“Shebl” Quadruped Robot System

A Graduation Project Report submitted to the Faculty of Engineering, Cairo University in partial Fulfillment of Bachelor of Science Degree in Computer Engineering

Presented by

Ayman Mohamed Abo El-Maaty
Abdelrahman Ahmed Ammar
Abdelrahman Nasr El-Shennawi
Abdelrahman Fawzi Ahmed

Supervised by

Assoc. Dr. Amr G. Wassal

Sponsored by

July 2013

Table of Contents

Acknowledgement 6

Abstract 7

الملخص 8

List of Tables 9

List of Figures 10

Contacts 11

Chapter1: Introduction 12

1.1 Problem Definition 12

1.2 Motivation and Justification 13

1.3 Domain of Applications 13

1.4 Summary of Approach 14

1.5 Document Overview 14

1.6 Table of Acronyms and Definitions 15

1.7 Keywords 15

Chapter2: Necessary Background 16

2.1 Terminology 16

2.2 Muybridge Picture Sequences 17

2.3 Classification of Gaits 18

2.4 Gaits and Stability 19

Chapter3: System Architecture 21

3.1 Hardware Modules 23

3.1.1 Robot Movement 23

3.1.1.1 Servo Control 23

3.1.1.2 Pulse Duration 23

3.1.1.3 Force 24

3.1.2 Robot State 25

3.1.3 Robot Planner 25

3.2 Software Modules 25

3.2.1 Trajectory Planning Module 25

3.2.2 Gait Planning Module 26

3.2.3 Servo Motors Manipulation Module 26

3.2.4 Gyroscope Sensors Module 27

Chapter4: Mechanical Design 28

4.1 Mechanical Structure 28

4.1.1 Number of legs 30

4.1.2 Leg Configuration 31

4.1.3 Ground Contact Point 32

4.2 Actuators 33

4.3 Actuators Details 34

4.4 Structure Specifications 35

Chapter5: Hardware Design 36

5.1 Arduino Mega 2560 36

5.1.1 Description 36

5.1.2 Specifications 37

5.1.3 Programming 38

5.2 Gyroscope sensor 39

5.2.1 Description 39

5.2.2 Specifications 40

5.2.3 Features 40

5.2.4 Applications 40

5.2.5 Pin definitions 41

Chapter6: Software Architecture 42

6.1 Gait Deliberative Planning Module 42

6.1.1 State Machine 42

6.1.2 Gait Selection Phase 43

6.1.3 Quad Shift Phase 44
6.1.4 Swing Phase 44
6.2 Trajectory Planning Module 45
6.3 Reactive Modules 46
6.3.1 Height Controller 46
6.3.2 Low Level Servo Controller 47
Chapter7: Experimental Results 48
7.1 Leg Trajectory Following 48
7.2 Power Consumption tests 49
7.3 Servo Motor Tests 49
7.4 Static Stand of the Robot 50
7.5 Blind Walking 50
7.6 Gyroscope Tests 50
7.7 Simple Walking 50
Chapter8: Conclusion 51
8.1 Main Conclusions 51
8.2 Challenges 52
8.2.1 Time Constraints 52
8.2.2 Market's Component 52
8.2.3 Hardware Challenges 53
8.2.3.1 Actuators 53
8.2.3.2 Gyroscopes 53
8.2.3.3 Arduino Board 54
8.2.4 Software Challenges 54
Chapter9: Future Work 55
9.1 Dynamic Support Stability Polygon 55
9.2 Leg Trajectory 55
9.3 Uneven Terrain Adaptation 55

9.4 The support ratio 56
9.5 Other Improvements 57
References 58
Appendix A: Feasibility Study 60
Technology & System Feasibility 60
Hardware Requirements 60
Software Requirements 60
Input and Output 61
Economic Feasibility 61
Cost 61
Benefit 62
Legal Feasibility 62
Operational Feasibility 62
Schedule Feasibility 63
Appendix B: Matlab Kinematic Codes 64
Direct Position Kinematics 64
Inverse Position Kinematics 64
Direct Velocity Kinematics 65

Acknowledgement

First of all we would like to thank our supervisor Assoc. Dr. Amr G. Wassal for the guidance, encouragement and inspiring us to make this project a step for the start of a robotics research group. We also appreciate the careful reading, editing of our presentations and reports, and for providing valuable comments.

We would like to thank also eng. Omar El-Seddek and Eng. Sherif Hany for their comments and their support.

We would like to thank Dr. Samir Shaheen for his help in getting the components we need.

Special thanks to Si-Ware Systems for their support, their constructive discussions and helpful comments.

Finally, we would like to thank all those who supported us and helped us completing this project.

Ayman, Abdelrahman, Abdelrahman, Abdelrahman

Abstract

Walking robots have a potential to traverse with greater mobility in certain types of terrain in a more efficient and stable manner than more conventional robots, using wheels or tracks. The property of walking robots that the contact with the ground is discontinuous gives them the ability to select footholds such that obstacles or holes are avoided. Other advantageous properties of walking robots are that they cause less damage to the terrain, active suspension is an intrinsic part of their structure, and they are omnidirectional, which gives them an advantage in maneuvering through cluttered and tight environments.

The control of walking robots requires that the issue of stability against tipping over is treated in a more specific fashion than for wheeled robots, as there are discrete changes in the support of the robot when the legs are lifted or placed. The stability of the robot is dependent on how the legs are positioned relative to the body and on the sequence and timing in which the legs are lifted and placed. In order to reduce the risk of the robot losing stability while walking, a measure for the stability of the robot is typically used in the gait and motion planning, in order to avoid, or detect, that the robot could become unstable.

The main purpose of the project is to build a simple quadruped robot “Shebl” with statically balanced gaits, and simple leg trajectory algorithm that ensure the stability of the robot motion. The motion of the robot is planned by determining the supporting force for each leg, which in turn will determine how the robot should shift its weight in order to remain statically balanced.

الملخص

الروبوتات المتحركه بالأرجل لديها القدره على الحركه بتنقليه أكبر فى أنواع طرق محدده بإسلوب أكثر كفائه و ثبات من الروبوتات المألوفه ذات العجل أو الجنازير و السيور. خصائص الروبوتات المتحركه بالأرجل التى تمس الأرض فى نقط متفرقه تعطيهم القدره على إختيار مواطئ الأقدام لكى تتفادى العوائق و الثقوب. من مزايا خصائص الروبوتات المتحركه بالأرجل الأخرى أن لها تدمير أقل للطريق, نظام التعليق الفعال هو جزء جوهرى من الهيكل, أنهم متعددوا الإتجاهات ذلك يعطى لهم أفضليه فى المناوره خلال البيئه المشوشه و الضيقه.

نظام التحكم فى الروبوتات المتحركه بالأرجل تحتاج إلى حل مشكله الثبات فى مواجهه الإنقلاب بطريقه أكثر تحديدا من الروبوتات ذات العجل حيث أن التغيرات فى دعامات الروبوت تتغير بطريقه منفصله عند رفع أو خفض أرجل الروبوت. ثبات الروبوت معتمد على طريقه وضع الأرجل بالنسبه إلى جسم الروبوت و على التسلسل و التوقيت حيث ترتفع و تنخفض عليه الأرجل. من أجل خفض خطوره فقدان توازن الروبوت عند الحركه قياسات لثبات الروبوت تستخدم عاده فى تخطيط المشيه و الحركه لكى نتفادى و نكتشف أن الروبوت أصبح غير متوازن.

الهدف الرئيسى للمشروع هو بناء روبوت بسيط ذا أربعه أرجل "شبل" بمشيه متوازنه بشكل ثابت و خوارزميه بسيطه لمسار الرجل التى تضمن توازن حركه الروبوت. تخطيط حركه الروبوت يتم عن طريق تحديد القوى الداعمه لكل رجل, التى بالتالى تحدد كيفيه زحزحه الروبوت لوزنه لكى يبقى متوازن بشكل ثابت.

List of Tables

Table 1: Table of Acronyms and Definitions 15
Table 2: List of Specifications and Dimensions of the Robot 35
Table 3: Specifications of Arduino Mega 2560 Board 37
Table 4: Gyroscope Pin Definitions 41
Table 5: Project Costs... 61

List of Figures

Figure 1: Picture sequence of galloping horse, photographed in the 19th century by Eadweard Muybridge 17
Figure 2: Support polygon, statically stable and unstable cases. The center of mass is the slightly larger circle, marked with 'X'. The smaller circles are the feet and are filled if they support the robot. 19
Figure 3: Gait diagram for (A) non singular, and (B) singular, quadruped crawl gait. 20
Figure 4: "Shebl" Quadruped Robot 21
Figure 5: System Architecture 22
Figure 6: PWM control signal 24
Figure 7: Trajectory Planning 25
Figure 8: Gait Planning 26
Figure 9: 3 Joints of the Robot, and the 4th passive joint that reacts the ground force on ground contact point 28
Figure 10: Backward Right Leg 29
Figure 11: Leg configurations of quadruped robots with forward walking direction to the right 31
Figure 12: Rubber foot contact at the end of the Leg 32
Figure 13: A Servo Motor broken down 33
Figure 14: HS-805BB Mega Power Servo Motor 34
Figure 15: HS-7980TH Servo Motor 35
Figure 16: Arduino mega 2560 board 36
Figure 17: Arduino software 38
Figure 18: Gyroscope Sensor L3G4200D 39
Figure 19: Gyroscope pin mapping and axes 41
Figure 20: A partial state machine of a single swing leg cycle 43
Figure 21: The Stable Quad Shift Algorithm 44
Figure 22: Desired trajectory for moving feet over obstacles 46
Figure 23: Actual Movement of the Leg 48
Figure 24: Simulated Movement of the Leg 48

Contacts

Name	Email Address	Mobile
Assoc. Prof. Amr G. Wassal	a.wassal@ieee.org	+201008715559
Ayman Mohamed Abo El-Maaty	eng_ayman_mohamed@hotmail.com	+201119865324
Abdelrahman Ahmed Ammar	abdelrahman.ammar@hotmail.com	+201225714226
Abdelrahman Fawzy Ahmed	abdo.fst.1991@gmail.com	+201227911517
Abdelrahman Nasr Elshennawi	abdulrahman.nasr@gmail.com	+201114485052

Chapter1: Introduction

Autonomous mobile robots are finding applications in human welfare areas, such as cleaning, running errands, and assisting handicapped or elderly. Mobile robots are also being developed to be used in areas that are inaccessible and/or dangerous for humans, for instance in demining, maintenance in hazardous environments, military, and exploration of volcanoes and space. For such robots to function autonomously, advanced sensing and algorithms are needed, for instance, to do localization, path planning, and decision making. Still, one important aspect of the development of autonomous mobile robots is the design and construction of walking robots that being developed to traverse certain types of terrain that can be classified as uneven or extreme terrain. Examples of such terrain would be found, for instance, in forests, mountains, or other rocky terrain, but also in indoor environments, where steps, stairways or high thresholds can cause conventional robots some difficulty.

1.1 Problem Definition

All the advantages of walking robots are dependent on the design of their mechanical structure and the control system. There are many challenges with designing and building legged robots. The large number of actuated degrees of freedom makes them heavier, more complex and expensive than wheeled systems. Most walking robots of today are quite slow and have bad payload weight-to-own-weight ratio compared to more conventional wheeled or tracked robots. The control of a walking robot has to cope with a highly nonlinear system with many degrees of freedom, changes in the system dynamics as the legs are being lifted and placed, and unknown dynamics such as the interaction of the foot with the ground. The control of walking robots also requires that the issue of stability against tipping over be treated in a more specific fashion than for wheeled robots, as there are discrete changes in the support of the robot when legs are lifted or placed.

1.2 Motivation and Justification

Despite the advantages of legs over wheels, almost none of today's robots actually showed this superiority. Most legged robots need at terrain to walk and run and have great difficulties to handle disturbances. The reason for this is mainly based on the highly non-linear dynamics of legged systems and the complexity to control many degrees of freedom in real-time.

