.

5.2 Links Specifications

5.2.1 Feet

The biped model has 2 rectangular feet as shown in Figure

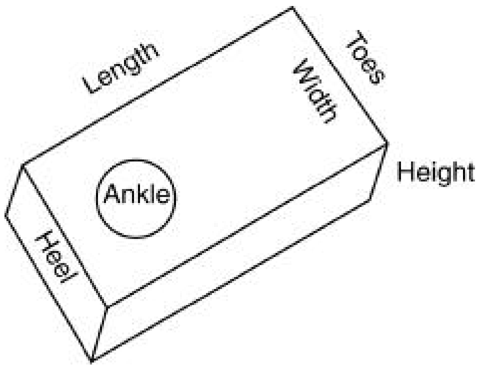


Figure 10: A rectangular foot

The contribution of the feet in natural-looking walking is extremely critical, for instance, if the feet are designed to be too wide, it might result in a very unnatural-looking landing of the foot. Obviously, feet cannot be too narrow either since the side-to-side control would become very challenging. Feet cannot be too long either since foot clearance in the swing phase would be a difficult task to achieve. Feet also play a major role in the toe-off state, where the robot's back-foot pushes against the ground in order for the robot to move forward and go into its opposite single support state, therefore if the feet are too narrowly designed, this task may not be completed successfully as the robot's feet can easily be twisted.

5.2.2 Shins and Thighs

The shins and thighs of the robot are formed of two brackets, two servo holders and two servo motors.

5.3 Joint Characteristics

5.3.1 Ankles

Each ankle has one degree of freedom (roll). The roll degree of freedom allows the robot to move its feet side-to-side. Although ankle roll is not necessary in 3D walking if hip roll joint is present, but its availability allows the robot's feet to stay flat on the ground during almost throughout the entire single support phase. Ankle roll can too contribute to the control of the biped such that it will not fall sideways while walking.

5.3.2 Knees

Each knee has one degree of freedom (pitch), which is made of one bracket, one servo holder and one servo. Just like the case in the humans, the knee is limited by a stop that does not allow the shin to bend out where out is defined the direction in which the swing shin is rotating up.
Therefore a knee stop is used in the simulation model in order for us to be able to lock the knees as soon as the leg is straightened during landing and support phases.

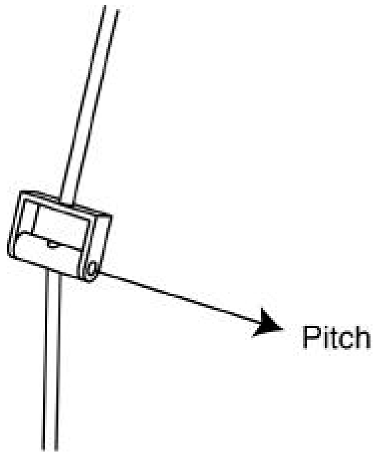


Figure 11 : Knee joint

5.3.3 Hips

Each hip has one degree of freedom (pitch). The pitch degree of freedom allows the robot to swing its leg forward and backward.

5.4 Summery

The robot model has six links and six degrees of freedom, which allows the biped to traverse in the 3D world. There are one degree of freedom on each hip, one degree of freedom on each knee, and one degree of freedom on each ankle. This biped is meant to have all necessary degrees of freedom in order to walk as naturally as possible without being held by an external object.

5.5 Robot's Natural Dynamics

Many researchers have exploited natural dynamics to make their walking machines walk passively meaning that their machines rely completely on their natural dynamics and gravitational force in order to traverse along. Powerful design of a robot can simplify the control significantly by making use of natural dynamics. For instance, spinning an object about its small and large axis is naturally stable and requires no complicated control system. Pratt et al have
employed natural dynamics in order to make a powered planar bipedal robot walk. They have also shown that natural dynamics can simplify control of a powered planar biped significantly.

5.6 Robot Control

5.6.1 Algorithm for Walking Initiation

A Finite State Machine (FSM), comprising two states, is used for walking initiation control algorithm as shown in Figure 4 -1. In the first state (Leaning Sideways), the robot uses one of its ankle rolls joints (in this case, the right one) to push against the ground (by twisting the right foot) and as a result of that, the robot leans to the opposite side. During the whole time that the robot is in state 1, all its joints are controlled using proportional derivative controller. The knees are in the locked position the whole time. The robot keeps pushing against the ground in the frontal plane until the position of its center of mass, which is measured from the left ankle falls on top of its left foot. This is when the biped goes into state 2 (Pick up Foot).

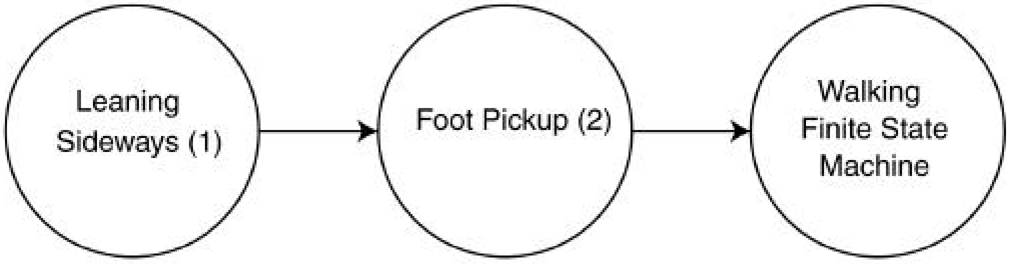
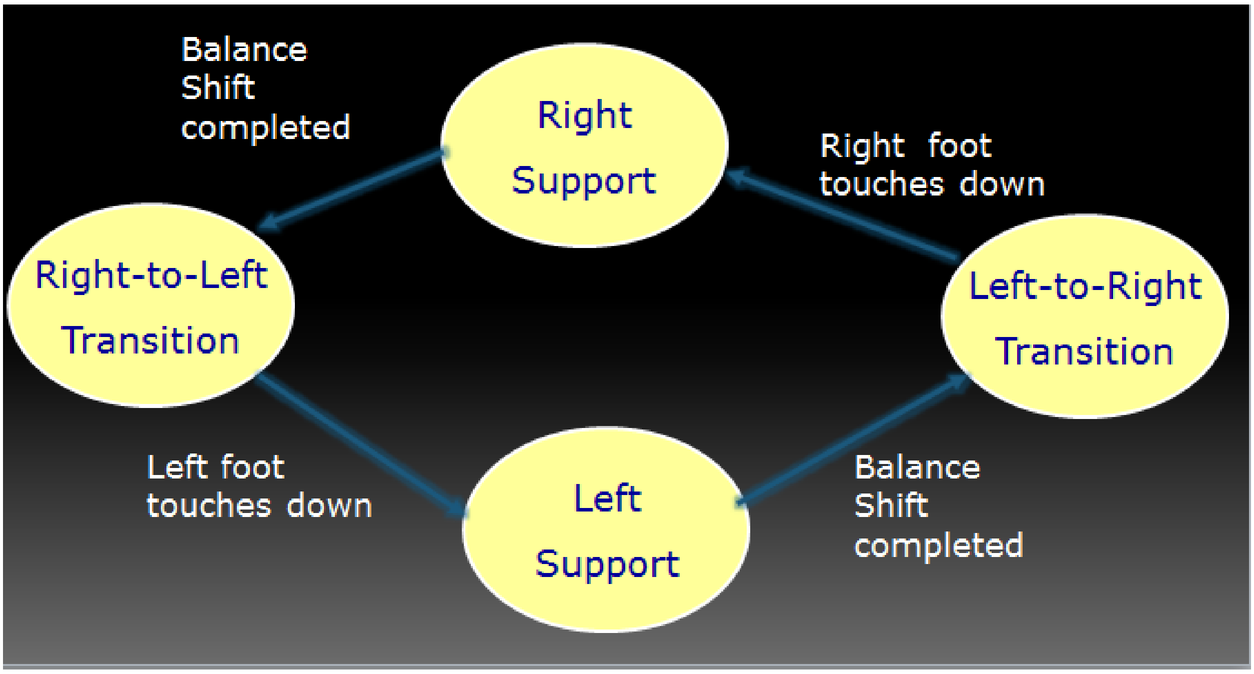


Figure 12 : Finite state machine is used for walking initiation.

