# Soccer playing robots

### By

{Faisal Jehangir}

{01-133102-027}

### Supervised by

{Nadia Imran}



{2010-2015}

A Report is submitted to the Department of Electrical Engineering,

Bahria University, Islamabad.

In partial fulfillment of requirement for the degree of BS (EE)

# Certificate

We accept the work contained in this report as a confirmation to the required standard for the partial fulfillment of the degree of BS (EE).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of Department Supervisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner External Examiner

# Dedication

I dedicate this project to my both supervisors and committee members of the control systems.

# Acknowledgements

##### I am very thankful to my supervisors, Dr.Salman Ahmed and Nadia Imran for their support and guidance that they gave along the way. I am also thankful to all of the committee members of control systems who took their valuable time in guiding me and sharing valuable information to achieve and improve my project in many aspects.

# Abstract

##### The main idea of this thesis was to make a software based controller that can allow two robots to play soccer against each other. This thesis is about making semi-automatic robots as bringing automation to machines is one of the most important aspect of modern technology.

##### Different portions made for the project included image processing coordination of multiple robots and trajectory tracking.

##### The thesis is based on study, implementation and changes in the above mentioned portions to enhance the outcome of the soccer playing robots. Image processing is used for the detection of objects on stadium field that includes robots, ball and the goals. Serial communication is done between robots and computer by Bluetooth modules. The controller made to control the robots is software based and communicates by serial interfacing done between the computer and robots.

Contents

[Certificate i](#_Toc418767165)

[Dedication ii](#_Toc418767166)

[Acknowledgements iii](#_Toc418767167)

[Abstract iv](#_Toc418767168)

[List of Figures viii](#_Toc418767169)

[List of Tables x](#_Toc418767170)

[Chapter # 1 1](#_Toc418767171)

[Introduction 1](#_Toc418767172)

[1.1 Project Overview 2](#_Toc418767173)

[1.2 Problem Description 2](#_Toc418767174)

[1.3 Project Objectives 2](#_Toc418767175)

[1.4 Project Scope 3](#_Toc418767176)

[Chapter # 2 4](#_Toc418767177)

[Literature Review 4](#_Toc418767178)

[2.1 RoboCup Background 5](#_Toc418767179)

[2.2 Small Size League (SSL) RoboCup Framework 5](#_Toc418767180)

[2.2.1 SSL-Vision 6](#_Toc418767181)

[2.2.2 Framework for SSL-Vision 7](#_Toc418767182)

[2.3 MATLAB 8](#_Toc418767183)

[2.4 Trajectory Tracking 8](#_Toc418767184)

[2.4.1 Control of a unicycle type robot 8](#_Toc418767185)

[Chapter # 3 10](#_Toc418767186)

[Requirement Specifications 10](#_Toc418767187)

[3.1 - Functional requirements 11](#_Toc418767188)

[3.1.1 – Capture from webcam 11](#_Toc418767189)

[3.1.2 – Process the captured frame 11](#_Toc418767190)

[3.1.3 – Conversion to binary image 11](#_Toc418767191)

[3.1.4 – Calculate frame dimensions 11](#_Toc418767192)

[3.1.5 – Calculate Region of interests (ROI) 11](#_Toc418767193)

[3.1.6 – Arranging the Region of interests (ROI) 11](#_Toc418767194)

[3.1.7 – Comparing the findings 11](#_Toc418767195)

[3.1.8- Sending the commands 11](#_Toc418767196)

[3.2 - Non functional requirements/Quality Requirements 12](#_Toc418767197)

[3.2.1 Speed 12](#_Toc418767198)

[3.2.2 Efficiency 12](#_Toc418767199)

[3.2.3 Reliability 12](#_Toc418767200)

[3.2.4 Legal and licensing 12](#_Toc418767201)

[Chapter # 4 13](#_Toc418767202)

[System Design 13](#_Toc418767203)

[4.1 Design Methodology 14](#_Toc418767204)

[4.1.1 Vision 14](#_Toc418767205)

[4.1.2 Image processing 14](#_Toc418767206)

[4.1.3 Localization 14](#_Toc418767207)

[4.1.4 Controller 15](#_Toc418767208)

[4.1.5 Behavior 15](#_Toc418767209)

[Chapter # 5 16](#_Toc418767210)

[System Implementation 16](#_Toc418767211)

[o Vision system 17](#_Toc418767212)

[o Controller 17](#_Toc418767213)

[o Robots 17](#_Toc418767214)

[5.1 Vision System 17](#_Toc418767215)

[5.1.1 Camera 17](#_Toc418767216)