An exception to all this, however, is the BigDog robot. During the last years, this quadruped robot regularly stunned the public with online videos that demonstrated its ability to perform highly dynamic tasks like running or balance recovery after slipping on ice. Besides these abilities, the robot's versatility allows it also to climb a pile of rubble or a steep slope. BigDog shows that today's technology is ready to construct such a versatile and highly dynamic robot; a fact that is a big source of motivation.

1.3 Domain of Applications

Possible applications of such a machine (other than research) principally include tasks in areas that are dangerous for human beings and not suitable for wheeled vehicles:

- Support for search and rescue operations in disaster areas after earthquakes, tsunami, landslides or avalanches.
- Transport of emergency supplies (such as food, first aid) to disaster areas that are difficult or impossible to reach by trucks or helicopters.
- Support for humanitarian demining of former war zones.
- Inspection tasks in dangerous areas.
- Various tasks in contaminated zones.
- Support for forestry related tasks such as cutting small bushes for prevention against spreading forest fires.

1.4 Summary of Approach

Our approach to design legged robots is to study the solutions that nature has provided to the locomotion of animals. The argument is that the way animals are built or move may in some sense represent an optimum, which may be beneficial in the design and control of legged robots. For instance the study of gaits has led to mathematical models that can explain why some gaits are preferred by animals. Today's technology is however far from being able to replicate complexity of the muscle-skeleton system and the neural system controlling it. On the other hand, machines can be built in ways that animals can't replicate, for instance, rotating actuators are not found in animals. Various innovative designs of legged robots have been tried. Furthermore, the development of very simplified mechanisms, in terms of number of degrees of freedom and actuators, has also proven that legged locomotion can be accomplished without trying to mimic the complexity of animals.

1.5 Document Overview

In this document we will talk about the technical details of our quadruped robot "Shebl". We will talk in Chapter 2 about the scientific background needed for understanding the rest of the document. While in Chapter 3, a brief overview of the System Structure and Architecture is demonstrated.

While in Chapters 4 and 5 we discuss the Hardware and Software design of "Shebl" robot. In Chapter 6 we see the results and conclusions we finished to through our work.

Finally in Chapter 7, we talk about the challenges that we faced and future work that can be added to improve the robot.

1.6 Table of Acronyms and Definitions

Term	Definition
COM	Center of Mass or Center of Gravity
DOF	Degree of Freedom
PWM	Pulse Width Modulation
ZMP	Zero Moment Point

Table 1: Table of Acronyms and Definitions

1.7 Keywords

Legged locomotion, Walking robots, Mobile robots, Quadruped robots, Stability measures, Center of Mass, Support triangle, Kinematics, Dynamics, Gaits, Foot trajectory.

Chapter2: Necessary Background

Research on legged locomotion has a long history. Biologists and other scientists have long studied the structure and motion of animals. In connection to the interest of the engineering community in building walking vehicles, there has been a search for more mathematical models for the study of gaits.

2.1 Terminology

Stride: The complete cycle of leg movements, for example, from the setting down of a particular foot to the next setting down of the same foot, where all the legs have been lifted and placed exactly once.

Stride length: The distance travelled by the center of mass of the body in one *stride*.

Stroke: The distance that foot translates relative to the hip during the *support phase*.

Duty factor: the fraction of the duration of the *stride* for which a foot is on the ground (in *support phase*).

Cycle time: Time duration of one *stride*, i.e. the time to complete one cycle of leg movements.

Events of the gait: The placing or lifting of any of the feet during locomotion.

Relative phase: The time elapsed from the setting down of a chosen reference foot until the foot of leg required is set down, given as the fraction of the *cycle time*.

Stability margin: The shortest distance from the vertical projection of the center of mass of the robot onto a horizontal plane, to the boundary of the support area.

Support area/polygon: The minimum convex polygon in a horizontal plane, with its vertices formed by the vertical projection of the feet being in support.

Support phase: The phase when a foot is in contact with the ground and able to support and propel the body. Also called *stance* or *retraction* phase.

Swing phase: The phase when a foot is in the air and repositioned for the next *support phase*. Also called *air* or *protraction* phase.

2.2 Muybridge Picture Sequences

In the 19th century, the English photographer Eadweard Muybridge achieved one of the first milestones in understanding animal running. He used several cameras to capture the single phases during running and proofed for the first time that all four of a horse's hooves left the ground at the same time during a gallop. Figure 1 shows a picture sequence of a galloping horse.

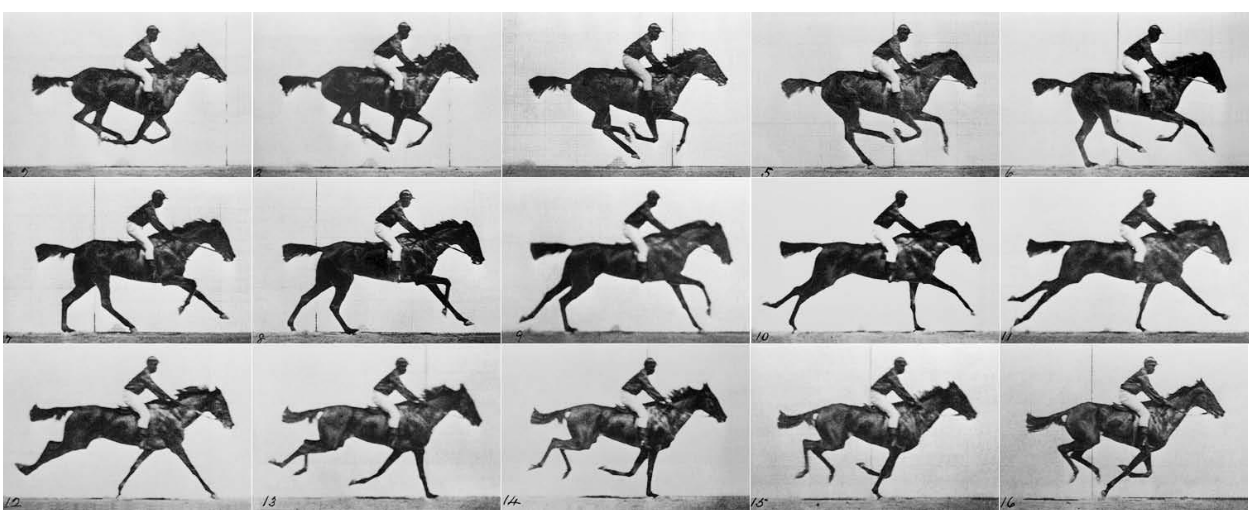

Figure 1: Picture sequence of galloping horse, photographed in the 19th century by Eadweard Muybridge

He took a big collection of picture sequences of fast motions of humans and animals, ranging from dogs and horses to elephants. His main interest lied in dynamic motions that are too fast for the human eye to analyze in detail, such as running, jumping, athletics and wrestling.

2.3 Classification of Gaits

Fundamental to the locomotion of animals is that they move by lifting their legs and placing them at new positions. While walking, the legs should be coordinated with respect to stability, propulsion and energy efficiency. The coordinated manner of lifting and placing the legs is called a gait. A gait is characterized by the sequence in which the legs are lifted and placed. The lifting or placing of a leg is called an event of the gait, and the sequence in which the legs are lifted and placed is called a gait event sequence. Most people are familiar with the names of some of these gaits, for instance, a horse will switch between different gaits when increasing speed, first walk, then trot, then canter, and finally gallop. Animals switch gaits depending on speed in order to be more energy efficient, and the speed at which animals switch gait is dependent on the size of the animal. It has been noted that animals of different species use similar gaits for certain types of motion.

A gait is usually cyclic in the sense that the same sequence of lifting and placing the legs is repeated. A complete cycle of leg movements, where all the legs have been lifted and placed exactly once, is called a stride. A quadruped gait has exactly eight events in one stride, as each of the four legs is lifted and placed once.

Gaits are divided into walking and running, where the main difference between a walking and running gait is seen in the duration of the support phase, i.e. in the size of the duty factor. The distinction, in general is that running gaits have a duty factor less than 0.5, and walking gaits have duty factor greater than 0.5. Consequently, running gaits have stages when both feet of a front or back leg pair are off the ground, whereas walking gaits have stages where both feet are on the ground simultaneously, creeping gaits for quadruped are defined as a subgroup of walking gaits, in which there are always at least three legs in ground contact at all times, requiring a duty factor greater than 0.75.

In addition, gaits are classified as alternating and non alternating gaits (or symmetric and asymmetric gaits). In alternating gaits, the left and right feet of a front and back leg pair have equal duty factors and relative phase differing by 0.5, i.e. by half a cycle. In non alternating gaits, the phase difference is 0, i.e. the front and back leg pairs move in unison.

2.4 Gaits and Stability

While walking or running, a legged locomotor has to remain balanced in order to avoid unwanted body motion or falling. Gaits are classified, depending on the strategy used in order to maintain balance, into statically and dynamically stable gaits. The strategy chosen is related to speed, as slower walking gaits, i.e. creeping gaits, are generally statically stable whereas faster gaits are dynamically stable. The main difference is that dynamically stable gaits remain balanced by moving whereas statically stable gaits remain balanced by relying on the support area formed by the legs in ground contact.

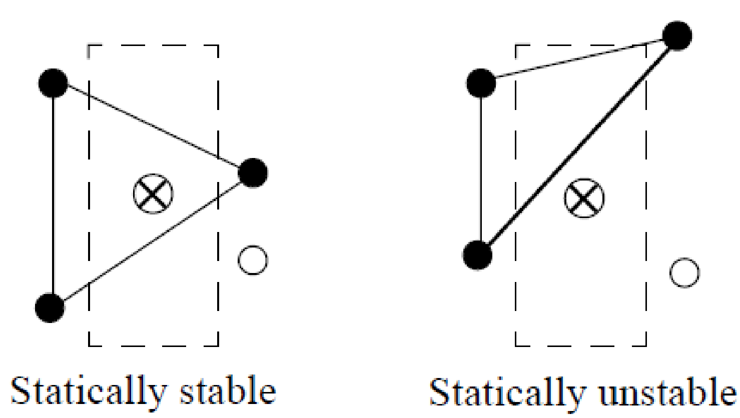

Figure 2: Support polygon, statically stable and unstable cases. The center of mass is the slightly larger circle, marked with 'X'. The smaller circles are the feet and are filled if they support the robot.

In a statically stable gait, the vertical projection of the center of mass (COM) onto a horizontal plane, is kept within the support area at all times, Dynamic stability is often referred to as active stability, and implies that balance is only achieved through motion, thereby in general, demanding more active strategies. If a walker is not statically stable it will start to fall.

Gait diagrams are used to show the development of the gait as a function of time. Each frame in the gait diagram is taken at an event of the gait, solid circles will denote foot in ground contact, open circles will denote a leg that has just been lifted, and dashed circles denote the position where a foot will be set down next.