In state 2 most of the robot's weight has been taken off of its right foot, which makes it plausible for the robot to pick it up by driving its right hip pitch joint to a desired position. The knee joint of the right leg is bent at the same time while the left knee maintains its locked position. The right foot is controlled to stay parallel with the ground to ensure foot clearance. Left ankle roll is used to control the position of the center of mass of the robot in the frontal plane so that it won't fall to the side. As soon as the position of the right hip pitch joint reaches a certain threshold, the robot goes into a different state, which is when it starts walking. The whole algorithm can be shown as such :



The walking initiation process can be best shown in the following figure:

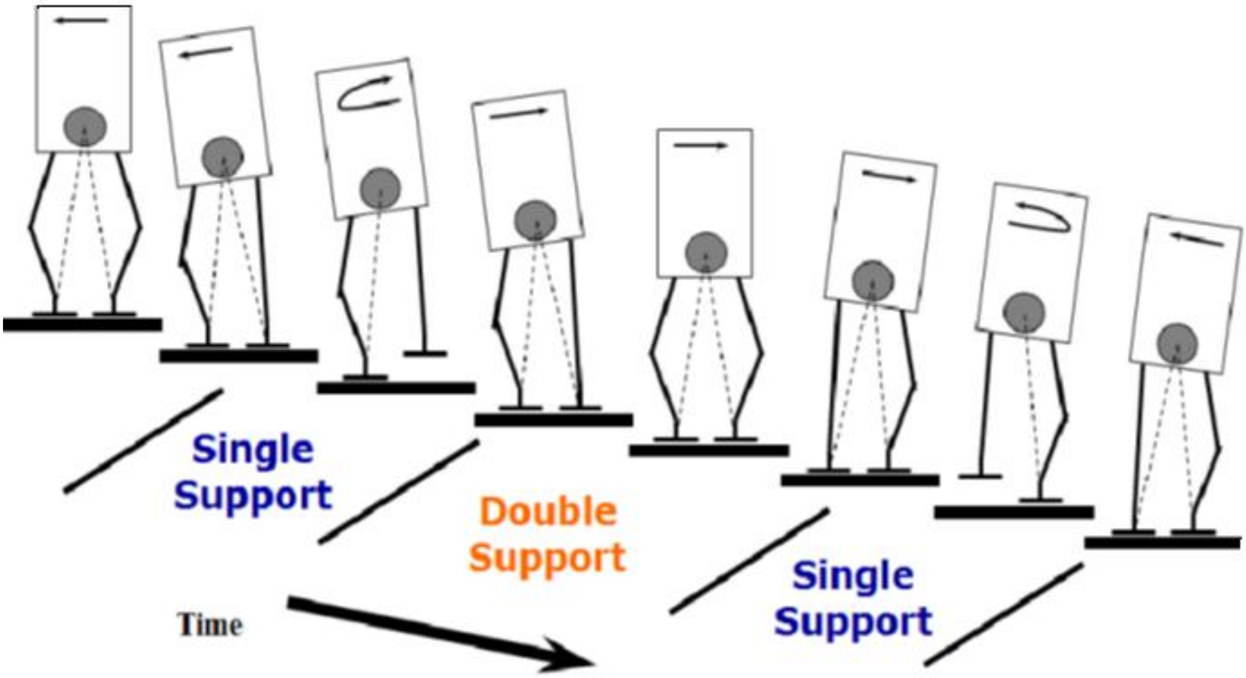


Figure 13: Walking Initiation Process

5.7 Future Developments

5.7.1 Model Structure

The current model can be modified and improved by using seven links and twelve degrees of freedom. That is three degrees of freedom at each hip (roll, pitch, and yaw), one degree of freedom at each knee (pitch), and two degrees of freedom at each ankle (roll, pitch). This three-dimensional seven-link biped will possess all the degrees of freedoms required in order to freely traverse in the three-dimensional world, including turning.

Among the extended degrees of freedom, the pitch degree of freedom of the ankles will allow the robot to move its feet up and down. The roll degree of freedom of the hip joint will provide the side-to-side motion of the leg which the robot needs in order to place its foot where it can prevent itself from falling sideways, and the yaw degree of freedom is the twist which is required for the robot to be able to turn. The revised model will have thirteen servo motors, twelve on link joints and one for mass transfer mechanism on the waist. The improved structure may look like the model in the following figure.

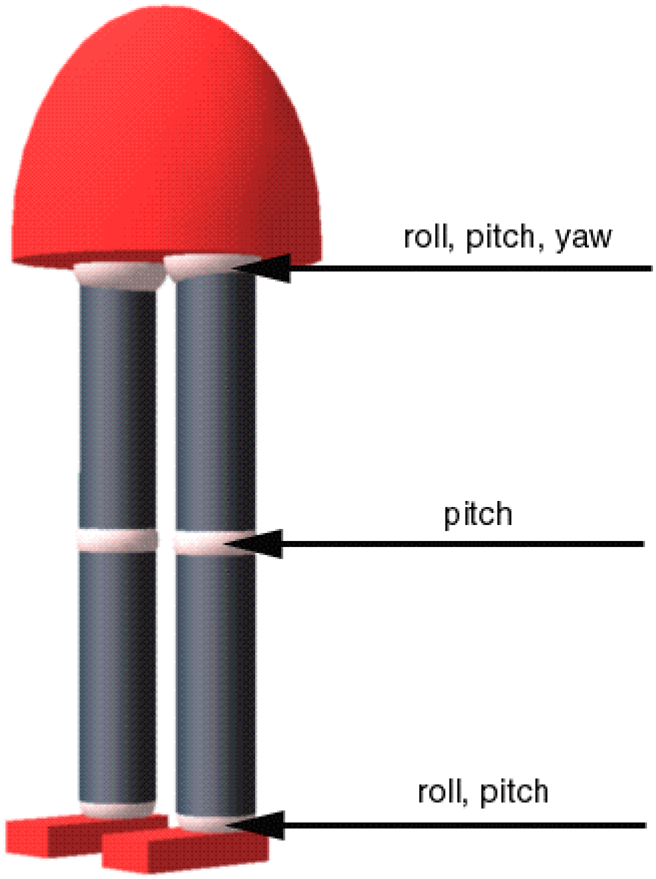


Figure 14 : The three-dimensional bipedal robot with three degrees of freedom at each hip, one degree of freedom at each knee, and two degrees of freedom at each ankle.

5.7.2 Link Structure

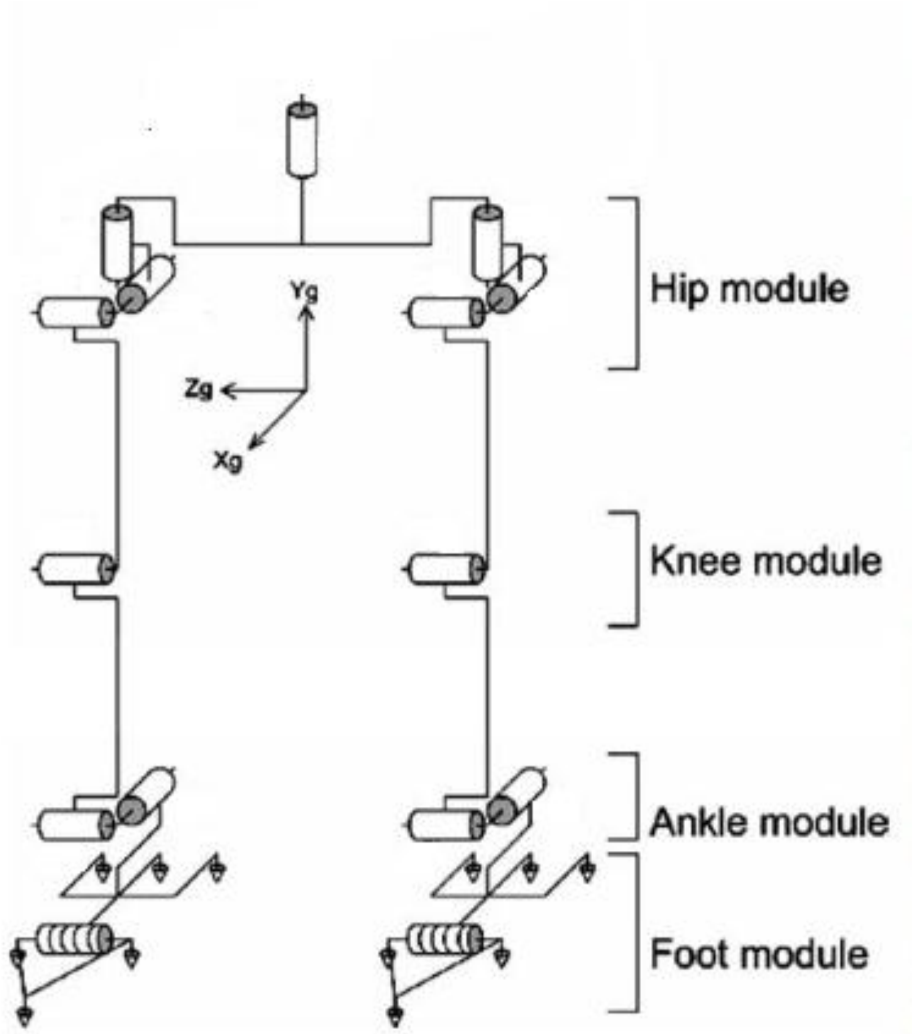


Figure 15 :Link Structure

Chapter 6

Design

For this project we have to design two components. One is the servo bracket and another one is the servo holder. Brackets are used for holding the servo gear and servo holders are used for holding purposes.

Dimensions and Mechanical drawings are provided below:

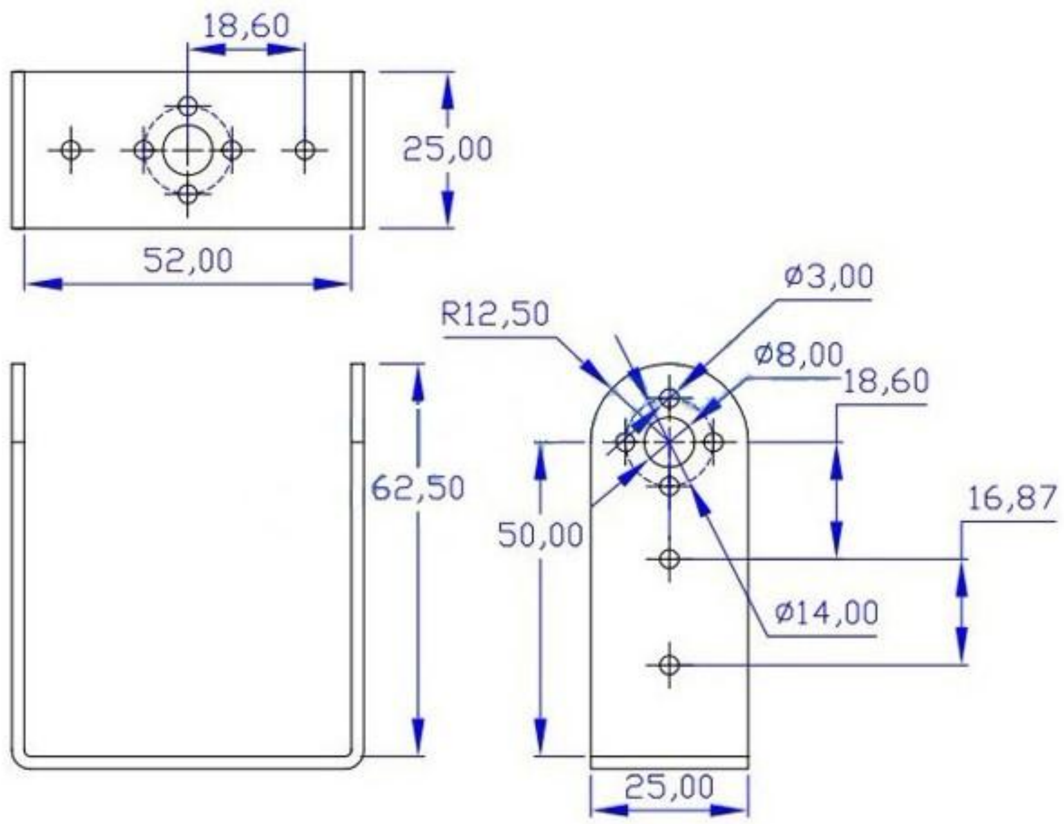


Figure16 : Servo Bracket

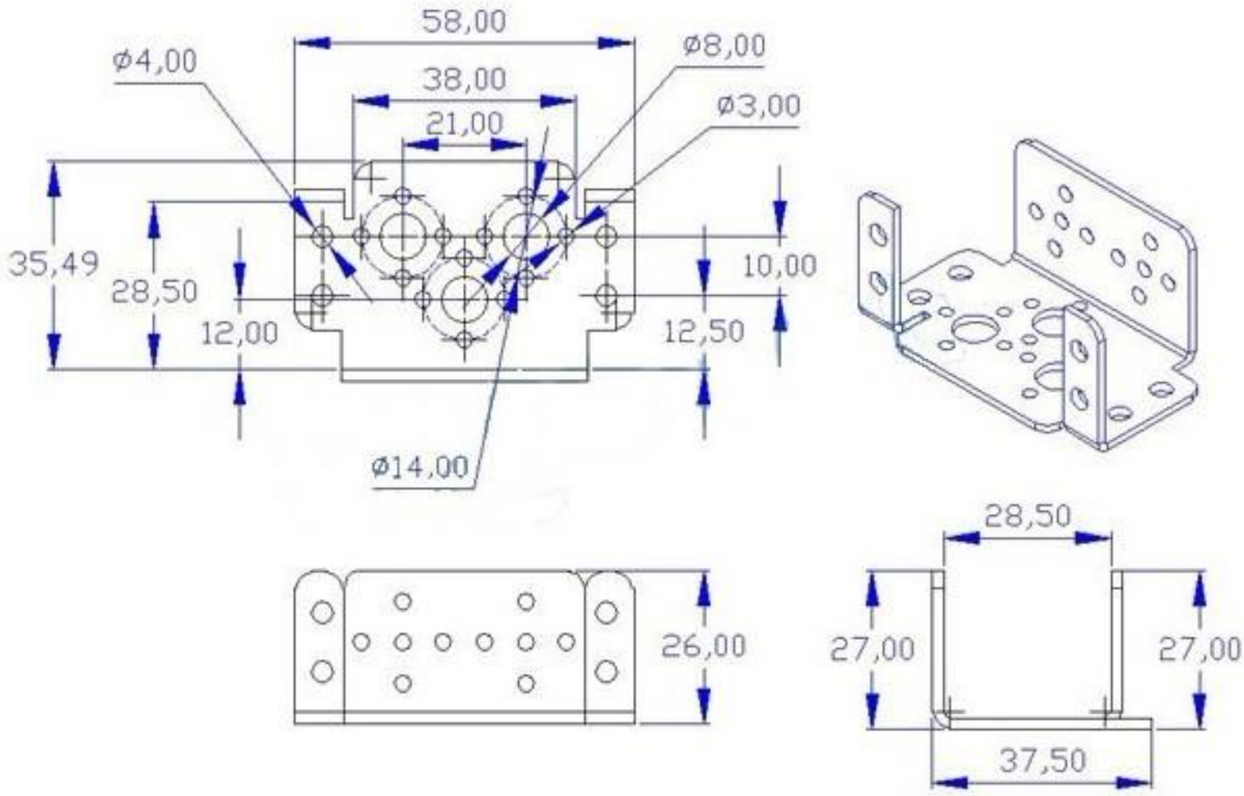


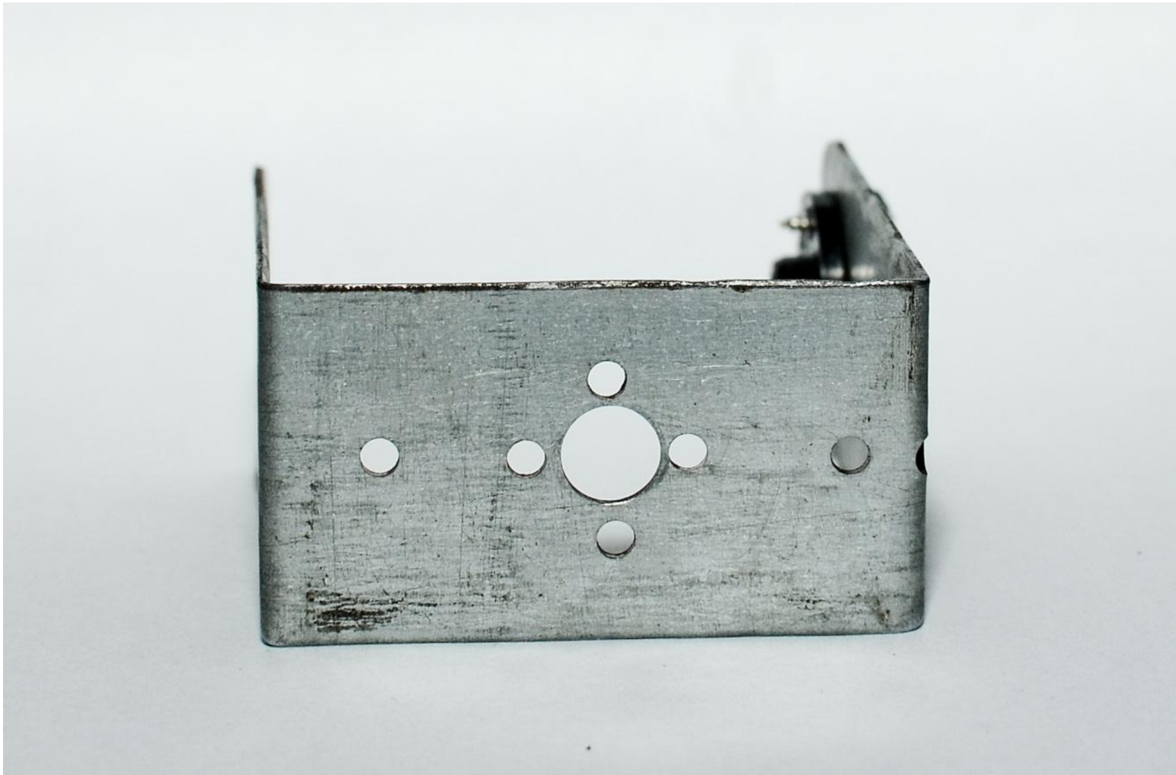
Figure 17: Servo Holder

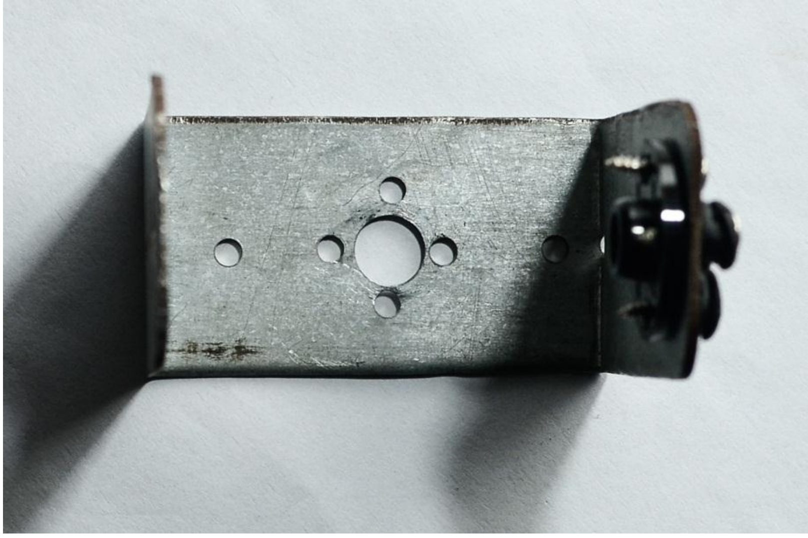
6.1 Construction

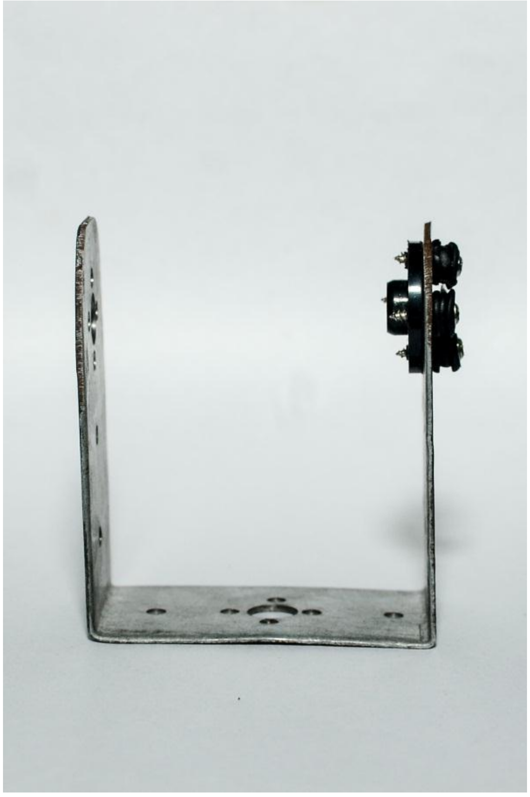
6.1.1 Servo Bracket

For constructing the bracket we have used mild steel sheet having a thickness of 1.5mm. Then we have made a layout on the fabrication lab. CNC drilling has been used to do the drilling job. After that bending is done using the bending machine. We have used the grinder in order to get the round shape in the upper part and to fine the edge.