[5.1.2 Computer 18](#_Toc418767217)

[5.1.3 MATLAB 18](#_Toc418767218)

[5.1.4 Bluetooth Transceiver 19](#_Toc418767219)

[5.2 Controller 19](#_Toc418767220)

[5.2.1 Checking for ball position 19](#_Toc418767221)

[5.2.2 Angle Comparison 19](#_Toc418767222)

[5.2.3 Hitting the ball 19](#_Toc418767223)

[5.2.4 Defending 20](#_Toc418767224)

[5.3 Robot 20](#_Toc418767225)

[5.3.1 Bluetooth Transceiver 21](#_Toc418767226)

[5.3.2 Arduino UNO R3 (Microcontroller) 21](#_Toc418767227)

[5.3.3 Motor Driver IC 22](#_Toc418767228)

[5.3.4 Motors 22](#_Toc418767229)

[5.3.5 Security 23](#_Toc418767230)

[Chapter # 6 24](#_Toc418767231)

[System Testing and Evaluation 24](#_Toc418767232)

[6.1 Graphical user interface testing 25](#_Toc418767233)

[6.2 Usability testing 25](#_Toc418767234)

[6.3 Software performance testing 25](#_Toc418767235)

[6.4 Compatibility testing 25](#_Toc418767236)

[6.5 Exception handling 25](#_Toc418767237)

[6.6 Hardware testing 26](#_Toc418767238)

[6.7 Security testing 26](#_Toc418767239)

[6.8 Limitations 26](#_Toc418767240)

[Chapter # 7 27](#_Toc418767241)

[Conclusion 27](#_Toc418767242)

[7.1 Knowledge acquired by work done in this project 28](#_Toc418767243)

[7.1.1 Image processing 28](#_Toc418767244)

[7.1.2 Wireless communication between machines 28](#_Toc418767245)

[7.2 Improvements 28](#_Toc418767246)

[References 29](#_Toc418767247)

Appendix-A…………………………………………………………………………………………………………………………31

[User Manual 32](#_Toc418767248)

## List of Figures

[Figure 2.1: Diagram of Aldebaran Nao [Taken from [1]] 5](#_Toc418747619)

[Figure 2.2: Small size league robot [Taken from [1]] 6](#_Toc418747620)

[Figure 2.3: Small size league robot [Taken from [3]] 7](#_Toc418747621)

[Figure 2.4: Results of a robot following straight line trajectory [Taken from[2]] 9](#_Toc418747622)

[Figure 2.5: Results of a robot following semi-circle trajectory [Taken from [2]] 9](#_Toc418747623)

[Figure 4.1: Methodology of the project 14](#_Toc418747624)

[Figure 5.1: Vision system components 17](#_Toc418747625)

[Figure 5.2: Imaginary point placement 20](#_Toc418747626)

[Figure 5.3: Components of robot 20](#_Toc418747627)

## List of Tables

[Table 5.1: Camera Specifitaions …………………………………………………………………………………………17](#_Toc418749051)

[Table 5.2: Specifications of computer used for thesis ………………………………………………..……18](#_Toc418749052)

[Table 5.3: Specification of USB Bluetooth module ……………………………………………………………19](#_Toc418749053)

[Table 5.4: Specification of HC-05 Bluetooth module………………………………………………………… 21](#_Toc418749054)

[Table 5.5: Specification of Arduino UNO R3 micro-controller ……………………………………………21](#_Toc418749055)

[Table 5.6: Specification of L293D motor driver IC ………………………………………………………………22](#_Toc418749056)

[Table 5.7: Specification of DC motors used in robots ……………………………………………………….22](#_Toc418749057)

# Chapter # 1

# Introduction

## 1.1 Project Overview

##### Autonomous robots are designed to do certain task on the basis of algorithm embedded in them and they are mostly limited to a described field. These robots need a system to judge their surroundings, their position in that environment, a way to perform the assigned task and how to do that precisely. When multiple robots are present the complication increases greatly as one robot has to be designed intelligent enough to perform the task without effecting the other.

##### The small size league (SSL) robots are a part of soccer playing robots team. The robots in this team are guided by software in all aspects like their teammates, opponents, position of goals and the ball. Both the teams have exact same hardware as for this competition hardware is standardized. So basically the competition is all about the software designed to control a team. This software consists of all the basics modules required to determine the visual aspects as well as the algorithm to tell a robot what to do and how to do. The main focus of this thesis was on image processing and trajectory tracking. [1] [8]

##### The main idea is motivated from the SSL robots but the structure is different, like this thesis does not comprise of a team rather, it consists of two opponent robots that will play soccer against each other. The design of robot, its dimension, way of movement and stadium dimensions are also different. The basic steps involved in the operation of project includes capturing of image through a camera then that image is converted to more suitable form that the software can handle like Grayscale image or standard RGB(Image made by the combination of standard colors only image that are red, green, blue). Image processing is done to determine position of the robots and ball. The controller designed gets the appropriate information from the image and determines what the robot should do next to score a goal.