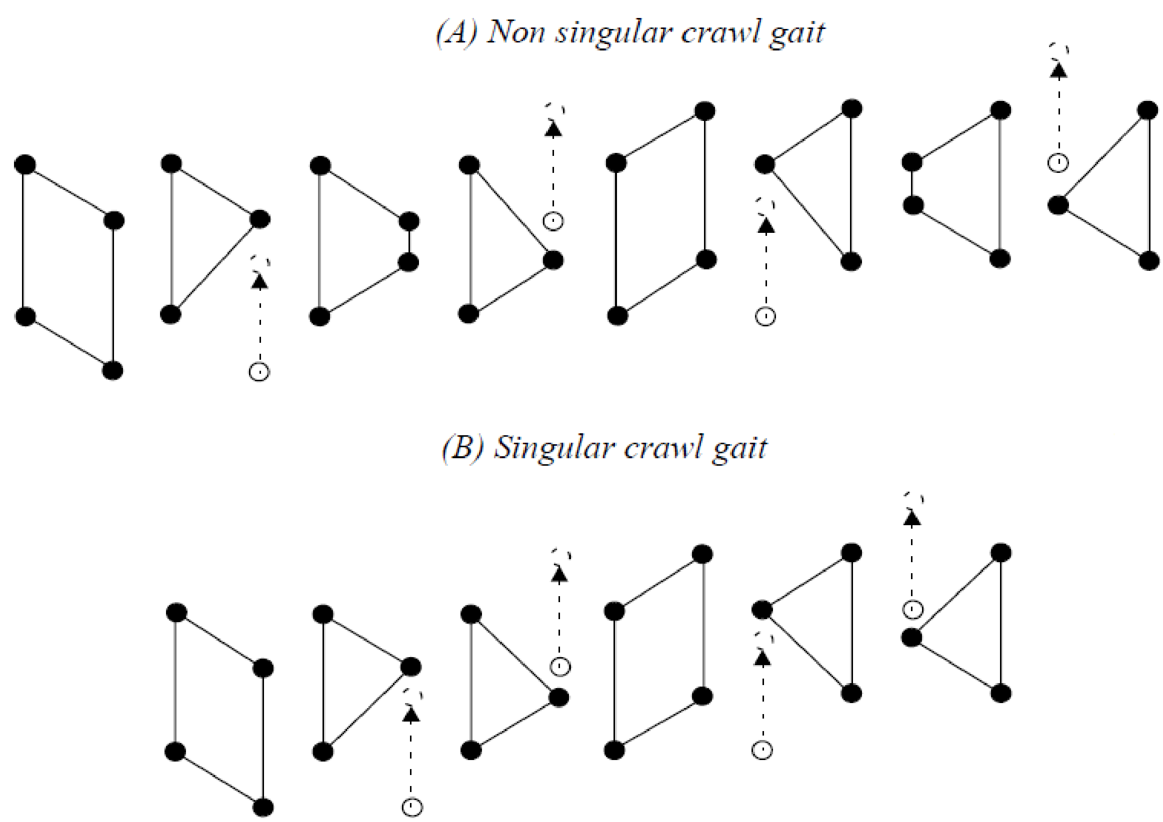

Figure 3: Gait diagram for (A) non singular, and (B) singular, quadruped crawl gait.

Chapter3: System Architecture

The system consists of some hardware and software modules each of which performs a specific function. These modules must interact in some way that enables the robot to move its legs in a specific order which achieve the required balance.

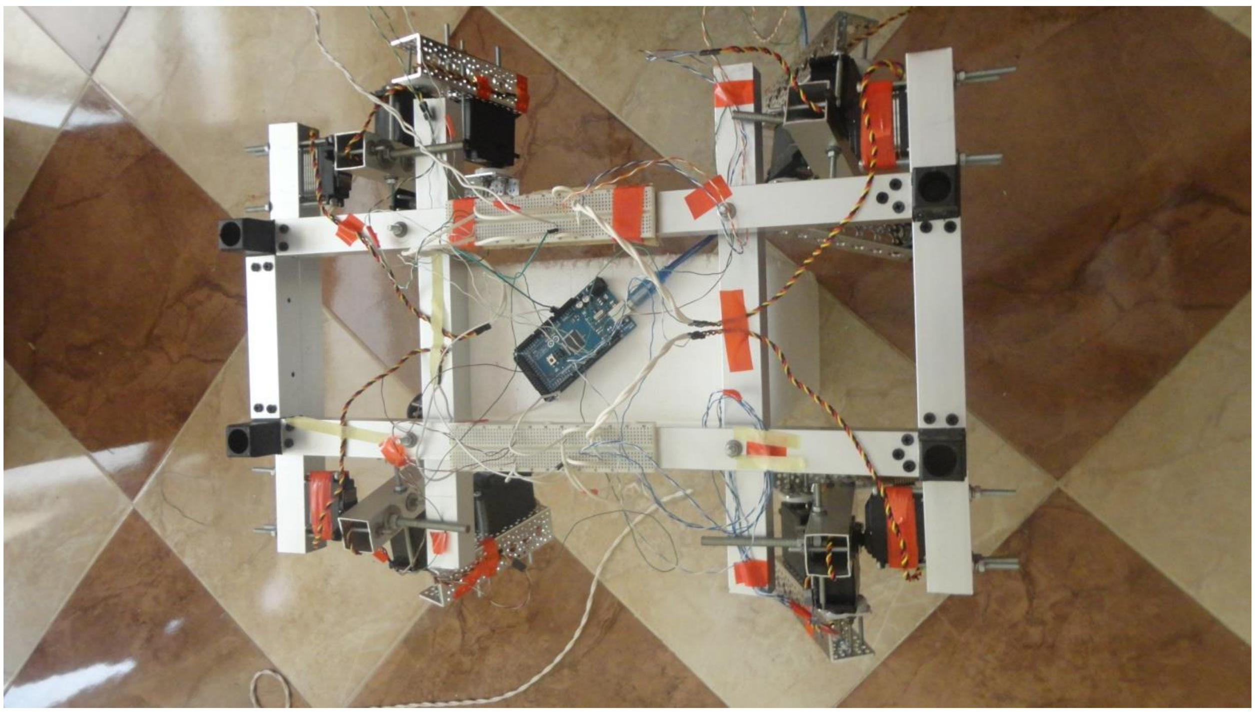

Figure 4: "Shebl" Quadruped Robot

In this chapter we will talk about the main modules in the system and the way they interact so as to accomplish the overall system requirements.

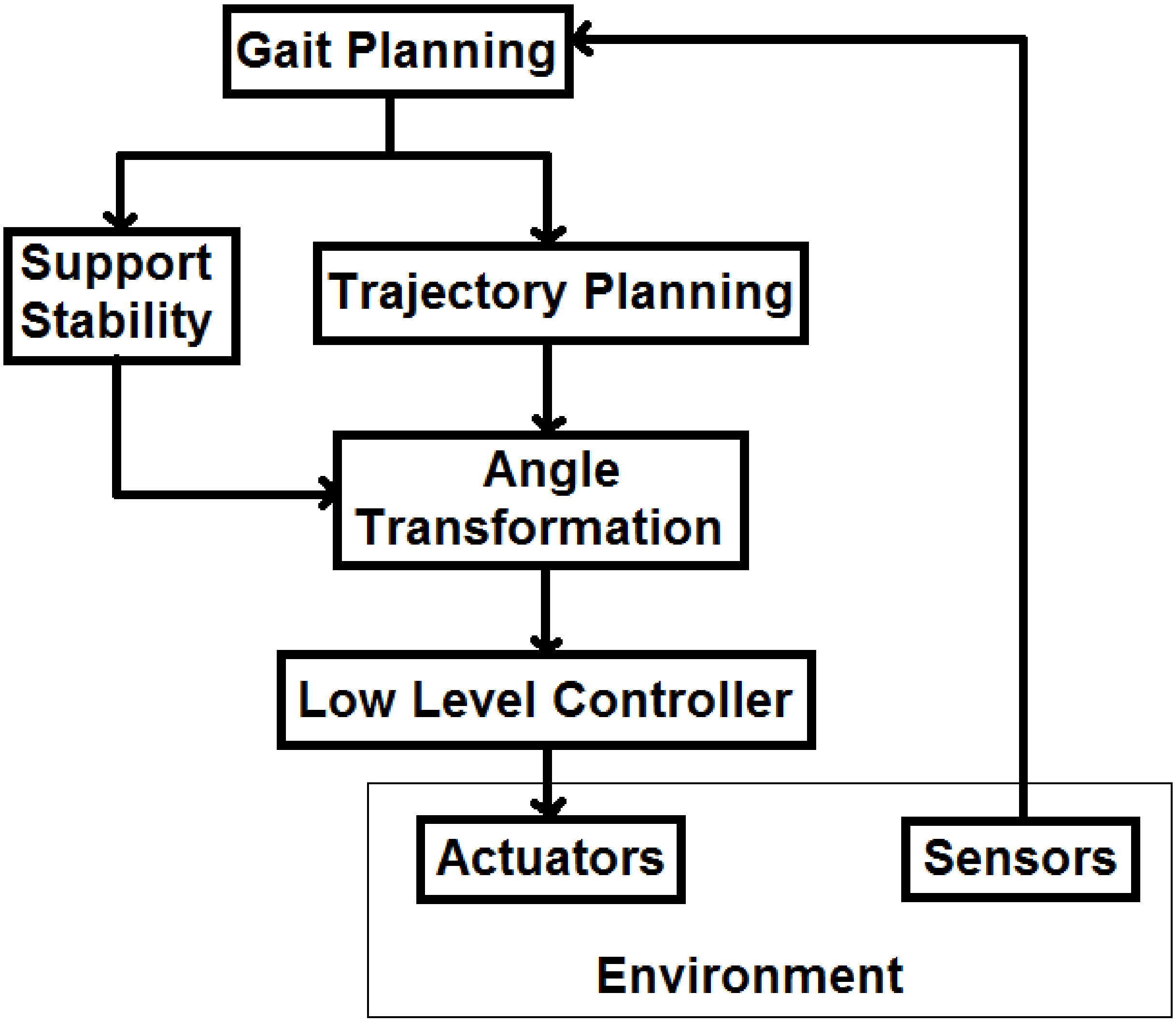

Figure 5: System Architecture

3.1 Hardware Modules

3.1.1 Robot Movement

The movement of the robot is performed via 12 servo motors connected to the 4 legs, each leg with 3 motors.

3.1.1.1 Servo Control

Servo motors are connected through a standard three-wire connection: two wires for a DC power supply and one for control (connected to the Arduino board), carrying the pulses.

Servo motor control from the Arduino board connected to the servos is done by sending each servo a PWM (pulse width modulation) signal, a series of repeating pulses of variable width.

3.1.1.2 Pulse Duration

The angle is determined by the duration of a pulse that is applied to the control wire. This is a form of pulse-width modulation, however servo position is not defined by the PWM duty cycle (i.e., ON vs. OFF time) but only by the duration of the pulse. The length of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position).

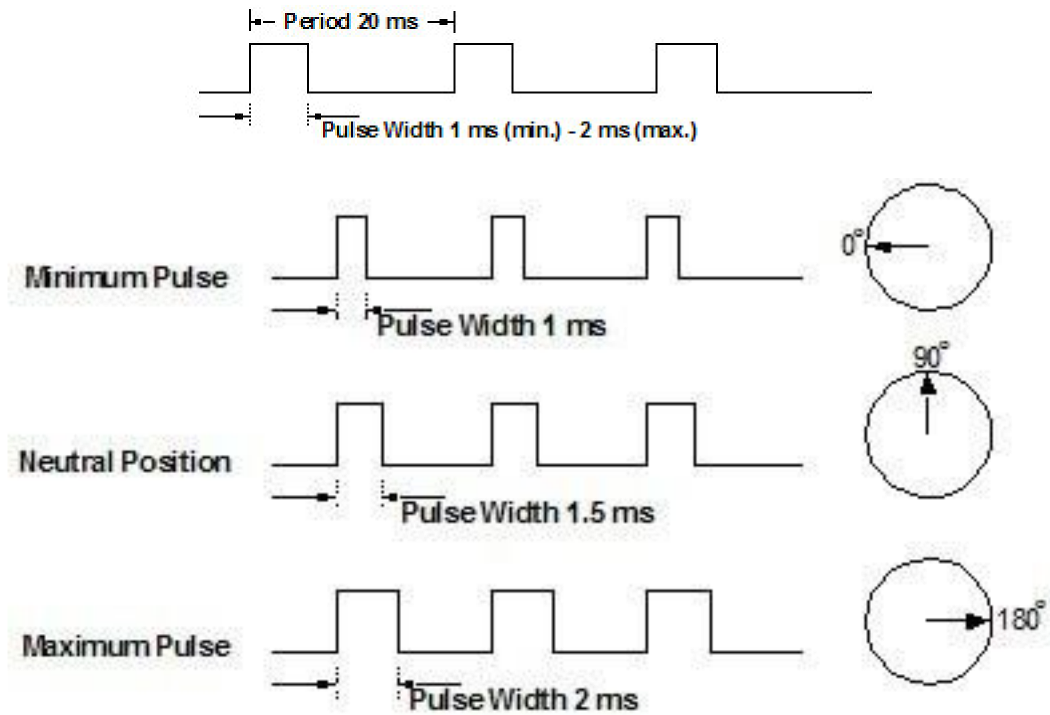

Figure 6: PWM control signal

By outputting the right PWM (pulse width modulation) signal on each of the motors, the robot can move its legs in the required direction.