Some figures are shown in different view-point:



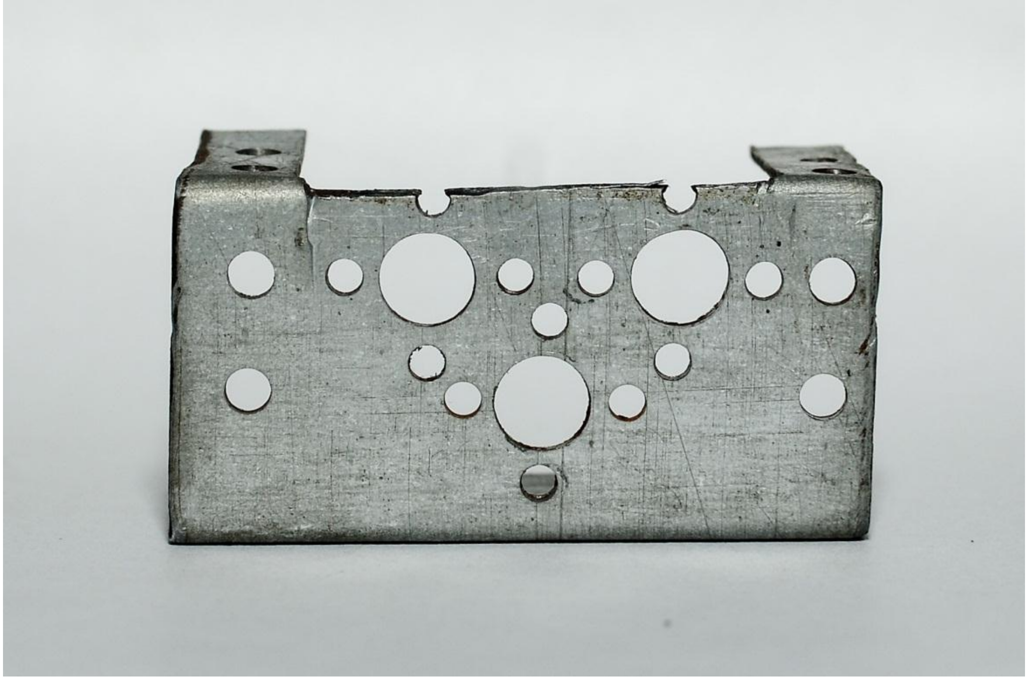




6.1.2 Servo Holder

For constructing the servo holder we have used mild steel sheet having a thickness of 1.5mm.Then we have made a layout on the fabrication lab. The drilling job is done on the CNC drilling machine as like as the bracket. And then necessary bending is done by using the bending machine. Grinding machine has been used to finish the job

Some figures of the holder are shown from different view-point:



6.1.3 Shoes

At the bottom we have attached Shoes. The dimension of each shoe is 130mm x 90mm x 1.5mm. The material of the shoe is mild steel.

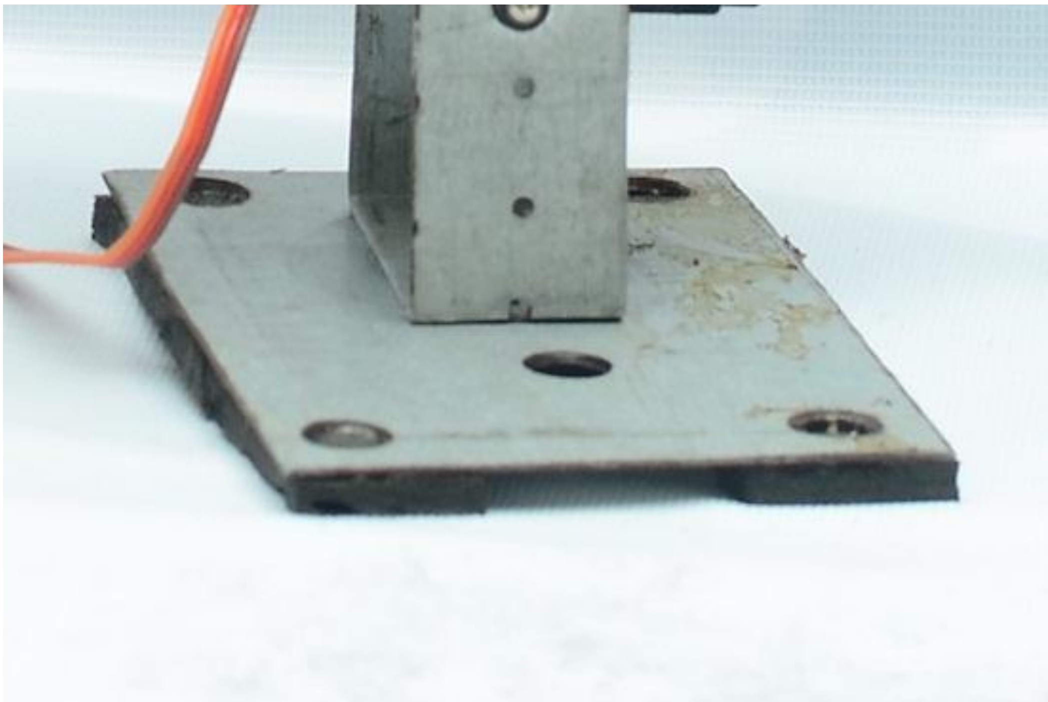
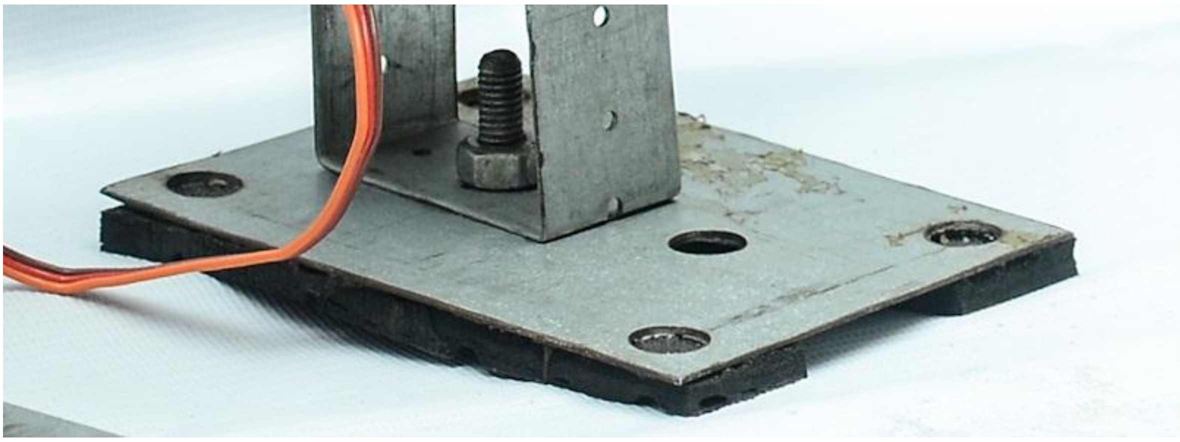


Figure 18: Shoe



6.2 Servo Motors

To drive the joints we have used servo motors.