## 1.2 Problem Description

##### The small size league (SSL) uses quite complex concepts for soccer playing robots teams and if one wants to make a robot for small size league (SSL) it is very difficult especially for beginners. So this project can serve as a guideline to all the basics concepts that can help to make a SSL robot.

#####

##### The main idea of this thesis was to make a software based controller that can allow two robots to play soccer against each other using image processing for object tracking and serial interface for communication.

## 1.3 Project Objectives

##### Image processing for orientation and position of robots and ball

##### Design a software based controller to make two robots play soccer against each other

## 1.4 Project Scope

##### This thesis shows one of the ways to bring automation to a robot by designing a software based controller. As bringing automation to machines is one of the most important aspects of modern technology. This thesis is different in most of the ways from the standards of small sized league robots (SSL). However the algorithm used in this project is same as used in SSL. This project depicts the most basic things and study required to develop robot for SSL and it can serve as a basic guideline to someone who is interested in developing SSL team.

# Chapter # 2

# Literature Review

## 2.1 RoboCup Background

##### RoboCup is a scientific initiative with a sole purpose to increase research and students interest in various technological fields by providing a standard platform like the robot soccer games. The idea behind this concept was to make robots intelligent enough by designing complex controller to have a humanoid robot team that can play against human soccer players under official FIFA rules.

##### RoboCup has four main interests: RoboCup Soccer, RoboCup Rescue, RoboCup home, and RoboCup Junior. RoboCup Soccer has 5 leagues: Humanoid, Middle Size, Simulation, Small Size, and Standard Platform. The WPI Warriors team first competed in the Standard Platform League in 2011. This league has fixed hardware, so only the software can be changed.

##### The RoboCup most advanced league uses the Aldebaran Nao robots as shown as shown in Fig 2.1 Project Nao launched in 2004 and the Nao robots have been the robot of the SPL since 2010. The most recent release of the heads for the Nao robots is version 4, which features 2 high-definition cameras (1280x960), an Intel ATOM 1.6 GHz CPU, and wireless communications over Wi-Fi. The body of the robots features 21 degrees of freedom [1]


Figure 2.1: Diagram of Aldebaran Nao [Taken from [1]]

## 2.2 Small Size League (SSL) RoboCup Framework

##### One of the categories in RoboCup is SSL i.e. Small size league robots category. It consist of smaller size robots teams. Each team consists of six robots. Hardware of both the teams is exactly same the only thing that is different is the software or the algorithm used to control these robots. Fig 2.2


Figure 2.2: Small size league robot [Taken from [1]]

##### The camera feed is given to both teams owners. SSL vision software is used to get required information from the real time captured frames. After processing the image the orientation and position information is given to the software based controller. This controller then tells the robots what to do and how to do it. [5]

### 2.2.1 SSL-Vision

##### It is the software and hardware setup required to find the ball and robots position and orientation in the field. Previously Small Size League rules allow each team has their own global vision system. The progress made by individual participating teams was pretty close to each other and only minor difference or upgrading were present. Hence the responsible committees decided to migrate to a shared vision system (including also sharing the vision hardware) for all teams by 2010. This system named SSL-Vision is currently developed by volunteers from participating teams. [4]

##### Other major problems with individual vision setup were that the setup time required before as well as during the competition was too much, as having five teams playing on a field; ten cameras need to be mounted and calibrated. During these preparations, a field cannot be used for any matches or other preparations.

##### Due to the standardized field size, SSL-Vision becomes an ideal solution for the Humanoid as well as the Standard Platform League.

### 2.2.2 Framework for SSL-Vision

##### The figure 2.3 shows an overview of the framework architecture. The entire system's desired processing flow is encoded in a multi-camera stack which fully defines how many cameras are used for capturing, and what particular processing should be performed. The system has been designed so that developers can create different stacks for different robotics application scenarios. By default, the system will load a particular multi-camera stack, labeled as “RoboCup Small Size Dual Camera Stack”.[3]

##### The software includes all the stuff from camera calibration to the pattern recognition for individual robots of each team. And updates the information provided to each team on real time basis.