3.1.1.3 Force

When these servos are commanded to move they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

3.1.2 Robot State

State of the robot can be determined by measuring the current angles of each joint of the robot chassis.

Those measurements can be determined through gyroscope sensors located at each joint of the robot chassis.

3.1.3 Robot Planner

The function of this module is to control the previous two modules via the Arduino board.

This control is done by taking the reading from the gyroscope sensors; performing some calculations then send the control signals to the motors to move the robot in a specific order.

3.2 Software Modules

3.2.1 Trajectory Planning Module

The function of this module is to move the robot legs from point A to point B using an algorithm called simple box.

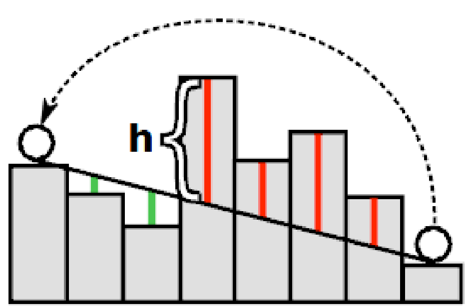

Figure 7: Trajectory Planning

3.2.2 Gait Planning Module

The function of this module is the formulation and selection of a sequence of coordinated leg and body motions that propel a legged robot along a desired path.

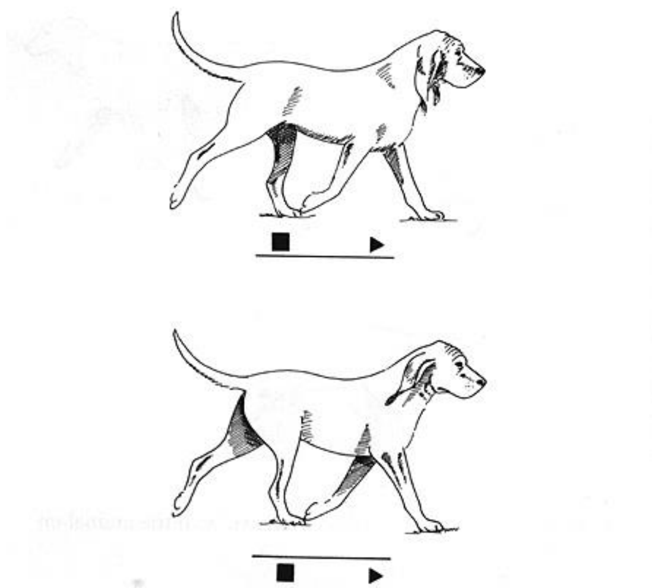

FIG. 91-6 The pace shows two feet to be on the ground at all times.

Figure 8: Gait Planning

3.2.3 Servo Motors Manipulation Module

This software module located at the Arduino board sets the required PWM control signal of each motor to achieve a specific move.

3.2.4 Gyroscope Sensors Module

This software module located at the Arduino board is responsible for get the angular velocity reading from each sensor and integrate those readings to get the current angle of each joint as mentioned before.

The gyroscope gives us the rate of change of the angular position over time (angular velocity) with a unit of [degree/s]. This means that we get the derivative of the angular position over time.

$$\dot{\theta} = \frac{d\theta}{dt}$$

To obtain the angular position, we can simply integrate the angular velocity. So, assuming that at t=0 theta=0, we can find the angular position at any given moment t with the following equation:

$$\theta(t) = \int_0^t \dot{\theta}(t)dt \approx \sum_0^t \dot{\theta}(t)T_s$$

Because we can't take a perfectly continuous integral, we have to take the sum of a finite number of samples taken at a constant interval Ts. Ts is called the sampling period. Of course this approximation will introduce errors. When gyroscope data changes faster than the sampling frequency, we will not detect it, and the integral approximation will be incorrect. This error is called drift, as it increases in time. It results in the sensor reading not returning to 0 at the rest position. For this, it is important that we choose a good sampling period.

Chapter4: Mechanical Design

Our Robot "Shebl" is a fully torque-controlled actuated Quadruped robot with set of 12 active joints that are used to generate the required Traverse gait driven servo motors. The Robot's leg is designed with 3 degrees of freedom where a joint for the connection with the body & the other 2 joints are used for smooth & fast motion of the leg, the robot body is connected with 4 legs with electrically-actuated joints that add a degree of freedom to move the leg inward or outward the robot's center, the 3 joints of the robot's leg gives high controllability on the whole robot motion and reacting applied forces.

4.1 Mechanical Structure

The Mechanical structure of the robot determines the way of movement of the robot, as the number of DOF affects the smoothness of the motion, as well as the Foot trajectory algorithm and the Gait used.

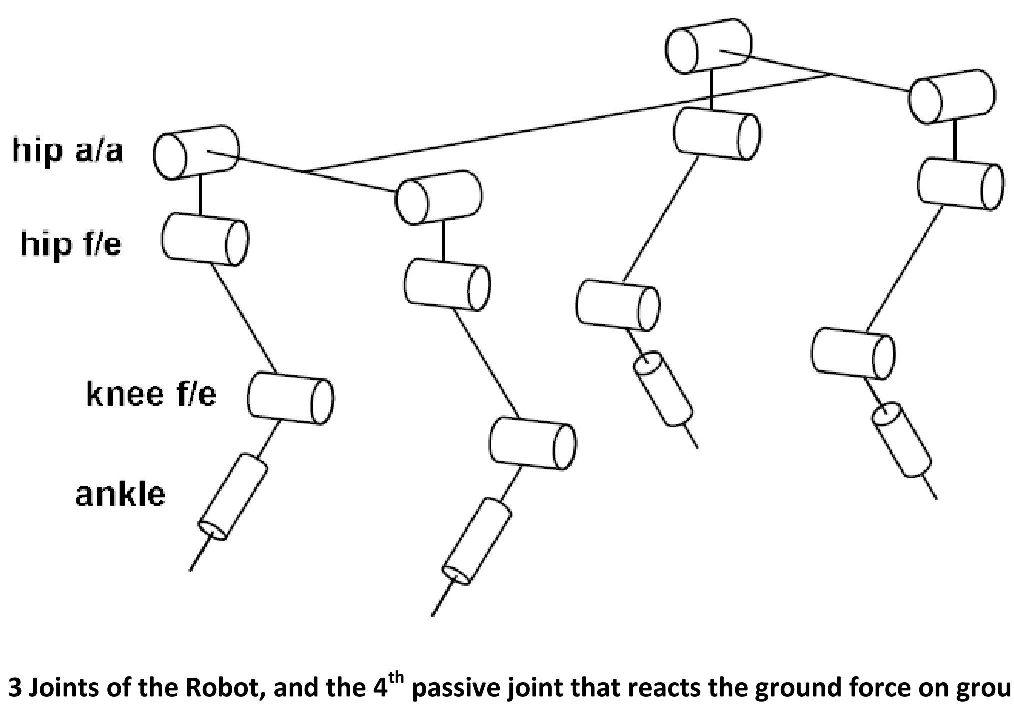

Figure 9: 3 Joints of the Robot, and the 4th passive joint that reacts the ground force on ground contact point

Our Robot has cursorial mammal type of structure with the legs are directly connected to the main body of the robot with strong joints in the leg-sagittal plane, while other robots may have an insect type of structure that has an additional horizontal link sticking out of the body between the body of the robot and the legs this link could be movable or static according to the required number of DOF.

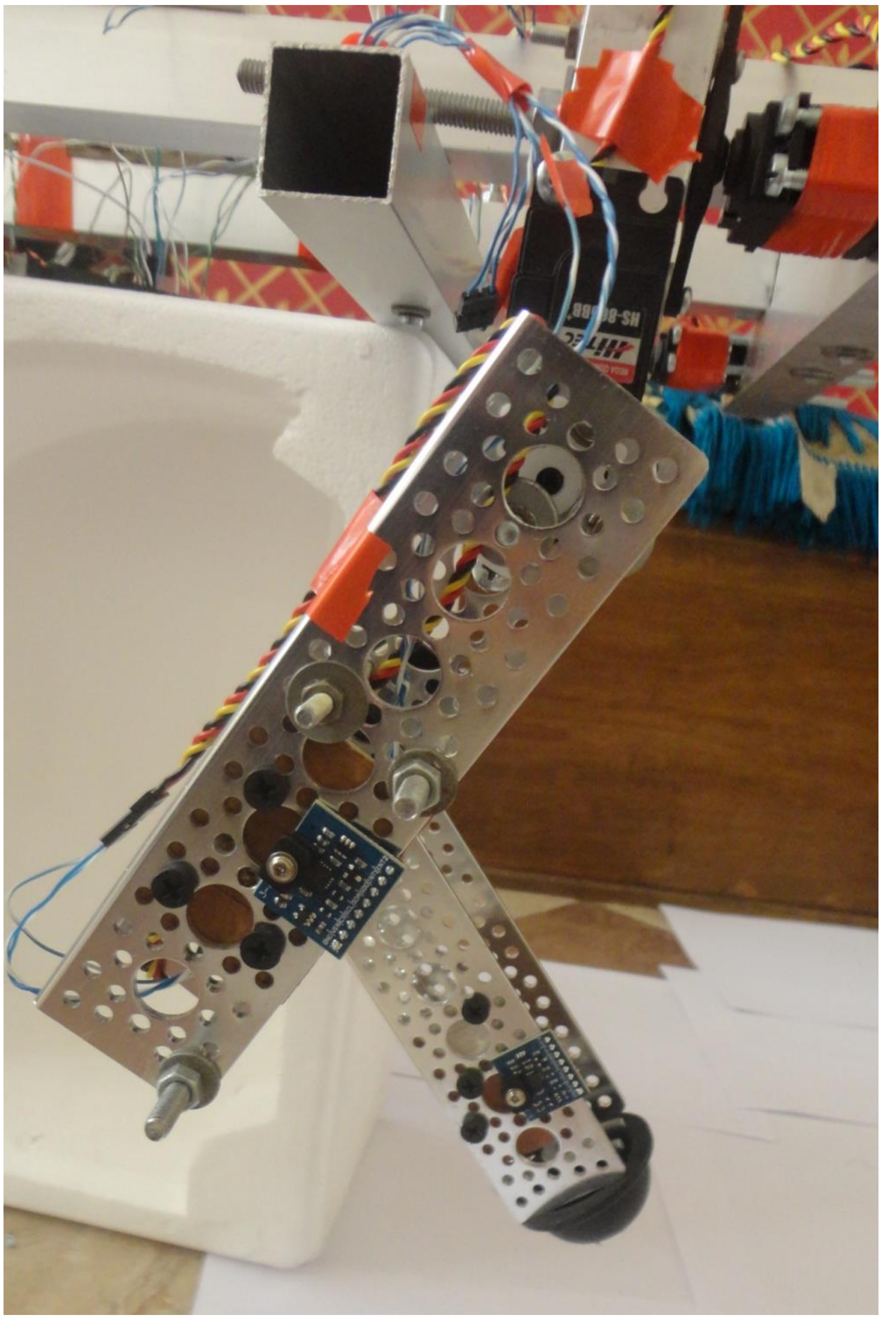

Figure 10: Backward Right Leg

The main actuators that affect the movement of the robot is the joints in the lower hip and knee joints where they are responsible of traversing the foot of the leg from point to point. While the 3rd joint in the upper hip increase the controllability of the system on the robot movement, it is used to adapt the robot movement through rough terrain and external forces.