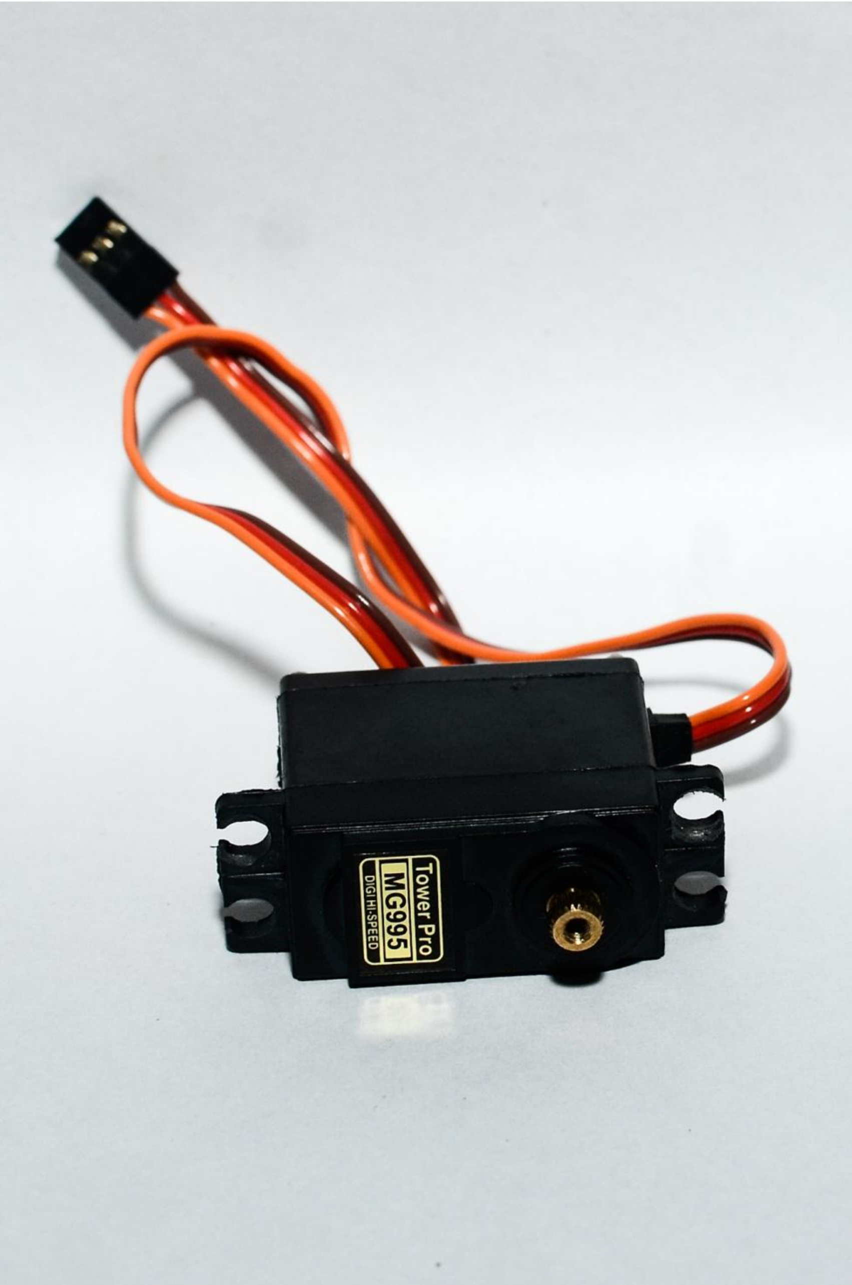


Figure 19: Servo Motor

6.2.1 The specifications:

Model type : Tower Pro MG995.

a) Modulation : Analog

b) Torque : **4.8V :** 138.9 oz-in (10.00 kg-cm)

c) Speed : **4.8V :** 0.20 sec/60°

d) Weight : 1.94 oz (55.0 g)

e) Dimensions : Length : 1.60 in (40.6 mm

Width : 0.78 in (19.8 mm)

Height : 1.69 in (42.9 mm)

f) Motor Type : Coreless

g) Gear Type : Metal

h) Rotation / Support : Dual Bearings

6.3 Motor Positioning

There are 3 points where we used motors for the movement of the legs.

1. **Hip Joint**
2. **Knee Joint**
3. **Ankle Joint**

1. **Hip Joint**

- **2 servo motors at 2 hip joints.**
- **For upward and downward movements.**

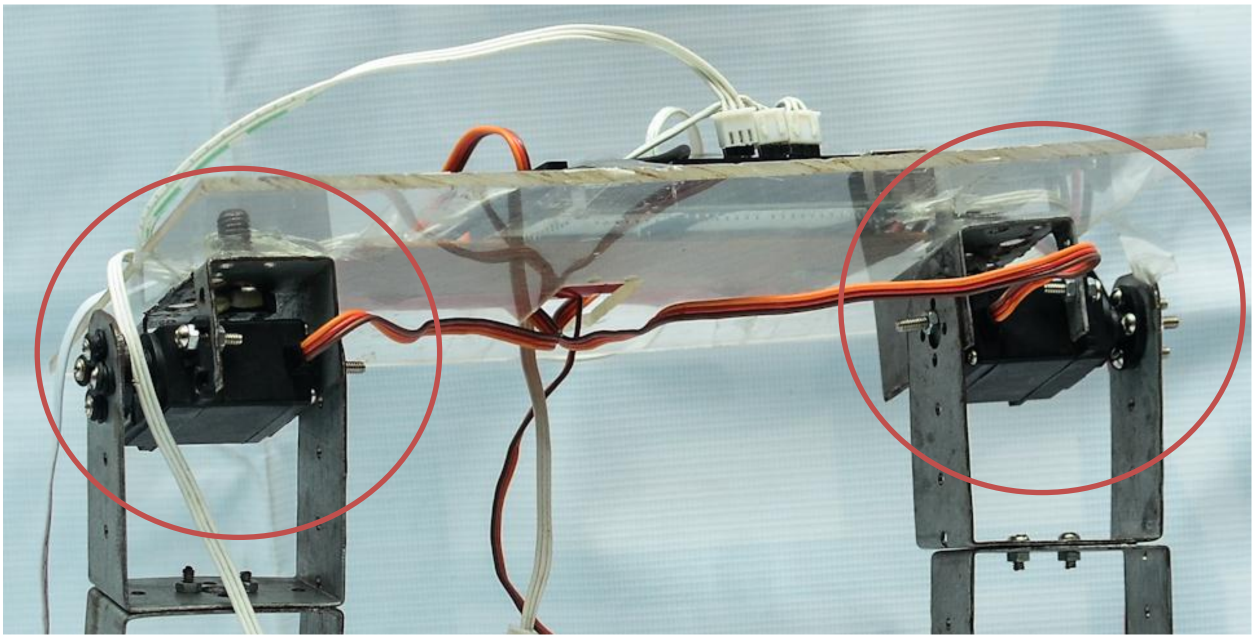


Figure 20: Hip Joint

2. Knee Joint

- 2 servo motors at 2 knee joints.
- For Upward and downward movements.

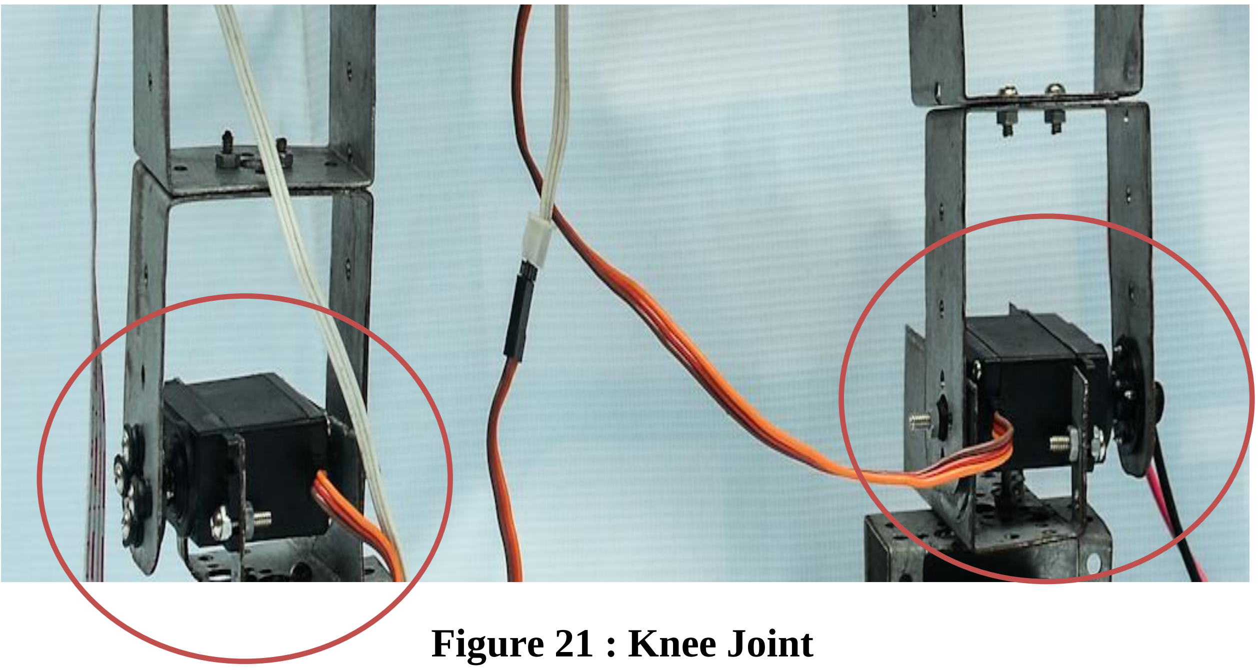
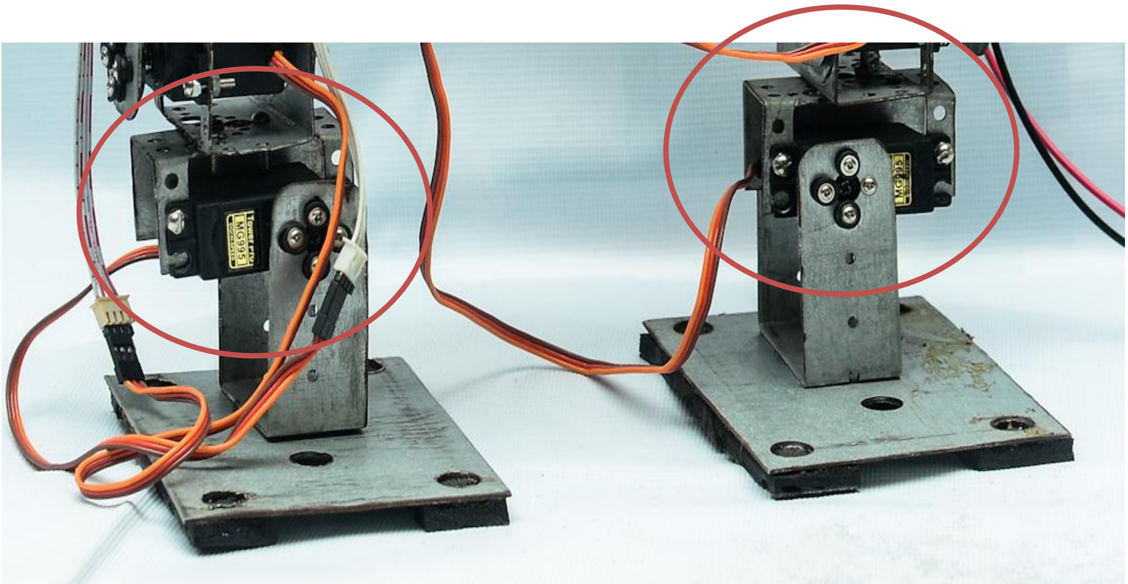


Figure 21 : Knee Joint

3. Ankle Joint

- 2 servo motors at 2 ankle joints.
- For sideway movements.