##### The real-time progress is achieved by using plug-in to interconnect the software with a good video card. Currently SSL-vision uses Nvidia Geforce 7800GTX videos card.

****

Figure 2.3: Small size league robot [Taken from [3]]

##### The SSL-vision uses extended Kalman filter for localization purposes. Robot localizations is quite difficult task when the field of view is limited and with limiting processing capabilities. The two most commonly used strategies for this purpose are Kalman filtering and particle filtering.

## 2.3 MATLAB

##### MATLAB is commercial "Matrix Laboratory" software that provides a user friendly environment for interactive programming. MATLAB provides a vast variety of built-in functions including almost all of the mathematical expressions and visualization like graphs and image processing. MATLAB provides matrix computations, signal processing, numerical analysis and graphical environment with easy built-in functions and thus reduces the requirement of traditional programming. [6]

##### In my thesis MATLAB is used basically for image processing and controller designing. MATLAB gives all the image processing features like image enhancement(sharpening or un-blurring an out of focus image, highlighting edges, improving image contrast, or brightening an image, removing noise), image restoration (removing of blur caused by linear motion, removal of optical distortions) and image segmentation (finding lines, circles, or particular shapes in an image, in an aerial photograph, identifying cars, trees, buildings, or roads etc.)

## 2.4 Trajectory Tracking

##### Tracking means following a defined path. For different wheel vehicles different equations are derived to make them follow the path keeping in mind the slippage of wheels and precision of movement. For unicycle type robot with two actuated wheels on a common axle and a midpoint “M" between the two wheels the equations is as follows

 [2]

##### Here v and w are the robot translation and angular velocity of the robot respectively and the theta denotes the vehicle's angle in the field with respect to a fixed co-ordinate system.

### 2.4.1 Control of a unicycle type robot

##### The equation of the controller for the unicycle-type robot in the “Trajectory tracking for unicycle-type and two-steering-wheels mobile robots” by “Alain Mi-caelli and Claude Samson” [2]

****[2]

##### The following are the results of the controller formed for two reference trajectories Fig 2.4 is a straight line trajectory tracking result of a robot and Fig 2.5 is of a semi-circle trajectory simulation result.

****

Figure 2.4: Results of a robot following straight line trajectory [Taken from [2]]

****

Figure 2.5: Results of a robot following semi-circle trajectory [Taken from [2]]

##### The deviation in these results is due to the slippage of wheels and the minute difference between the rpm of the two driving wheel motors as two motors cannot be hundred percent identical.

# Chapter # 3

# Requirement Specifications

## 3.1 - Functional requirements

### 3.1.1 – Capture from webcam

##### The software will interact with the webcam attached and capture a frame,

### 3.1.2 – Process the captured frame

Once the frame is captured it will be processed in the following way

### 3.1.3 – Conversion to binary image

##### The captured frame is converted into binary frame for easy processing and reducing the load on the system.

### 3.1.4 – Calculate frame dimensions

##### The captured frame’s dimensions are acquired for calibration.

### 3.1.5 – Calculate Region of interests (ROI)

##### Once the frame is ready and filtered the regions of interest are extracted that are basically the detected objects on the stadium field.

### 3.1.6 – Arranging the Region of interests (ROI)

##### The detected regions are then arranged according to their respective sizes so they can be sequenced and respective information can be acquired. Each object size represents a specific ROI like the biggest detected ROI is for the goal position.

### 3.1.7 – Comparing the findings

##### The controller then looks for the conditions of movement. Like it compare the angle between the robot and the ball. If the angle is greater or less then certain amount the robot rotates to face the ball.

### 3.1.8- Sending the commands

##### After the conditions are checked, if the condition is not satisfied like the robot is facing not towards the ball the controller will send rotation command to rotate the robot so that it faces the ball. On the other hand if the condition is satisfied the controller checks for the next condition.

## 3.2 - Non functional requirements/Quality Requirements

### 3.2.1 Speed

##### The software controller and the image processing algorithm is fast and can process four frames per second and it can perform better if a better computer is used.

### 3.2.2 Efficiency

##### The motion of the robots is precise and accurate by reducing the slippage and better design model.

### 3.2.3 Reliability

##### The project is reliable and secure as each wireless communicating modules is secured by a unique password. Proper circuits are designed and components are selected after doing calculation so that nothing is loaded then more it can handle.

### 3.2.4 Legal and [licensing](http://en.wikipedia.org/wiki/Software_license_agreement)

##### The software base controller has no legal or licensing issues, as I am using open source software like Arduino 1.0.5 and MATLAB 2014a.