4.1.1 Number of legs

The decision to build a quadruped robot rather than a biped or hexapod seems to be a good choice from the point of view of a system and controller designer. Four legs are less complicated to construct and maintain than six. The robot costs less and its weight is lower. Furthermore, the computational unit is simpler since fewer sensors have to be sampled and fewer actuators controlled. The power system can also be smaller. In terms of controller design, obviously it is less demanding to coordinate and control four than six legs.

With above reasoning a biped would be better than a quadruped. The control of bipeds, however, is considered to be difficult due to their small support base. An active balancing control is necessary and therefore a much more complex control system. On the other hand, quadruped is stable as soon as three or four feet are on the ground and the COM of the robot is inside the support polygon.

4.1.2 Leg Configuration

The configuration of legs on the body of the robot was experimented by multiple researches and projects and “BiosBot” robot was one of the projects used for this reason, it was experimented to get the best configuration of legs, and concluded that “X” design (forward/Backward) configuration is the most suitable; as it reduces the slipping between the feet & the ground, and it increased the stability by decreasing the robot pitching motion.

That is done assuming that the difference between the front & back legs will be neglected & the front legs won’t have big freedom in motion as in mammal animals.

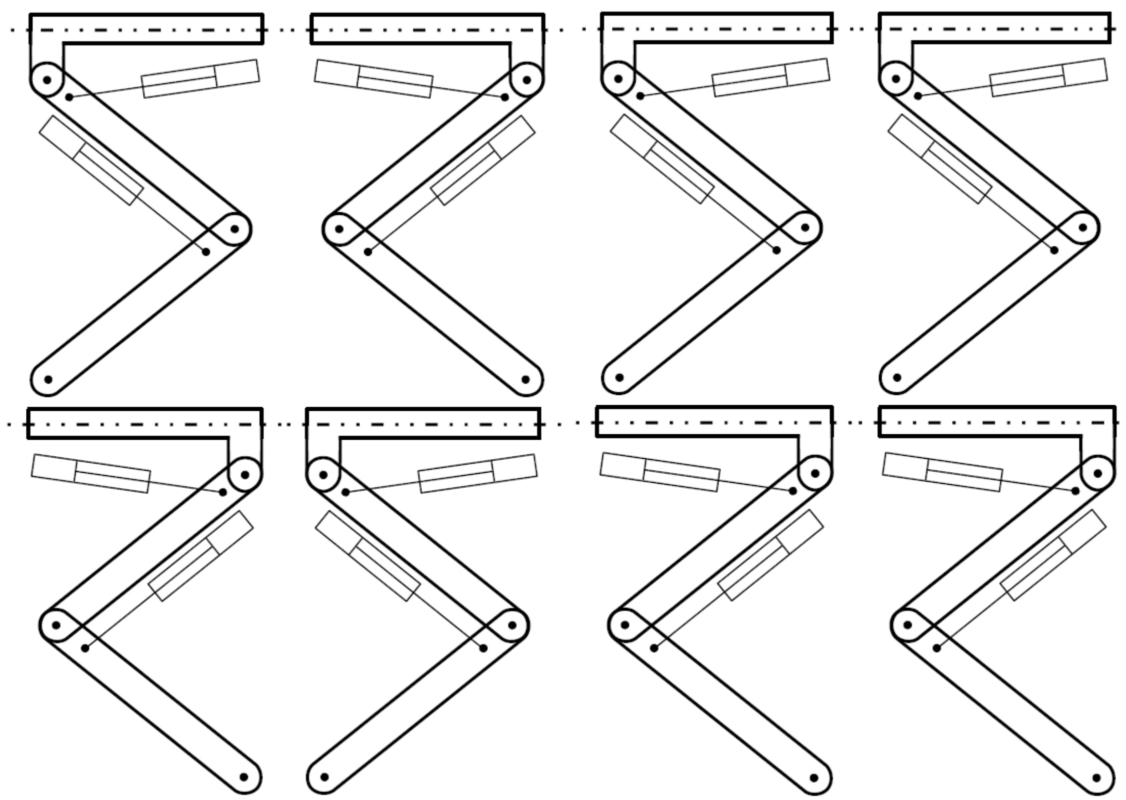

Figure 11: Leg configurations of quadruped robots with forward walking direction to the right

4.1.3 Ground Contact Point

The usage of point-foot instead of having a flat-foot for the leg has multiple of advantages as Handling Loads, as the load of the robot is directed on one point, the foot is passive as there is no dynamic response needed in the control algorithm and no actuators required in the foot, also it reduce the complexity of the control algorithm by removing the computations needed for Foot-to-Leg attachment and Foot trajectory planning.

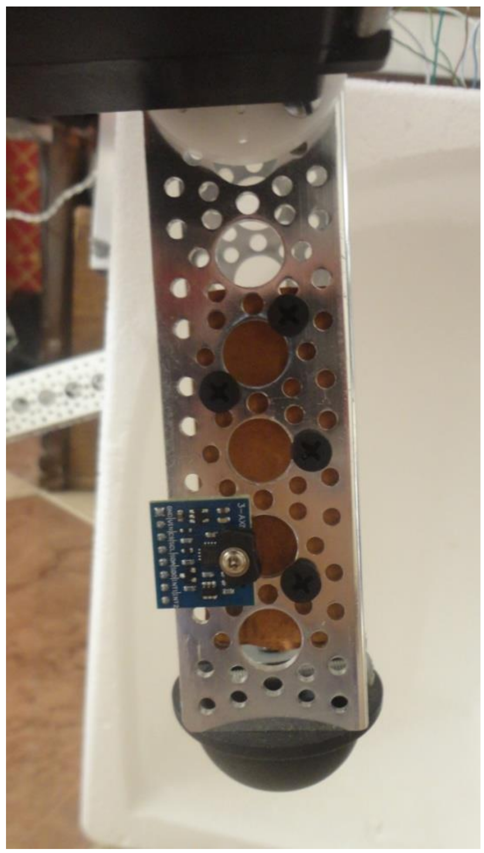

Figure 12: Rubber foot contact at the end of the Leg

Despite these advantages, there is still risk of losing the robot stability by traction or slipping but it is solved by using a rough rubber at the end of the leg, while flat-foot needs to have a DOF in the direction normal to the motion for the leg stability.

4.2 Actuators

The fundamental principle of all electrical actuators is based on the force that is generated by an electric current owing inside the wire of a coil in the presence of a magnetic field. Brushed DC motors are one of the simplest electric actuators as it is constructed with a wound rotor (with coils) and a stator with either permanent magnets or coils. They usually have two or more poles and use direct current and brushes to alternately power the rotor coils.

Brushless DC motors do not have brushes to accomplish the switching of the coils. Instead, they need an electronically controlled commutation system. While the rotor contains the permanent magnet, the coils are located on the outside at the stator. This way, the problem of leading current to a moving armature is solved. However, sensors that measure the position of the rotor become necessary.

Servo Motors are Geared DC Motors that have high speed and torque capabilities, higher power density, low maintenance and improved efficiency in comparison with ordinary DC motors, Servo motors depends on rotary encoders to get the orientation of the motor axis in instance of time, beside having an additional control circuit that process the required angle that is determined by PWM control signal as input.

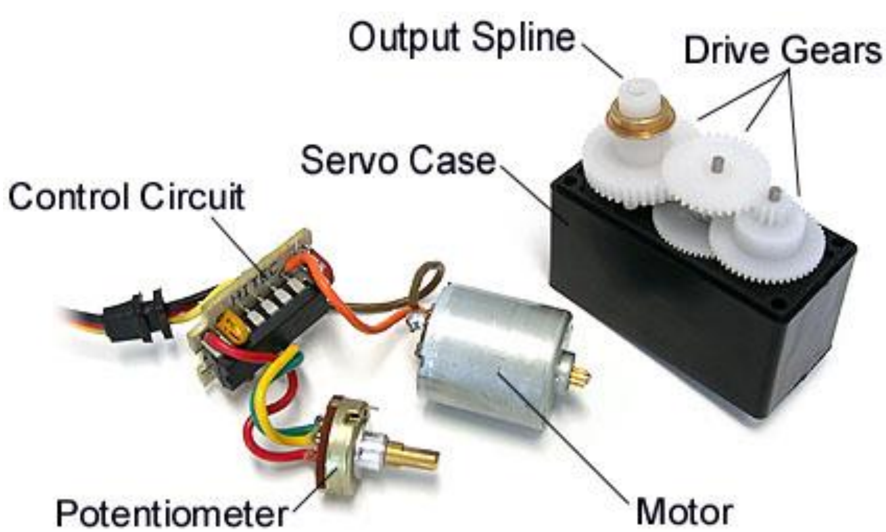

Figure 13: A Servo Motor broken down

4.3 Actuators Details

There are 2 different types Servo Motors used in the robot to satisfy the required torques at each joint. The first motor is “HS-805BB Mega Power” Servo used for the lower hip and the knee, it has maximum torque 19.8 kg.cm at 4.8V operating voltage. The motor has maximum speed of 0.19 sec/60degree at no load state that is about 315.79 degree per second speed. Dual ball bearings, extra thick nylon gears, heavy H-bridge and large output spline. The motor is configured to have reverse rotation and 180 degree rotation.

Figure 14: HS-805BB Mega Power Servo Motor

The second motor used for the upper hip of the robot is “HS-7908TH Servo” Designed to operate on a two cell LiPo Pack. Features the newest high resolution "G2.5" 12 bit generation programmable digital circuit and Titanium gears, it has maximum torque 36 kg.cm at 6V operating voltage. The motor has maximum speed of 0.21 sec/60degree at no load state that is about 285.7 degree per second speed. The motor is configured to have reverse rotation and 180 degree rotation. Other features in the HS-7980TH include a 7.4V optimized coreless motor, integrated heat sink case, and a top case with two hardened steel gear pins supported by axial brass bushing.

Figure 15: HS-7980TH Servo Motor

4.4 Structure Specifications

A summary of the robot's details and mechanical specifications is shown in the following Table.

Description	Value
Weight	xx kg
Body Length	48 cm
Body Width	30 cm
Link 0 length (upper hip)	8 cm
Link 1 length (lower hip)	10.5 cm
Link 2 length (knee)	13.1 cm
Link 3 length (foot)	1.8cm

Table 2: List of Specifications and Dimensions of the Robot

Chapter5: Hardware Design

5.1 Arduino Mega 2560

5.1.1 Description

The Arduino Mega 2560 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 an AC-to-DC adapter or battery to get started.

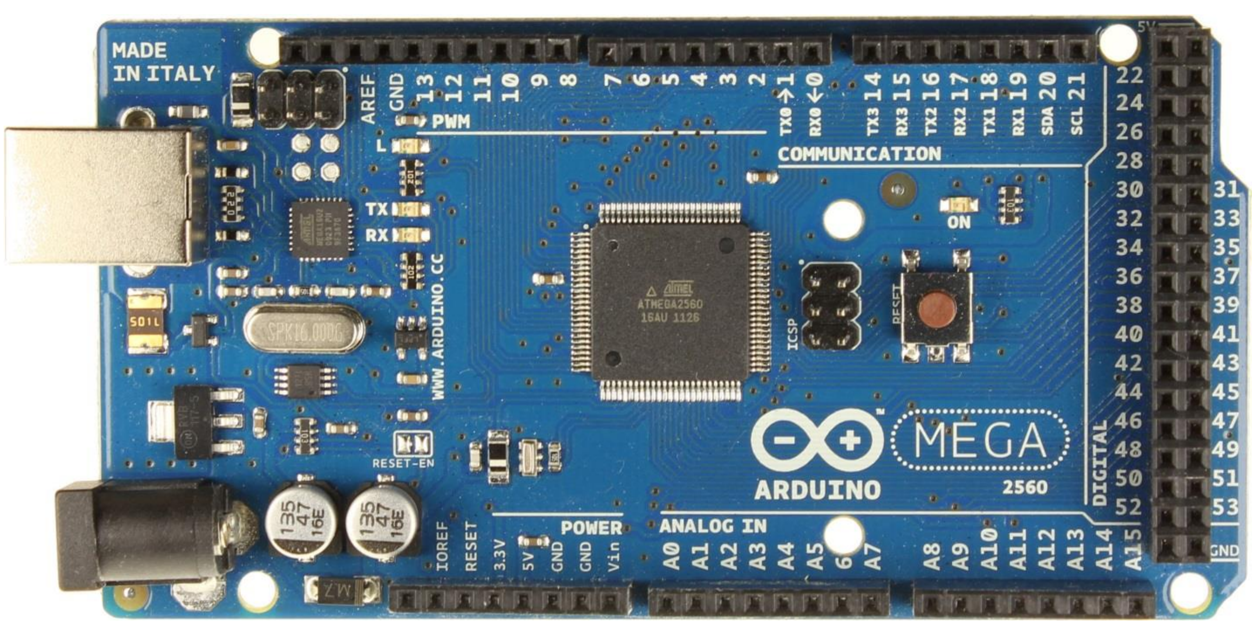

Figure 16: Arduino mega 2560 board

5.1.2 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

Table 3: Specifications of Arduino Mega 2560 Board

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.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

5.1.3 Programming

The Arduino Mega can be programmed with the Arduino software.