6.4 Complete Structure

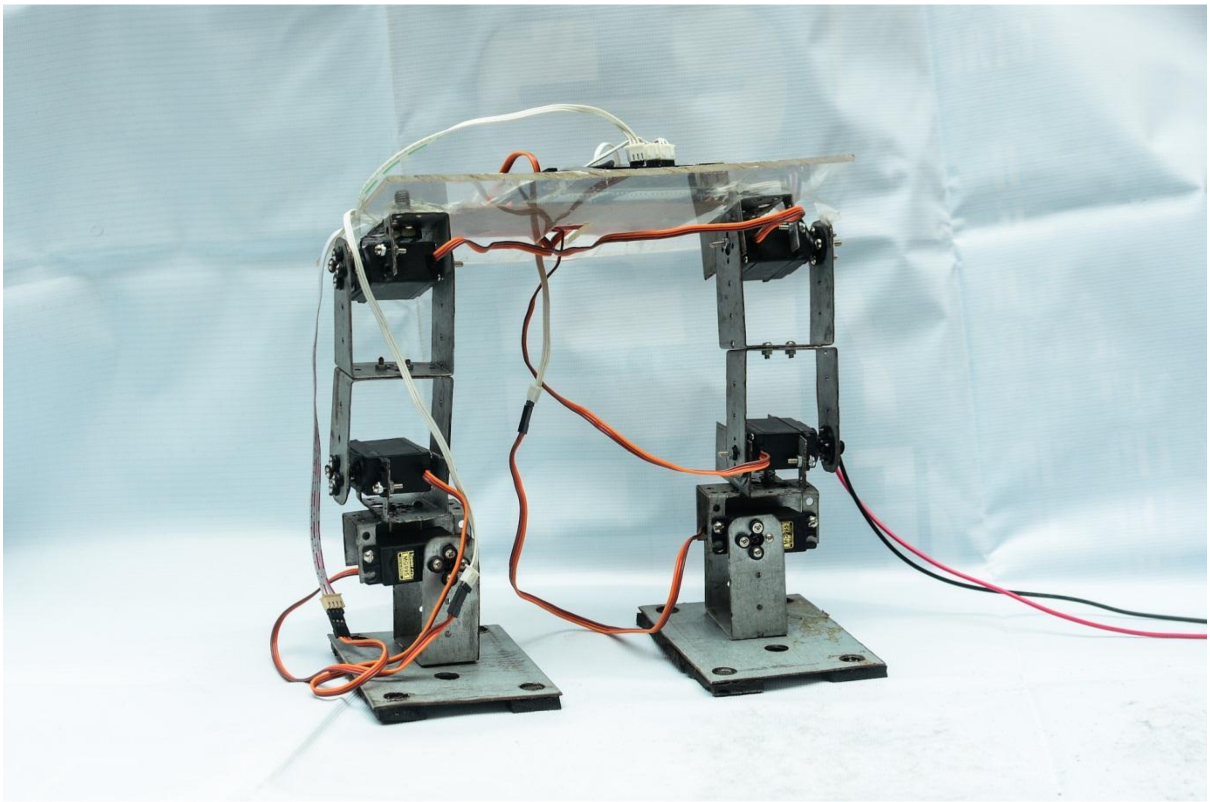


Figure 22: Front view

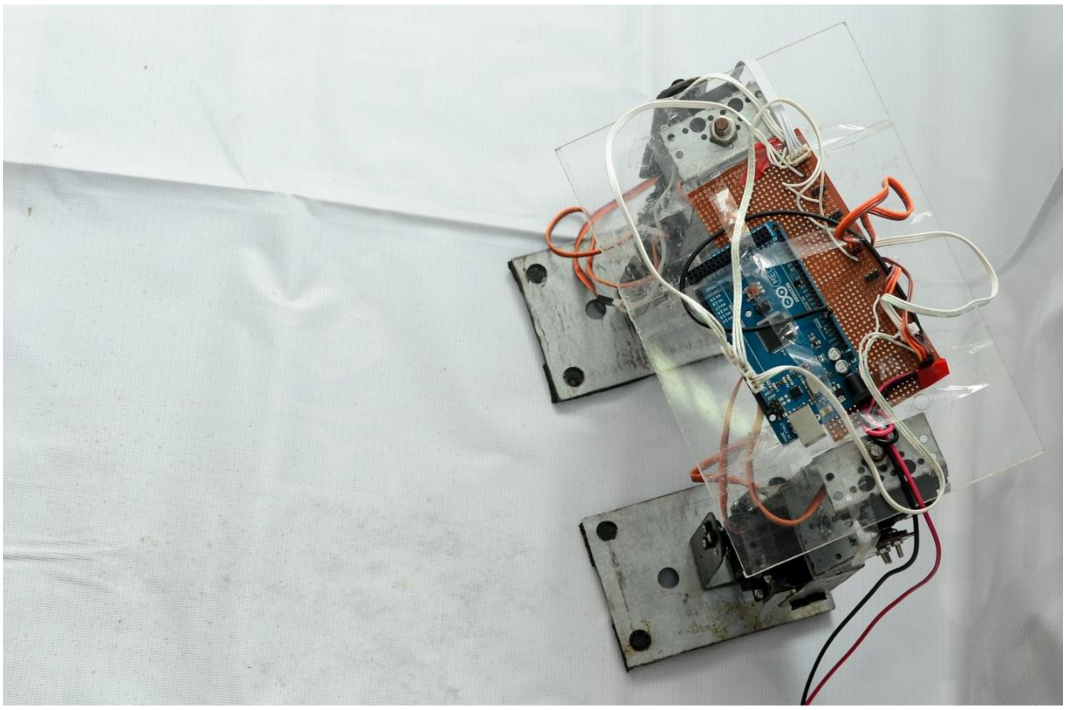


Figure 23: Top View

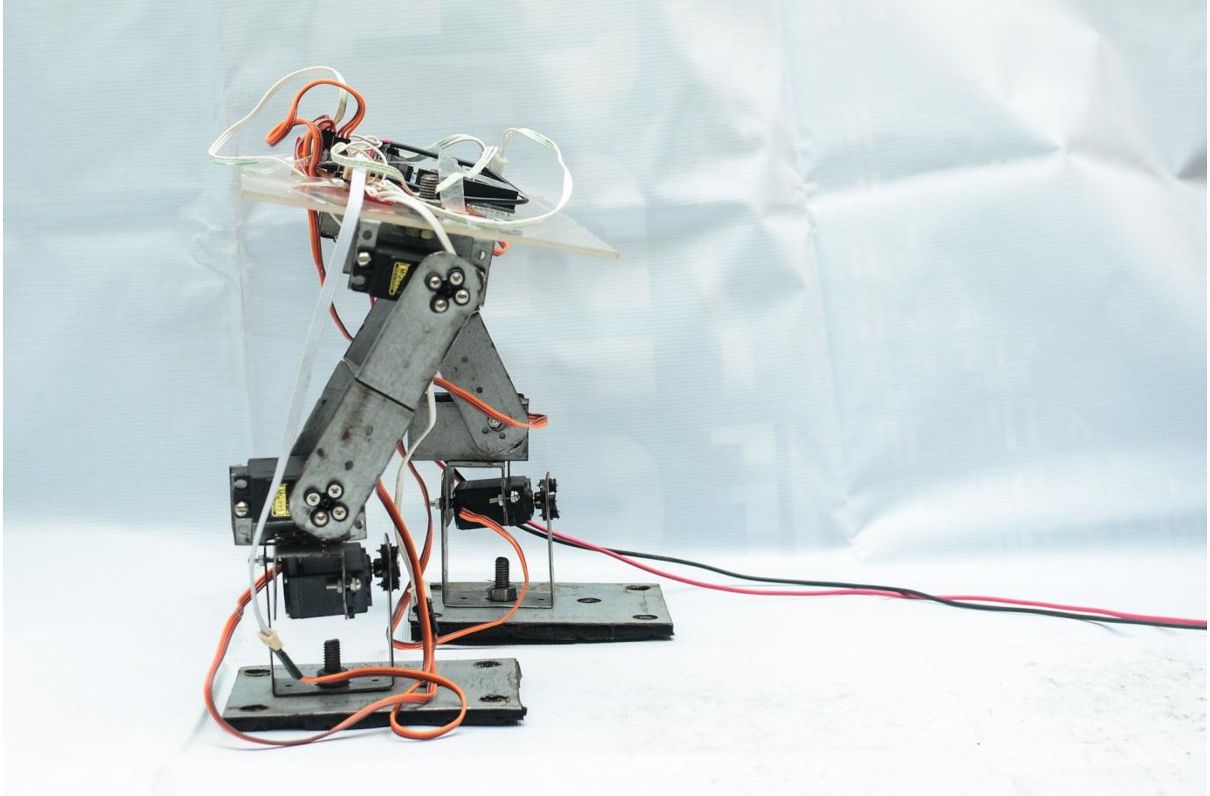
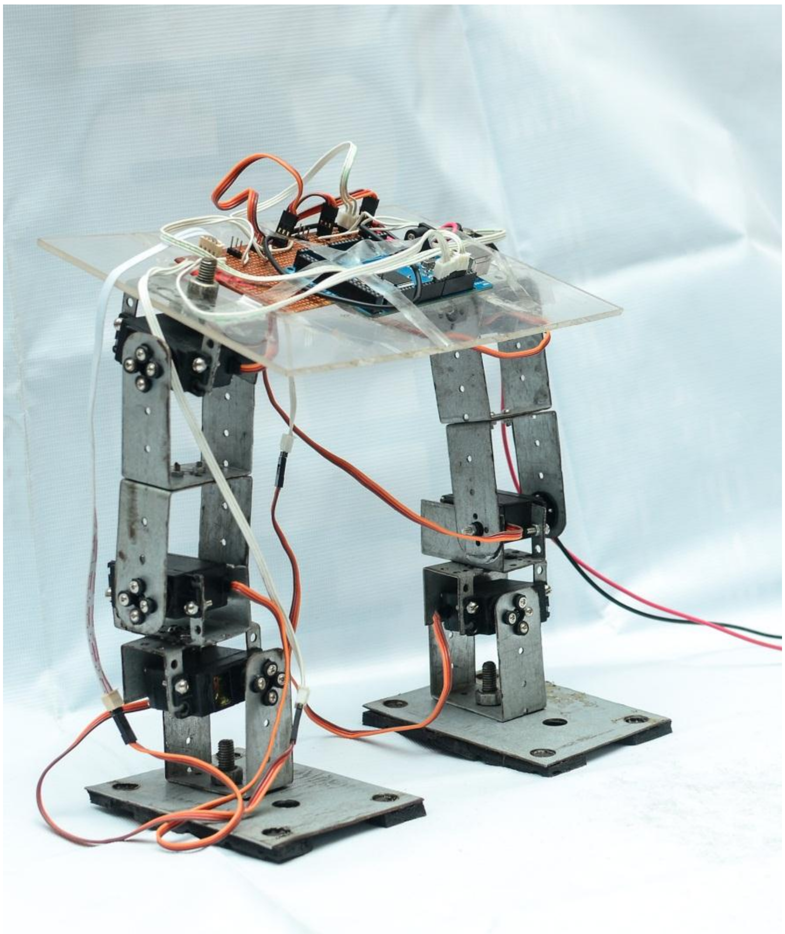