# Chapter # 4

# System Design

## ****4.1 Design Methodology****

##### The Fig 4.1 is the methodology on which the project operates



Figure 4.1: Methodology of the project

##### For vision a web-cam is used to acquire frames. Image processing is done on MATLAB. From the processed image information like the position of goals, ball and angle and position of robot is determined. The information acquired then is used to feed the controller for motion of the robot.

### ****4.1.1 Vision****

##### **A webcam is selected to get the live streaming of the stadium field. The reason for selecting a webcam is that the image quality is good enough to get the information from the frames being captured and small resolution reduces the processing load as well.**

#####

### ****4.1.2 Image processing****

##### **Each captured frame is processed. The processing includes the conversion to binary images, splitting the frame into channels to get better results. Each captured frame has slight variation of threshold of the detected objects for the red, green and blue channel. Once the objects are detected in each channel the channels are merged together to get the region that is common in all.**

### ****4.1.3 Localization****

##### **The detected objects are then labeled and their values like the area they occupy and centre points are stored in matrices so they can be used by the controller.**

### ****4.1.4 Controller****

##### **Each robot has its own software based controller that is programmed in MATLAB software.**

##### **The controller’s main job is to check for the conditions and decide what it should do next to make robot approach the ball and score the goal. Or in the other case it if the other robot is trying to score a goal how can it stop it.**

### ****4.1.5 Behavior****

##### **After the controller checks for conditions and decides what to do it send motion commands to the robots and the robot’s job is to act on the commands being send. Including moving forward, backward and rotation in either directions.**

#####

#

# Chapter # 5

#

# System Implementation

##### **To implement the project it’s been categorized in the following three main parts.**

## Vision system

## Controller

## Robots

## ****5.1 Vision System****

##### The diagram 5.1 shows the data flow of the vision system



Figure 5.1: Vision system components

### ****5.1.1 Camera****

##### I am using Lenovo q350 USB web-cam for image capturing. It has resolution of 320x240.The smaller resolution makes the image less detailed hence the image processing can be done at a better speed. It has a USB interface and is compatible with every operating system.

|  |  |
| --- | --- |
| Resolution | 320x240 |
| Operating voltage | 4V-6V |
| Capturing speed | 25 fps |

#### Table 5.1: Camera Specifitaions

### 5.1.2 Computer

##### Computer is used for image processing and data communication between robot and the software based controller. Any computer can be used preferably with 1GB of RAM and a dual core processor as image processing requires resources like RAM and Video memory. More memory helps in having a bigger buffer that can help in speeding up the image processing process after capturing the frames from the attached camera.

|  |  |
| --- | --- |
| Processor | Core 2 Duo E8400 |
| RAM | 2 GB |
| Chip-set | Intel G41 series |

#### Table 5.2: Specifications of computer used for thesis

### 5.1.3 MATLAB

##### MATLAB is used for image processing and designing the controller for the robot motion. MATLAB Image acquisition tool is used to link the camera with the MATLAB software. This tool has built-in functions for processing an image and acquiring required information from either an image or an array of images or even live stream.

##### In this thesis I used MATLAB for tracking the objects. In my case there are two robots, ball and two goals which are the main objects to be tracked. Ball tracking just requires position but for a robot its angle is as important as its position. Without angle a robot cannot be moved to a desired location, the object will not be in front of it all the time so it had to move on certain angle so that it faces its target and then move straight towards the target. As displacement is the shortest path to the target. For tracking an object in real time streaming, each frame is processed individually to get region of interest. The acquired frame is then broken down into RGB channels. From these channels the region of interest is obtained by subtraction or using logical operations like "OR" operation and "AND" operation. Dividing the frame into channels helps in getting more precise results as different channel have a little different threshold of different colors. [7] [9]

##### After the region of interest is highlighted in each channel they are joined together by "AND" operation to get the common region in all of the respective channels. Filtering can be done after this step to avoid noise. The frame is converted into binary for blob analysis. Blob analysis gives the region in a binary image that has different properties with respect to other image like a white spot on a black colored area. Blob analysis in MATLAB also provides several functions like bounding box (Draws a box around the region of interest in an image), Centroid (Gives the center point of the region of interest in an image), Labeling (to label the region of interest in an image) and blob count (number of regions of interest in an image). After the frame's information has been extracted it is given to the controller that decides what should be done next for robots motion. All these steps are repeated for individual frames that are acquired from the camera. [8]

##### The controllers main job is to direct the robot towards the balls. The secondary objectives of controller are to avoid collisions, direct the robot to hit ball in the direction of the goal and take the shortest path to reach the ball. The controller designed uses computer's serial port to send the data via USB Bluetooth to the robot.