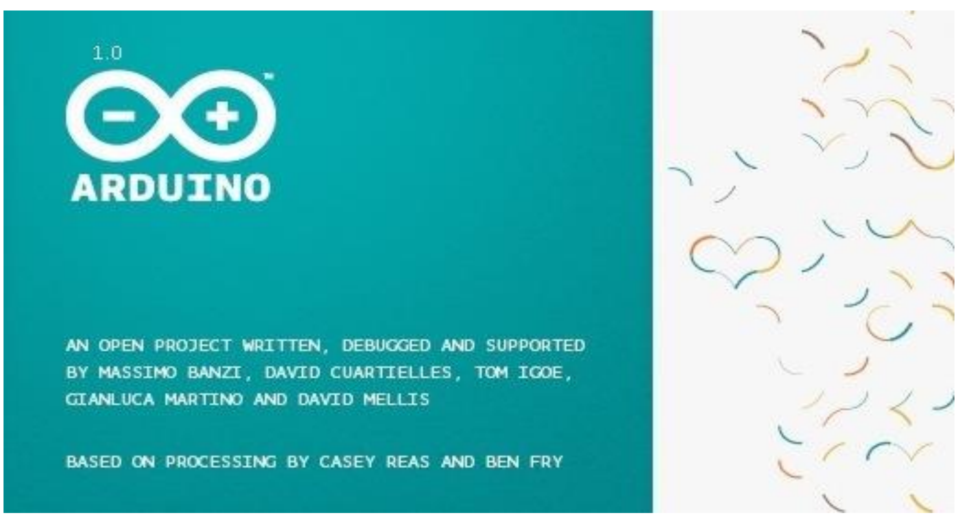

Figure 17: Arduino software

The ATmega2560 on the Arduino Mega comes pre-burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

5.2 Gyroscope sensor

5.2.1 Description

The L3G4200D is a low-power three-axis angular rate sensor able to provide unprecedented stability of zero rate level and sensitivity over temperature and time. It includes a sensing element and an IC interface capable of providing the measured angular rate to the external world through a digital interface (I2C/SPI).

The sensing element is manufactured using a dedicated micro-machining process developed by STMicroelectronics to produce inertial sensors and actuators on silicon wafers.

The IC interface is manufactured using a CMOS process that allows a high level of integration to design a dedicated circuit which is trimmed to better match the sensing element characteristics.

The L3G4200D has a full scale of ±250/±500/ ±2000 dps and is capable of measuring rates with a user-selectable bandwidth.

The L3G4200D is available in a plastic land grid array (LGA) package and can operate within a temperature range of -40 °C to +85 °C.

Figure 18: Gyroscope Sensor L3G4200D

5.2.2 Specifications

- Power Requirements: 2.7 to 6.5 VDC
- Communication Interface: I2C (up to 400 kHz) or SPI (10 MHz; 4 & 3 wire)
- Operating temperature: -40 to +185 °F (-40 to +85 °C)
- Dimensions: 0.85 X 0.80 in (2.16 X 2.03 cm)

5.2.3 Features

- Three selectable full scales (250/500/2000 dps)
- I^2C/SPI digital output interface
- 16 bit-rate value data output
- 8-bit temperature data output
- Two digital output lines (interrupt and data ready)
- Integrated low- and high-pass filters with user-selectable bandwidth
- Ultra-stable over temperature and time
- Wide supply voltage: 2.4 V to 3.6 V
- Low voltage-compatible IOs (1.8 V)
- Embedded power-down and sleep mode
- Embedded temperature sensor
- Embedded FIFO
- High shock survivability
- Extended operating temperature range (-40 °C to +85 °C)

5.2.4 Applications

- Gaming and virtual reality input devices
- Motion control with MMI (man-machine interface)
- GPS navigation systems
- Appliances and robotics

5.2.5 Pin definitions

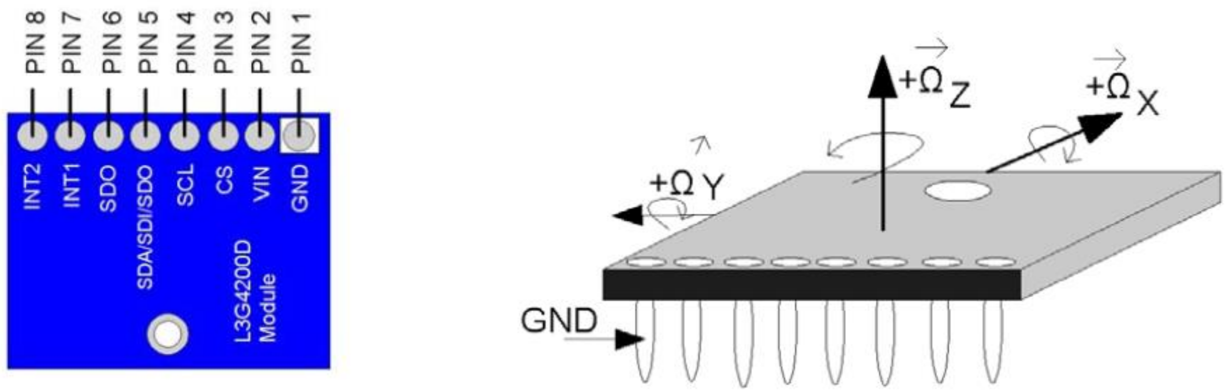

Figure 19: Gyroscope pin mapping and axes

Pin Definitions and Ratings

Pin	Name	Type	Function
1	GND	G	0V Supply, Ground Pin
2	VIN	P	Supply Voltage from +2.7 – +6.5VDC
3	CS	I	SPI enable (Default is I^2C enabled) I^2C/SPI mode selection (1: I^2C communication enabled; 0: SPI communication mode / I2C disabled)
4	SCL	I	I^2C & SPI serial clock (SCL)
5	SDA/SDI/SDO	IO	I^2C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)
6	SDO	O	SPI serial data output (SDO) I^2C least significant bit of the device address (SA0)
7	INT1	I	Programmable interrupt, see datasheet for more details
8	INT2	I	Data ready/FIFO interrupt, see datasheet for more details

Pin Type: P = Power, G = Ground, I = Input, O = Output

Table 4: Gyroscope Pin Definitions

Chapter6: Software Architecture

The Robot control software combines aspects of deliberative and reactive controllers. The main loop of the controller is a gait deliberative planning algorithm that plans the next foot step and determines the motion of the legs and body to achieve the step according to the determined gait while maintaining stability. This deliberative layer addresses the problem of rough terrain when choosing the foot step but neglects foot slippage, accidental collision, and modeling and sensor errors. Reactive modules were created to address specific problems that arose from these issues.

6.1 Gait Deliberative Planning Module

6.1.1 State Machine

The deliberative controller was implemented as a state machine. The first phase is Gait Selection, which chooses the next leg to swing. The second phase is an optional body shift with all four feet on the ground, which we refer to as a Quad Shift. The Quad Shift phase moves the body to position the center of mass inside the upcoming support polygon to ensure stability. The third phase is the Swing, which swings the leg and may involve a simultaneous body shift. This simple three phase model allowed us to achieve several gaits. The only knowledge shared between these states is the swing leg identified in the gait generation phase.

The algorithm starts by deciding which leg to swing in gait generation. The body is then shifted on all four legs during Quad Shift to position the COM inside the support triangle of the non swing feet. The robot then executes the swing while possibly shifting the body.

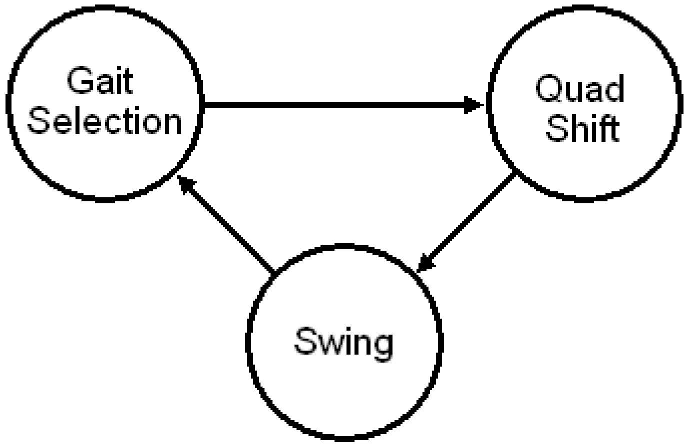

Figure 20: A partial state machine of a single swing leg cycle

We use the concept of a *stance* to define the configuration of the robot at any point in time. The stance is composed of body position and orientation and the location of all four feet. Any state that results in motion will output a series of waypoint stances that start with the current stance and end with the final desired stance of the state.

These stances are then played back by performing inverse kinematics and passing the resulting desired joint angles to a low level servo motor controller each control cycle. The transition to the next state occurs when all stances in the current stance list have been played.

6.1.2 Gait Selection Phase

Assuming terrain that allows a fairly straight path, we use the standard gait identified by Muybridge. This gait is sufficient to provide reasonably fast movement on flat terrain and adapts well to variation in mildly rough terrain.

For our assumption, the gait of the robot is pre-defined to be the motion of the legs in the sequence of front right leg, back right leg, back left leg, then front left leg. This sequence is simple traverse walking gait that produce a static motion for flat terrains.

6.1.3 Quad Shift Phase

The Quad Shift phase ensures stability of the initial support triangle of the upcoming swing phase by shifting the body center. In practice, it is useful to minimize the time spent in the Quad Shift phase since no leg is swinging.

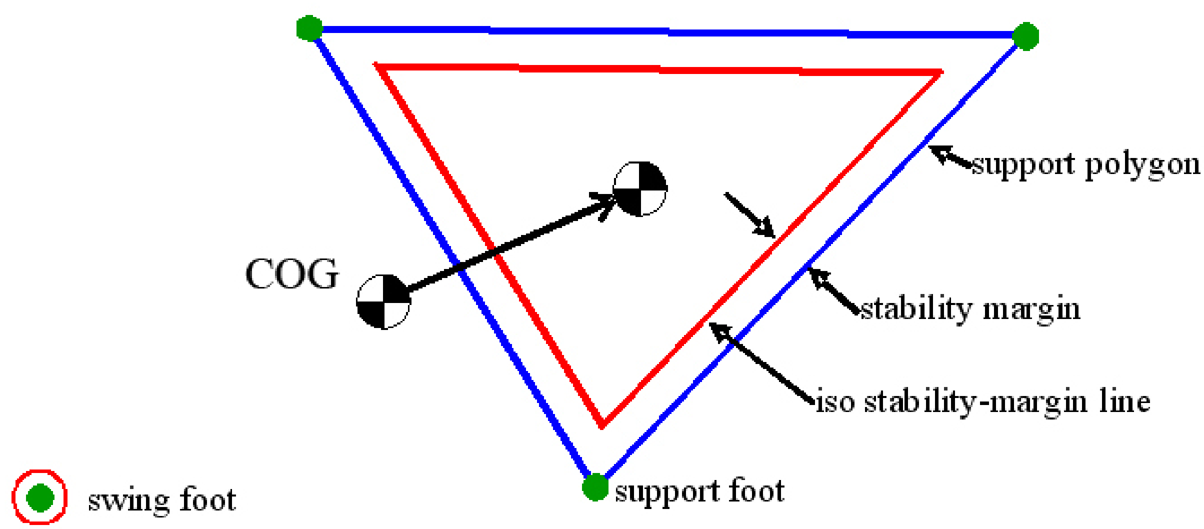

Figure 21: The Stable Quad Shift Algorithm

Using a fixed gait as traverse walking makes the algorithm results in a shift only on hind leg swings, as the COM is already inside the support polygon after the hind swing due to overlapping of support triangles.