Figure 24: Side View



Chapter 7

Microcontroller

In order to control the servo motors Arduino Mega 2560 microcontroller unit is being used. It is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

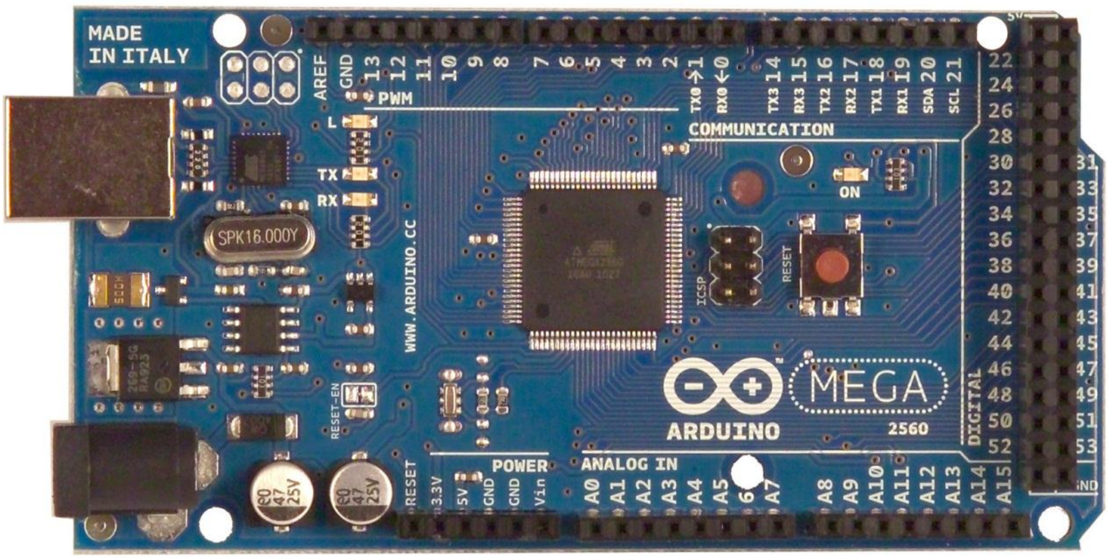


Figure 25 : Arduino Mega 2560

7.1 Specifications:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 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

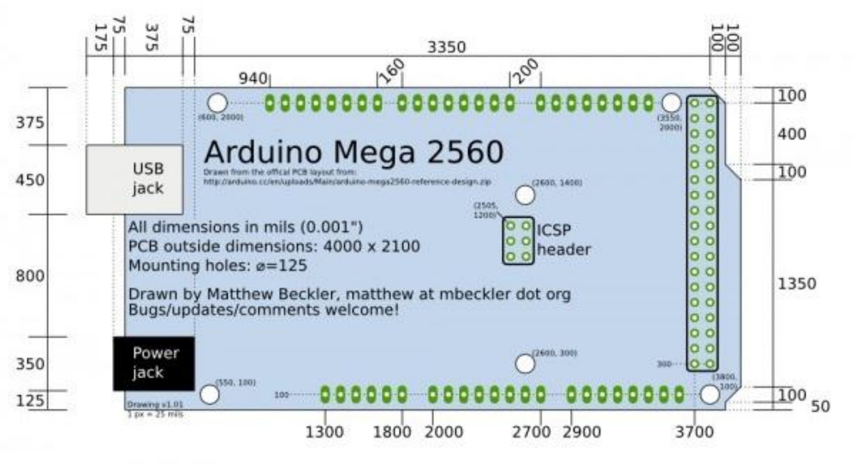


Figure 26: Schematic Diagram of the Microcontroller

7.2 Power Supply

7.2.1 Servo Motors

For supplying power to the servo motors we have used Wintrade ATX-500W Power Supply. Which gives constant 5Volts to each motor.

7.2.2 Microcontroller

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

In our project we used the USB connection to supply power to the microcontroller.

7.3 Circuit Diagram

For the connection purpose the following circuit diagram has been used.

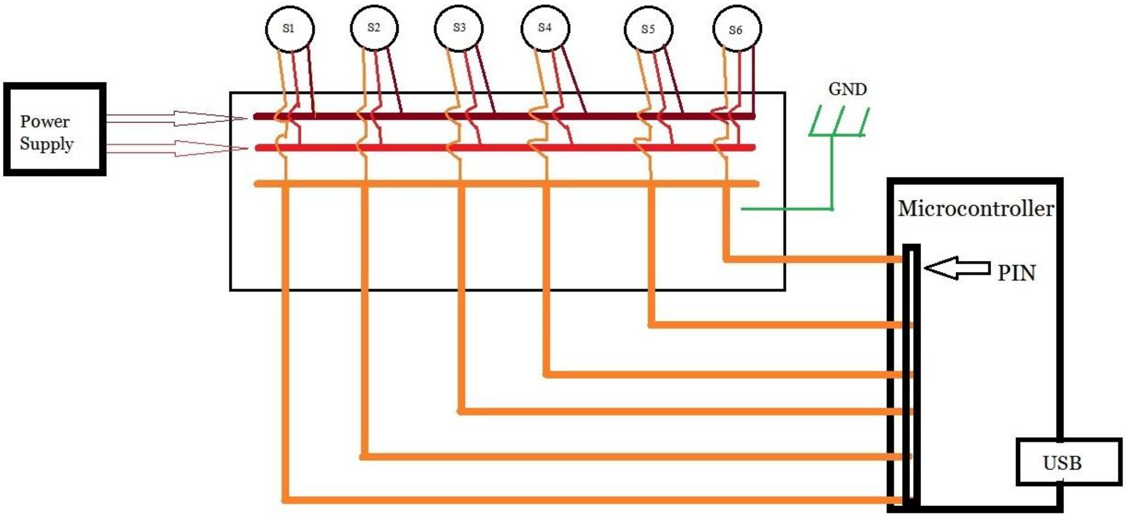


Figure 27: Circuit Diagram

Chapter 8

Programming

The programming is done by using C++ based on JAVA platform

8.1 Source Code (for Arduino)::

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////
#include <Servo.h>


Servo servoR1;

Servo servoR2;

Servo servoR3;


Servo servoL1;

Servo servoL2;

Servo servoL3;


int pos = 90;

int i,a,b,cR1,cR2,cR3,cL1,cL2,cL3;

void setup()

{

  servoR1.attach(9);

  servoR2.attach(6);
```

```
  servoR3.attach(5);


  servoL1.attach(13);
  servoL2.attach(11);
  servoL3.attach(12);


}
void stand(){
   servoR1.write(pos);
   servoR2.write(pos);
   servoR3.write(pos);
   servoL1.write(pos);
   servoL2.write(pos);
   servoL3.write(pos);


   Serial.println("O K ");
   delay(3000);
}
void loop()
{
   delay(1000);
   stand();
```