### 5.1.4 Bluetooth Transceiver

##### After the image is processed and required information is given to the controller, it generates an output to be sent to the robot for movement. Different communication modules can be used here like Radio transceivers, IR transceivers and Bluetooth transceivers. I am using Bluetooth transceivers for their simplicity and lower power consumption as compared to the other two modules. Bluetooth transceivers are plug and play devices so easy to install and are ideal for smaller distances. The device I used is 2.0 Bluetooth USB dongle.

|  |  |
| --- | --- |
| Interface | USB 2.0 |
| Support system | Windows 98/98SE/ME/2000/XP/Vista |
| Symbol rate | Symbol rate |
| Receiving and sending range | Up to 20 m |
| Support Bluetooth | V 2.0 and V 1.2 |

#### Table 5.3: Specification of USB Bluetooth module

## 5.2 Controller

##### The controller checks for several conditions before sending a movement command to the robot.

### 5.2.1 Checking for ball position

##### The controller first job is to heck for the ball position that whether it lies in front of it or behind it. If it lies behind it the robot alligins it self with x axis and starts reversing until the ball is in front of it. If its already in fron the controller jumps to the angle comparison algorithm.

### 5.2.2 Angle Comparison

##### Instead of trying to approach ball the robot tries to get to a point that is at a specific distance behind the ball. It first faces the ball by comparing the angle between ball and robot’s head to ball and robot’s base. If that difference is greater then +\_ 5 degrees it rotates the robot left or right to face that imaginary point. Once it faces that point it starts moving towards it.

### 5.2.3 Hitting the ball

##### After reaching the imaginary point the controller alligns the robot in direction of goal and hit the ball in direction of goal with some momentum. The figure 4.2 shows the imaginary point placement. The point ‘G’ represents the postion of goal on the stadium, the point ‘B’ represents the current position of ball and ‘I’ represents the imaginary poit defined by the controller.



Figure 5.2: Imaginary point placement

### 5.2.4 Defending

##### The controller also checks for the condition like if one robot is too close to the ball its obvious that it will reach the ball first. So instead of trying to approach it the controller makes the robot defend the goal by making robot come in between ball and goal.

### 5.3 Robot

##### The figure 5.3 shows the data flow in the robot



Figure 5.3: Components of robot

### 5.3.1 Bluetooth Transceiver

##### The robot uses HC-05 module for getting instruction from the software based controller present in the computer. HC-05 is the most widely used transceiver having both master and salve configuration options. The module has built-in memory for information storage and provides a number of configuration options. The options include changing of module display name, password, master slave configuration and baud rate settings. [14]

|  |  |
| --- | --- |
| Mini Size(L x W x H) | Approx. 27 x 13 x 2mm |
| Operating Frequency Band | 2.4GHz |
| Bluetooth Specification | V2.0+EDR |
| Output Power Class | Output Power Class |
| Operating Voltage | 3.3V |
| Flash Memory Size | 8Mbit Storage |
| Temperature | −40◦C to 80◦C |
| Working Temperature | −25◦C to 75◦C |

#### Table 5.4: Specification of HC-05 Bluetooth module

### 5.3.2 Arduino UNO R3 (Microcontroller)

##### In this project the micro-controller is used as a medium for converting the bits received from Bluetooth module of robot into information that a motor driver IC can understand. Arduino UNO R3 uses ATmega328 micro-controller that is ideal for a scenario like this project with enough speed and error free translation. [15]

|  |  |
| --- | --- |
| Micro-controller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage(recommended) | 7V-12V |
| Input Voltage(limits) | 6V-20V |
| Digital I/O Pins | 14 (6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | DC Current for 3.3V Pin |
| Flash Memory | 32 KB 0.5 KB used by boot loader |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Clock Speed | 16 MHz |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |

#### Table 5.5: Specification of Arduino UNO R3 micro-controller

### 5.3.3 Motor Driver IC

##### L293D is a dual H-bridge IC used for controlling DC motors. Dual H-bridge makes it capable to control two DC motors simultaneously. Smaller size, low power consumption for IC operation i.e. 0.120 watt, less heat and separate rails for IC operation and motor driving make it ideal for small size robots

|  |  |
| --- | --- |
| Product | DC Motor Controllers / Drivers |
| Type | H-Bridge |
| Operating Supply Voltage | 4.5V to 36V |
| Output Current | 600 mA |
| Operating Temperature | −40◦C to +150◦C |
| Supply Current | 2 mA |
| Mounting Style | Through Holes |
| Package/Case | PowerDIP-16 |
| Number of Outputs | 2 |