6.1.4 Swing Phase

The swing phase determines a trajectory for the swing leg to take and possibly a shift as well. The trajectory of the swing is just creating the required point to reach in the swing phase and the height of the leg lifting, while the trajectory is generated in the trajectory planning module, currently the trajectory parameters are statically determined in the algorithm.

The shift generation is using the robot’s full body range to extend its reach during the swing phase. This method generates a single step and some fairly uniform height that permits a continuous standard gait with the body staying in constant motion while traversing the terrain.

The swing shift generation algorithm is

```
IF Backward Leg:
        THEN shift the COM to the median of the support stability triangle.
IF Front Leg:
        THEN shift the COM to the front stability boundary.
```

6.2 Trajectory Planning Module

The trajectory planning module produces a simple box trajectory for the swing foot that start from the current position of the foot as initial state and ends at the desired end point referenced from the gait planning module. In the simple box trajectory the foot is raised a constant height above the ground. Then a vertical movement to the desired point is done before placing the leg down to the ground.

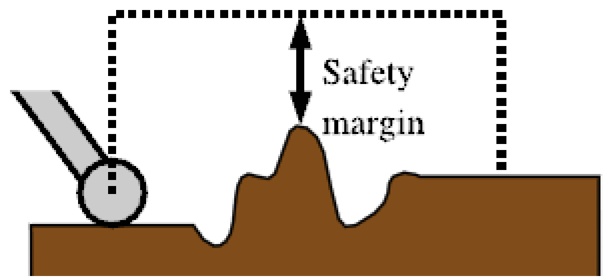

Figure 22: Desired trajectory for moving feet over obstacles

The algorithm depends on pre-computed path for simplicity as the path is computed according to the trajectory parameters determined from the gait planning module, the algorithm starts by specifying the milestone points as being the vertices of the box trajectory, then using inverse position kinematics the possible orientation of the leg is generated to reach the required vertices.

6.3 Reactive Modules

The deliberative algorithm alone fails under certain conditions. These failure modes were identified and addressed by specific modules with control components combining aspects of reactive and deliberative schemes appropriate to the problem they address.

6.3.1 Height Controller

Foot slippage can cause discrepancies between the current stance of the robot and the current desired stance in the list. These foot location errors translate into height, pitch and roll errors, the pitch and roll controller reactively cancels these errors by adjusting each support leg's extension. The current implementation of the controller adjusts the robot's height from the ground to a specific height that tries to adapt the robot's legs to that height.

6.3.2 Low Level Servo Controller

Stances are generated by the deliberative modules for playback on the robot, and are then modified by the reactive control modules to account for accumulated error. Every control tick the controller performs inverse position kinematics on the desired stance to produce desired joint positions. These positions are then transmitted to the Arduino Board on the robot, which generates the proper PWM for each joint, and then a low level P control is performed on the individual joints of the robot by the P controller implemented on the Servo motor micro-controller.

Chapter7: Experimental Results

Set of Experiments are done on the Robot and its components to adjust the behavior of the system and the controller.

7.1 Leg Trajectory Following

This experiment is performed with 2 of 3 DOF of the robot's leg while attached to the robot's body. The main test is to check the tracking performance of the leg along the simple box trajectory path generated by the trajectory planning module and its reaction if the leg didn't reach to the desired point.

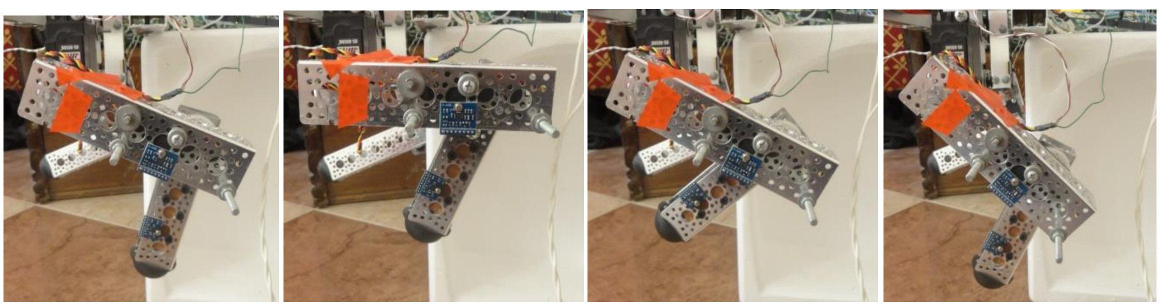

Figure 23: Actual Movement of the Leg

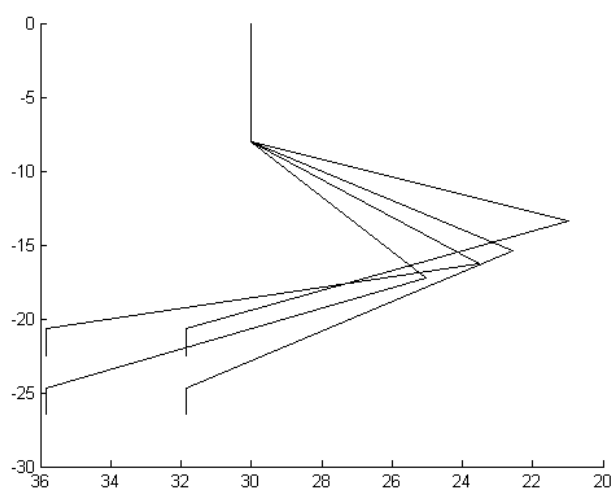

Figure 24: Simulated Movement of the Leg

The results shows that the servo motor P controller will adapt the leg orientation during motion till the motor reach the reference point that is pre-computed in the trajectory planning module, as long as the motor can handle the opposite torque.

7.2 Power Consumption tests

The Servo motors we are dealing with has a relatively high torque with respect to voltage, as it works at 5V that makes the motor requires high electric current to apply the torque on load. The test was made manually using ammeter as the datasheet of the motors doesn’t specify the current required for the motor.

Our results showed that the HS-7980TH Motor consumes about 0.8A during motion and requires 2A at start time, which also may overshoot to 5A. While the HS-805BB Motor consumes about 0.8A during motion and requires 1.3A at start time, which also may overshoot to 3A.

7.3 Servo Motor Tests

This test is used to calibrate the angles of the servo motor that is defined from the orientation reference of the servo motor, the motor shaft is centered on “90” degree, then we get the minimum time required for the motor with the robot load to perform the motion between angles.

The reason of the test is to adjust the motor axis on the robot axis and define the transformation equations to get the actual motor angles from the reference angle of the joint in the robot coordinates.

7.4 Static Stand of the Robot

The test of the robot standing is to determine the effect of the robot's weight on the drift in the motor angles that occurs due to the external torques applied by the robot's weight.

7.5 Blind Walking

The blind walking test is a simple gait pre-computed on the robot's model and applied without any feedback from the world. The main target is to determine the model error and the motors calibration error, with a know sequence of footsteps.

7.6 Gyroscope Tests

Testing the Gyroscope readings to calibrate the readings and remove the noise component from the current readings using a simple Kalman filter, and detect any drift in the gyroscope axis from the robot axis.

7.7 Simple Walking

The main experiment as it tests the entire robot. The test is that the robot moves along with the gait planning algorithm module and the state of the robot is changed according to the sensors readings from the world. This experiment should show how the gait planning module and trajectory planning module adapt the robot with the simple changes in the environment.

Chapter8: Conclusion

Legged robots are a popular field of research. While the main attention is given to biped and especially humanoid robots, quadruped machines promise to provide an even higher mobility in terms of moving over rough terrain, especially outside. Unfortunately, the control of the motion and stability of the robot isn't that easy, their potential for handling certain types of difficult terrain should be utilized. This project work has dealt with statically balanced gaits for quadruped robots. The main emphasis has been on the stability of such gaits and how they can be accomplished. The motivation for studying statically balanced gaits is that they can be executed arbitrarily slow. This is beneficial in the event that the terrain is uncertain and the robot has to search for footholds, as the robot can remain balanced at all times.

8.1 Main Conclusions

The focus of this project has been to develop and implement a method to achieve a statically balanced gait for our quadruped robot "Shebl". As a goal, this has been accomplished with the center of mass used to develop a stability measure. The support stability polygon is used in the planning of a statically balanced body motion. The motion of the robot is planned by determining the supporting position for each leg, which in turn will determine how the robot should shift its weight in order to remain statically balanced. The approach proposed solves simultaneously the problem of determining a statically balanced motion trajectory for the body, as well as, the distribution of forces to the feet, to compensate for the weight of the robot.

8.2 Challenges

8.2.1 Time Constraints

We faced three types of constraints, the first one in the research phase as we had to finish the robot structure and the approached solution we will work on to have enough time for implementing and shipping the required components.

The second constraint is the shipment time constraint as we had to wait till the motors are shipped to Egypt to start working on the low level control of the motors, as well as the Aluminum bars to complete the leg of the robot.

The third constraint is in the design and implementation of the System Architecture which is time challenging given the time required to build the robot.

8.2.2 Market's Component

The components available in the market aren't efficient enough for the current design as we intended to work with pneumatic parts but the pneumatic components in the market isn't efficient in the pressure gained or the accuracy of the movement, so we decided to move on to electric motors.

But also the electric motors available in the market isn't efficient as the available torques are small and not enough as the maximum torque found is 20 kg.cm while we need about 40 kg.cm for some motors.

The only solution we had is to order the missing components from distributors outside Egypt, to get efficient components that satisfy the design requirements.

8.2.3 Hardware Challenges

The Hardware had lots of challenges through the project, designing, assembling, testing, and debugging each piece of the robot.

8.2.3.1 Actuators

The Servo motors we used had lots of challenges in assembling them on the robot body or in defining the required control signals to move them properly. The Servo motors had a very limited torques compared to the hydraulic pistons and motors that created some restrictions on the dimensions of the body, and required very light materials to be used as leg and trunk of the robot body.

The Servo motors created some restrictions on the upper control unit too as the required control signals are specified on the angles of the motor that must be calibrated and transformed to the actual angles in the robot kinematics and dynamics.

8.2.3.2 Gyroscopes

The Gyroscopes we are using has only two serial interfaces with the controller, but there is some limitations on the usage of these interfaces, the first serial interface is I2C interface that use a synchronization clock between the devices and a shared bus between the devices connected on the network, the constraint is that the address space of the I2C network is limited to 2 slave devices as the addresses of the gyroscope is hardcoded in the chip.

The second serial interface is SPI interface that use a synchronization clock between the devices, single data bus for the master device to write on, and single data bus for the master device to receive data from it.

Other Challenges is to calibrate the gyroscope reading as the error in gyroscope readings accumulate by the time due to the behavior of the system, so an efficient calibration process is needed to have accurate gyroscope readings.

8.2.3.3 Arduino Board

The Arduino board has limited computation and memory power that requires an outer source of computation that decreases the complexity of the program on the board. Also the Board has limited number of serial interfaces and PWM modules that needs the gyroscopes to be connected in a network manner, and to find enough PWM modules by using Analog ports in the board as PWM generators.

8.2.4 Software Challenges

The software had some challenges too as the Arduino board doesn't support parallel computing or multi-threading that is very important in handling multiple data readings at very short time, as well as synchronizing between the gyroscopes and motors at the same time instance.