```
//for incl
  for(i=0;i<40;i++){
    servoL1.write(pos+i);
    //servoR1.write(pos+i);
    delay(30);
  }
   //R3 wt on left leg
   delay(1000);
  for(i=0;i<40;i++){
    servoR3.write(pos-i);
    servoL3.write(pos+i);
    delay(20);
  }
   delay(5000);
  servoR3.write(pos);

    for(i=0;i<25;i++){
    servoR1.write(pos+i);
    delay(20);
    }
    for(i=0;i<15;i++){
    servoR2.write(pos+i);
    delay(200); }
```

```
    for(i=0;i<10;i++){
    servoL3.write(pos-i);
    delay(500);
    }
  for(i=0;i<30;i++){
    servoL1.write(pos-i);
    servoR1.write(pos+i);
  delay(20);
  }
  for(i=0;i<20;i++){
    servoR2.write(pos-i);
delay(20);
}
delay(1000);
for(i=0;i<40;i++){
    servoR3.write(pos-i);
    servoL3.write(pos+i);
    delay(20);
  }
   delay(20000)
 }
```

//

8.2 Source Code (C):

```
//shift body weight to left leg
	send_cmd(L1, mid_L1+300, 30);		//move L1 to mid_L1+300 position with speed of 30
	send_cmd(R1, mid_R1-300, 30);		//move R1 to mid_R1-300 position with speed of 30
	send_cmd(L2, mid_L2+300, 30);		//move L2 to mid_L2+300 position with speed of 30
	send_cmd(R2, mid_R2-300, 30);		//move R2 to mid_R2-300 position with speed of 30
	delay(200000);				//delay for certain period
	send_cmd(L3, mid_L3-200, 20);		//move L3 to mid_L3-200 position with speed of 20
	send_cmd(R1, mid_R1, 30);		//move R1 to mid_R1 position with speed of 30
	send_cmd(R2, mid_R2, 30);		//move R2 to mid_R2 position with speed of 30
	send_cmd(R3, mid_R3-200, 20);		//move R3 to mid_R3-200 position with speed of 20
	delay(300000);				//delay for certain period


	while(1)
	{
		//right leg moved forward
		send_cmd(L3, mid_L3-60, 20);	//move L3 to mid_L3-60 position with speed of 20
```

send_cmd(R3, mid_R3-60, 20); //move R3 to mid_R3-60 position with speed of 20

delay(200000); //delay for certain period

send_cmd(L1, mid_L1-375, 40); //move L1 to mid_L1-375 position with speed of 40

send_cmd(L2, mid_L2, 20); //move L2 to mid_L2 position with speed of 20

send_cmd(R1, mid_R1+675, 40); //move R1 to mid_R1+675 position with speed of 40

send_cmd(R2, mid_R2+300, 15); //move R2 to mid_R2+300 position with speed of 15

send_cmd(L3, mid_L3+60, 10); //move L3 to mid_L3+60 position with speed of 10

send_cmd(R3, mid_R3+60, 10); //move R3 to mid_R3+60 position with speed of 10

delay(450000); //delay for certain period

send_cmd(L3, mid_L3, 20); //move L3 to mid_L3 position with speed of 20

send_cmd(R3, mid_R3+200, 20); //move R3 to mid_R3+200 position with speed of 20

delay(200000); //delay for certain period

send_cmd(R1, mid_R1-300, 60); //move R1 to mid_R1-300 position with speed of 60

send_cmd(R2, mid_R2-300, 30); //move R2 to mid_R2-300 position with speed of 30

send_cmd(L1, mid_L1, 30); //move L1 to mid_L1 position with speed of 30

send_cmd(L3, mid_L3+200, 30); //move L3 to mid_L3+200 position with speed of 30

delay(300000); //delay for certain period

//left leg moved forward

send_cmd(R3, mid_R3+60, 20); //move R3 to mid_R3+60 position with speed of 20

send_cmd(L3, mid_L3+60, 20); //move L3 to mid_L3+60 position with speed of 20

delay(200000); //delay for certain period

send_cmd(R1, mid_R1+375, 40); //move R1 to mid_R1+375 position with speed of 40

send_cmd(R2, mid_R2, 20); //move R2 to mid_R2 position with speed of 20

send_cmd(L1, mid_L1-675, 40); //move L1 to mid_L1-675 position with speed of 40

send_cmd(L2, mid_L2-300, 15); //move L2 to mid_L2-300 position with speed of 15

send_cmd(R3, mid_R3-60, 10); //move R3 to mid_R3-60 position with speed of 10

send_cmd(L3, mid_L3-60, 10); //move L3 to mid_L3-60 position with speed of 10

delay(450000); //delay for certain period

send_cmd(R3, mid_R3, 20); //move R3 to mid_R3 position with speed of 20

send_cmd(L3, mid_L3-200, 20); //move L3 to mid_L3-200 position with speed of 20

delay(200000); //delay for certain period

send_cmd(L1, mid_L1+300, 60); //move L1 to mid_L1+300 position with speed of 60

send_cmd(L2, mid_L2+300, 30); //move L2 to mid_L2+300 position with speed of 30

send_cmd(R1, mid_R1, 30); //move R1 to mid_R1 position with speed of 30

send_cmd(R3, mid_R3-200, 30); //move R3 to mid_R3-200 position with speed of 30

delay(300000); //delay for certain period

}

}

//

Chapter 9

Observation

We constructed a Biped humanoid robot according to the model design and tested it. The robot was successful in making one clean human like step. We are still working on the second phase of the project that is to successfully conduct the walking continuation of the biped structure. While testing the robot we found out that although it was possible to initiate the walking process with the six degrees of freedom structure, the continuation of humanoid walking is rather difficult in this design. This is due to fact that while initiating the walking both the legs were in same axis, so it was possible to shift the CG of the structure by tilting the robot on one side. But after the completion of the first step the legs stood on different axes. From this stance of the structure it is more difficult to shift the CG from one leg to another in order to make a humanoid step; only by the rolling movement of the structure. Thus the algorithm and the programming become much more complex and can only be determined by a process which is more or less like **trial and error**.

Chapter 10

Conclusion

After construction and testing of biped humanoid legs of six degrees of freedom structure we can conclude that although the walking initiation was successfully completed, to achieve walking continuation the design is not satisfactory. To make a humanoid robot that can initiate and continue walking and can change direction, the twelve degrees of freedom design, as suggested in the earlier part of this paper would be much more effective. The modified design can adjust the CG more efficiently as it has rolling motion on the hip joint and also a rotating mass on the waist above the hip joint. This setup makes the shifting of CG easier and the transition of weight from one leg to another smoother.

Also the use MS sheet metal made the structure relatively heavy. Using Aluminum instead of MS would reduce the weight significantly. Furthermore if bearings can be used on the joints the vibration would lessen and reduce the load on the servos. During the construction of the biped structure all the measurements, cutting, bending, grinding and on some occasions drilling were done manually on hand. This might have caused minimal errors in the dimensions of the structure which can be crucial in this experiment. Use of automatic CNC machines can remove these errors. These constructional improvements would make the movement of the biped leg more fluid and human-like.

LIST OF FIGURES

References

1. Control of a Simulated, Three-Dimensional Bipedal Robot to Initiate Walking, Continue Walking, Rock Side-to-Side, and Balance By Allen S. Parseghian

2. PR18 Walking with Cytron Servo Motor

3. Robot Leg Mechanisms By Amol Deshmukh

4. The Biped Robot Project by Ian Hume

5. Archer Prototype Testing

6. Archer Simple Leg Design

7. Minimax Differential Dynamic Programming:Application to A Biped Walking Robot By Jun Morimoto, Garth Zeglin and Christopher G. Atkeson

8. http://en.wikipedia.org/wiki/Bipedalism

9. C. Chevallerau and Y. Aoustin. Optimal running trajectoriesfor a biped. In 2nd International Conference on Climbing and Walking Robots, pages 559–570, 1999.

10. P. Dyer and S. R. McReynolds. The Computation and Theory of Optimal Control. Academic Press, New York, NY, 1970.

11. “Design of Advanced Leg Module for Humanoid Robotics Project of METI” Kenji KANEKO, Shuuji KAJITA, Fumio KANEHIRO, Kazuhito YOKOI, Kiyoshi FUJIWARA, Hirohisa HIRUKAWA, Toshikazu KAWASAKI, Masaru HIRATA, and Takakatsu ISOZUMI, Proceedings of 2002 IEEE / RSJ International Conference on Intelligent Robots and Systems Washington DC

12. “Mechanism and Control of a Leg-Wheel Hybrid Mobile Robot” H. Adachi and N. Koyachi, Mechanical Engineering Laboratory Namiki 1-2, Tsukuba, Ibaraki 305- 8564 Japan, T. Arai Osaka University Machikaneyama 1-3, Toyonaka Osaka, 560- 8531 Japan A. S k u and Y, Nogami Nisshinbo Industries, INC. Azukizaka 30, Miai, &chi 444-0802 Japan, Proceedings of the *1999* IEEE / RSJ International Conference on Intelligent Robots and Systems

13. “Design and control of a 3DOFs parallel actuated mechanism for biped application” R. Sellaouti, F.B. Ouezdou, Universite´ de Versailles St. Quentin—CNRS FRE2508, 10-12 Avenue de l’Europe, 78140 Ve´ lizy, France, Mechanism and Machine Theory 40 (2005) 1367–1393, Science Direct

14. “Development of a walking machine: mechanical design and control problems” Teresa Zielinska, John Heng, Institute of Aeronautics and Applied Mechanics, Warsaw University of Technology, ul.Nowowiejska 22, 00-665 Warsaw, Poland School of Mechanical and Production Engineering, Robotics Research Centre, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore, Received 1 October 1999; accepted 13 November 2000

===