#### Table 5.6: Specification of L293D motor driver IC

### 5.3.4 Motors

##### The robot uses 4 DC brushed motors for movement. Each motor has a gearbox for better torque and lower speed. Lower speed gives precise movement in the stadium field and high torque reduces motor load as the robot weighs 2 kg. The motors are used in pairs for movement. If right pair of motors turn clockwise and left pair of motors turns anti clockwise the robot turns left, similarly if right pair of motors turn anti clockwise and left pair of motors turns clockwise the robot turns right. To move forward both pair of motors turn clockwise and anticlockwise for backward motion respectively. All this motion control is achieved by the motor driver IC.

|  |  |
| --- | --- |
| Operating Temperature | −10◦C to −60◦C |
| Rated Voltage | 6.0VDC to 12.0VDC |
| Rated Load | 10 g\*cm |
| No-load Current | 70 mA max |
| No-load Speed | Speed 9100 +-1800 rpm |
| Loaded Current | 250 mA max |
| Loaded Speed | 4500 +-1500 rpm |
| Starting Torque | 20 g\*cm |
| Starting Voltage | 2 VDC |
| Stall Current | 500 mA max |
| Max Body Size | 27.5 mm x 20 mm x 15 mm |
| Shaft Size | 8 mm x 2 mm |

#### Table 5.7: Specification of DC motors used in robots

### 5.3.5 Security

##### The robots communicate with the computer by the Bluetooth modules. Their default passwords are ‘1234’ so if anyone knows the default passwords he can connect with the robots. So the default passwords of both of the module present on the robots have been changed to avoid misuse.

# Chapter # 6

# System Testing and Evaluation

## 6.1 [Graphical user interface testing](http://en.wikipedia.org/wiki/Graphical_user_interface_testing)

##### As this project is not a prototype of something that has to be presented in the market, it doesn’t have a GUI made for the software. The sole purpose of this project was to implement what was learnt during the bachelor degree program and to learn more about semi-autonomous robotics and image processing.

## 6.2 [Usability testing](http://en.wikipedia.org/wiki/Usability_testing)

##### Although the software lacks graphical user interface its quite simple to use. The user will require MATLAB version 2014a and the “m” file built for this project having the vision system and the controller. So the person who wants to see working of this project just needs to have the basic knowledge about the MATLAB software.

##### The hardware part is even simpler as the user just need to switch the power button present on the robots.

## 6.3 [Software performance testing](http://en.wikipedia.org/wiki/Software_performance_testing)

##### MATLAB is not made specifically for image processing although the image processing tool in MATLAB has almost all the functions required to process an image and acquire useful information from the captured image. Image processing in MATLAB is not as fast as it can be in software solely made for this purpose. Keeping this thing in mind still the controller loop takes only 0.5seconds to 0.8 seconds to process a single frame that is adequate. As the knowledge about image processing was in learning phase so still there might be couple of things that can make the software performance even better.

## 6.4 [Compatibility testing](http://en.wikipedia.org/wiki/Compatibility_testing)

##### As far as compatibility is concerned it all depends upon the system the software is being run at. The system use for testing of the project was having a dual core processor and 2GB or RAM. So if the system is faster the software will perform better and if it’s slower the performance of the software will decrease also.

##### In hardware the robots require quality batteries and having voltage level above 8v. Although the Arduino can work on lower voltages fine but as the batteries fail to provide enough power to run the motors and do serial communication at same time, the connection of the Bluetooth modules drop. So 8V to 12V batteries are recommend that are charged above 20 percent for flawless operation of the project.

## 6.5 [Exception handling](http://en.wikipedia.org/wiki/Exception_handling)

##### The default baud rate of the hc-05 and hc-06 devices is 9600 bits per second; it’s enough if the data send serially is at regular intervals or requires fewer rates. For my project the commands from the controller are not sent at exact same interval as some frames require less time in processing and some require more. So the default baud rate was resulting in connection drop. To overcome this problem default baud rate were set to 115200bits per second and to change the default security pins of the Bluetooth modules they were modified.

The speed of the robots’ motion was also improved by sending twice motion command per frame processed instead of sending a single motion command.

## 6.6 [Hardware testing](http://en.wikipedia.org/wiki/Load_testing)

##### The motor driver IC L-293D can handle a maximum current of 650mA per channel with 2 300mA current rated brushed motors attached to each channel the current reaches 600mA. Although it lies within specs but it is quite close to maximum limits and heats the IC L-293D.