The Computation Time is limited by the sampling frequency of the gyroscopes to finish all computations before starting in a new sample input and have enough time to send the response to the motors.

The servo motor control code has some difficulties too as the Servo motor is controlled by the desired angles not by torques, which makes the feedback control loop more complex in behavior as the motor's torque isn't controllable.

Chapter9: Future Work

The goal of the “Shebl” project is to have a walking robot that can handle difficult terrain. As a step towards that goal, several improvements can be made on the controller presented in this section.

9.1 Dynamic Support Stability Polygon

The support stability polygon was only used to plan the motion of the robot in a feedforward manner. However, the support stability polygon can also be used to detect the risk of instability while walking, to allow the robot to take appropriate action to avoid tipping over or to recover from failure.

9.2 Leg Trajectory

The current simple box leg trajectory has multiple disadvantages as the motion isn’t smooth, and the required height position to lift the foot to it is pre-computed, that makes the adaptation of the motion to obstacles harder. A better approach is to use more dynamic leg trajectory planning algorithm, as using a simple parabola trajectory that is computed by determining the desired velocity and swing time of the leg.

9.3 Uneven Terrain Adaptation

To improve the ability to handle uneven terrain, the controller should be able to adapt to the terrain it is walking on. In the current implementation, the height of the legs were fixed, where it was assumed that the terrain the robot is walking on is approximately planar. However, if the ground is not planar, for instance when going up a step this could cause a foot to lose ground contact as the robot is moving. Another problem could be that a foot, that should be placed, would not find any ground contact, or hit the ground earlier than expected causing a large impact. This problem could be solved by getting feedback from the foot contact point to detect if ground is found or not and when, and it could be solved also by using active control of the robot height.

The first solution, the control of the leg during landing can be improved by making it able to handle minor variations in the terrain. For instance, if no ground contact is established, the foot could be moved further downwards, until ground contact is found, or if no ground contact is found, a warning should be sent to a trajectory control level which would take other actions. On the other hand, if the foot hits the ground earlier than expected, the leg should switch to support phase directly, using the actual landing position as the initial condition for the reference position trajectory during the support phase.

The second solution is active control of the height and attitude of the body of the robot. The goal of the attitude controller, given the estimate of the attitude and distance to the ground plane, would be to maintain the body parallel with the ground plane at a desired height above the plane. This can be accomplished without interfering with the current controller by only varying the height of the legs, normal to the ground plane.

9.4 The support ratio

The support ratios are pre-calculated based on the desired positions of the feet. If the desired velocity is varying, the calculated support ratios do not correspond to the real positions of the feet. There are two problems with this, the desired velocity of the center of mass will not be correct and there is a risk that the support stability polygon might be violated.

One approach might be to calculate the support ratio online while robot movement by using the actual position of the feet, and predicting one stride ahead in time, using the desired position of the feet. In that case the support ratios would correspond with the desired velocity and the current support pattern.

Improvements on the support ratio calculations would result in better velocity tracking, smoother motion and improvement of the stability.

9.5 Other Improvements

The current implementation allows further improvements as adding more statically stable gaits as crawling, and improving the gait planning algorithm to process required gait parameters to produce dynamically stable gaits as running and climbing, which requires more complicated stability measurements as illustrated before.

References

Baines, P. J. (1998). Feedback linearized joint torque control of a geared, DC motor driven industrial robot. *International Journal of Robotics Research 17* , 169-192.

Brooks, R. A. (1985). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation, Vol. 2, No. 1* , 14-23.

H. Adachi, M. K. (1997). *Human-operated walking control of a quadruped by event-driven method.* 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97.

Hardarson, F. (2002). *Stability analysis and synthesis of statically balanced walking for quadruped robots.* Stockholm: Royal Institute of Technology, KTH.

J. R. Rebula, P. D. (2007). A controller for the LittleDog quadruped walking on rough terrain. *IEEE International Conference on Robotics and Automation.*

Jimenez, P. d. (1995). Generation of discontinuous gaits for quadruped walking vehicles. *Journal of Robotic Systems* , 599-611.

John R. Rebula, P. D. *A Controller for the LittleDog Quadruped Walking on Rough Terrain.* Pensacola, Florida: Florida Institute for Human and Machine Cognition.

Katsuyoshi Tsujita, K. T. *Adaptive Gait Pattern Control of a Quadruped Locomotion Robot.* Kyoto University, Sabanci University.

Li LIU, A. B. (2000). *Trajectory Planning and Control for a Human-Like Robot Leg With Coupled Neural-Oscillators.* Little Rock: University ofArkansas at Little Rock.

M. Buehler, R. B. (1998). SCOUT: A simple quadruped that walks, climbs, and runs. *IEEE Int. Conf. on Robotics and Automation.*

Marko B. Popovic, A. G. (2005). Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications. *The International Journal of Robotics Research, Vol 24* , 1013-1032.

Muybridge, E. (1957). *Animals in Motion.* Dover.

Paul Vernaza, M. L. *Search-based Planning for a Legged Robot over Rough Terrain.* Philadelphia: GRASP Laboratory, University of Pennsylvania, Philadelphia.

R. Kurazume, S. H. (2001). Feedforward and feedback dynamic. *IEEE Int. Conf. on Robotics and Automation.*

Raibert, M. (1986). *Legged robots that balance.* Cambridge, Massachusetts: MIT Press.

S. Hirose, H. K. (1986). The standard circular gait of a quadruped walking vehicle. *Advanced Robotics, Vol 1* , 143–164.

S. Hirose, K. Y. (1991). Design of prismatic quadruped walking vehicle TITAN VI. *Int. Conf. on Advanced Robotics.*

S.M. Song, a. K. (1989). *Machines that walk: The adaptive suspension.* Massachusetts: MIT Press.

Semini, C. (2010). *HyQ - Design and Development of a Hydraulically Actuated Quadruped Robot.* Genoa: Italian Institute of Technology (IIT).

Tsujita, K. T. (1999). A principle of design of an Autonomous Mobile Robot. *4th International Symposium on Artificial Life and Robotics, Vol. 1*, (pp. 320-323).

Zhe Tang, C. Z. (2003). Trajectory Planning for Smooth Transition of a Biped Robot. *2003 IEEE International Conference on Robotics & Automation*, (pp. 2455-2460).

Appendix A: Feasibility Study

Technology & System Feasibility

Hardware Requirements

The robot has lots of hardware requirements it needs:

- Efficient high torque servo motors, which isn't available in the Egyptian market.
- Very sensitive gyroscopes.
- Pressure Sensors.
- Controller Board, that is Arduino Mega Board in our project to have enough input and output ports to control 12 motor and get readings from 12 gyroscope, IMU, and 4 pressure sensors.
- High Switched-mode Power Supplies.
- And finally High current electric wiring.

Software Requirements

The project is built using:

- Arduino IDE
- Java SDK
- Matlab

The low level controller is implemented by the Arduino IDE to be executed on the Arduino Mega 2560 Board, while the Gait Planning and Trajectory Planning modules are implemented on Java SDK to be executed on computer and then communicate with the Arduino Board using USB Library.

And finally Matlab is used to create the robot model and simulation before actual work in implementation.

Input and Output

The system needs some pre-defined user configuration as the step size and the height of the step that determines the movement parameters of the robot.

While the Output is the system movement that is performed using the actuators assembled on the robot body.

Economic Feasibility

The Cost/Benefit study shows that the main part of the costs are reusable components as Servo Motors and Gyroscopes that are used along the project life time, these components aren't affected by the change in the software architecture of the robot, so using a parabola trajectory path instead of using a simple box will not change the hardware requirements of the robot.

Cost

The actual cost of the robot depends on the accuracy and the properties of the components, as the lower efficiency components used the lower the cost will be. The following table shows the cost of the whole project with total cost of 12145 EGP.

	Part#	Unit Price	Quantity	Total Price
Body	-	55	1	55
6" Aluminum Ch	585446	41.574	8	332.592
Geared Servo	HS-7980TH	1004.71	4	4018.82
Small Servo	HS-805BB	346.45	8	2771.6
Wheel Arm	155SH	3.4645	8	27.716
Shipment	-	1301.27	1	1301.27
Controller	Arduino Mega 2560	330	1	330
Limit Switch	-	2	4	8
Gyroscope	L3G4200D	250	12	3000
Power Source	SMPS 5v 10A	150	2	300

Table 5: Project Costs

Benefit

The project is a stepping stone towards building capacity to produce practical legged robotic systems. This can have many applications in the Egyptian market in areas with rough terrain both for strategic applications such as removal of mines in the western desert, disaster response, law enforcement, and some industrial applications in factory automation setups and even consumer applications such as automated elderly assistants. All of these and many other applications share in the same basic principles and technologies that we aim to build know-how and capacity in and deliver products, such as, feedback control, path planning, mechatronics and embedded systems. These technologies can obviously be extended to other products in many other fields and applications.

Legal Feasibility

The project doesn’t conflict with any Intellectual property, as all the sources of information and other projects similar to ours are published in Technical journals and conferences, and the main algorithm of the gait planning algorithm was published in 1985 by Brook in IEEE Journal of Robotics and Automation.

Operational Feasibility

The approach presented in the document contributes in:

- The creation of a versatile robotic platform able to perform basic balanced operations as walking on a straight terrain.
- To study and test the applicability of legged robots to traverse through the world.
- To evaluated low-level control algorithms, system configurations and to test the robot’s motion speed.
- To study biologically inspired locomotion focusing on statically balanced gaits, that can be extended to dynamic gaits, also the importance of gait pattern generation, gait transition and robot balancing skills.

Schedule Feasibility

Building the robot took about 3 months to be completely assembled with actuators. The main time consumption phase is the research phase as the design of the robot needs to be done after studying the gaits of animals and how the structure of the robot could help in the robot movement, and also figuring out a good efficient way to process the environment data and decide which process should be executed next according to the situation.

Appendix B: Matlab Kinematic Codes

All the kinematics equations and formulas are computed as the contact point of the leg with the robot body is the reference origin point.

Direct Position Kinematics

```
%Given The desired Joint's angles
%End-Effector foot position
X = L1*sin(T1) + L2*sin(T1-T2);
Y = -sin(T0)*(L0 + L1*cos(T1) + L2*cos(T1-T2) + L3);
Z = -cos(T0)*(L0 + L1*cos(T1) + L2*cos(T1-T2) + L3);
```

Inverse Position Kinematics

```
%Given the required Position and orientation of the 1st joint
Z_dash = -Z/cosd(T0) - L0 - L3;

P = (X^2) + (Z_dash^2) + (L1^2) - (L2^2);
R = 2*X*L1;
Q = 2*Z_dash*L1;

a = 2*P*Q;
b = (P^2) - (R^2);
c = 2*((R^2) + (Q^2));

C1 = (a ± sqrt((a^2) - 2*b*c))/c;
S1 = ± sqrt(1 - (C1^2));

T1 = atan2(S1, C1)
```

```
%Given the required Position and orientation of the 1st joint
Z_dash = -Z/cosd(T0) - L0 - L3;

C2 = ((X^2) + (Z_dash^2) - (L1^2) - (L2^2)) / (2*L1*L2);
S2 = ± sqrt(1 - (C2^2));

T2 = atan2(S2,C2);
```

Direct Velocity Kinematics

```
%Give the Required Angles & Angular Velocities.
%Pre-computed variables
S0 = sin(T0);
C0 = cos(T0);

S1 = L1*sin(T1);
C1 = L1*cos(T1);

S2 = L2*sin(T1-T2);
C2 = L2*cos(T1-T2);

%Jacobian Matrix
J = [             0,                  C1+C2,            -C2;
     -C0*(L0+C1+C2+L3),    S0*(S1+S2),       -S0*S2;
      S0*(L0+C1+C2+L3),    C0*(S1+S2),       -C0*S2];

%Result the Kinematics X_dot = J x Theta_dot
Velocity = J * [T0_dot; T1_dot; T2_dot];
```