## [6.7 Security testing](http://en.wikipedia.org/wiki/Security_testing)

##### The only components that require security are the Bluetooth modules. This is the reason that their default security pins were changed. But still if someone knows the key then during the operation of the project the robots will be unlinked from the computer and connect to the device that attempted for the connection known as paring in case of Bluetooth devices.

## 6.8 Limitations

##### During operation of project no white or red colored objects should be placed on the stadium field. The extra objects detected will slow down the image processing process and that object might be considered as part of the robots, ball or the goal position resulting in false movements of robots.

##### The robots battery should be charged above 20% to avoid connection drop during operation between paired Bluetooth modules.

##### The robots cannot hit the ball if it’s lying too close to the stadium borders as in that case the imaginary point defined by the controller might be lying outside the stadium. So the ball should be moved in manually.

# Chapter # 7

# Conclusion

## 7.1 Knowledge acquired by work done in this project

### 7.1.1 Image processing

##### Image processing proves to be useful in many ways like detecting objects, comparing objects sizes and getting almost all the information about an object including its size, pixels, centre point, color, length etc. Moreover with even a low cost webcam the results of image processing are quite acceptable and accurate so it’s an in expensive yet very powerful tool to have.

### 7.1.2 Wireless communication between machines

##### Wireless communication on the other hand is little expensive if used for short distances. But for machine like a robot that has to rotate and move back and forth it the best way of communication as it cannot be done using wires. The distance to what two wireless modules can communicate depends on its quality and quality is directly proportional to cost. I learnt how to do serial wireless communication and also about how much important is the security of wireless communication.

## 7.2 Improvements

##### Following are the things that can improve the outcome of the soccer playing robots.

##### The wireless range of communication between computer and robots can be improved if high quality Bluetooth modules are used or radio communication can be done in case of big soccer field.

##### More robots can be added to make the game more interesting.

##### Brushless motors can be used in robots for reliability.

##### Algorithm used for vision system as well as controller may be improved for better performance

# References

##### F. Clinckemaillie, D. Kent, W. Mulligan, R. O'Meara, and Q. Palmer. AU- TONOMOUS MULTI-ROBOT SOCCER, pages 1to5, 2012.

##### A. Micaelli and C. Samson. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. In Robotique,image et vision, pages 9to27, 1993.

##### Vision System for the RoboCup Small Size League, 2009.

##### RoboCup Small Size League: SSL Web Site. http://small-size.informatik.uni-bremen.de (2009)

##### SSL-Vision Developer Team: RoboCup Small Size League Shared Vision System Project Home. http://code.google.com/p/ssl-vision/ (2009)

##### Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins. Digital Image Processing Using MATLAB® Second Edition, 2009

##### Raquib Buksh1, Soumyajit Routh2, Parthib Mitra3, Subhajit Banik4, Abhishek Mallik5, Sauvik Das Gupta6. Implementation of MATLAB based object detection technique on Arduino Board and iROBOT CREATE, pages 3to5 2014.

##### Hrvoje Turic, Vladimir Plestina, Vladan Papic and Ante Krolo. Robot Soccer Educational Courses, Robot Soccer, Vladan Papi (Ed), 2010.

##### V. Subburaman and S. Marcel. Fast Bounding Box estimation based face detection in ‘Workshop on Face Detection of the European Conference on Computer Vision (ECCV)’, 2010.

1. <http://nf.nci.org.au/facilities/software/Matlab/toolbox/images/regionprops.html>
2. [www.mathworks.com](http://www.mathworks.com). Regoinprops<http://www.mathworks.in/help/images/ref/regionprops.html>
3. Zickler, S.: The VarTypes System. http://code.google.com/p/vartypes/ 4. Bruce, J. Balch,
4. Veloso, M.: Fast and inexpensive color image segmentation for interactive robots. In: Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’00). Volume 3. (2000) 2061–2066
5. Stan. How to connect an Arduino Uno to an Android phone via Bluetooth <http://42bots.com/tutorials/how-to-connect-arduino-uno-to-android-phone-via-bluetooth/>
6. Hazim Bitar. Modify The HC-05 Bluetooth Module Defaults Using AT Commands. <http://www.techbitar.com/modify-the-hc-05-bluetooth-module-defaults-using-at-commands.html>

**Appendix-A**

## User Manual

##### Place the robots and ball in the stadium field.

1. Make sure that the robot’s batteries are charged and then switch on the robots
2. Connect the webcams and USB Bluetooth device with the computer.
3. Make sure that the robots are already paired once using Bluetooth connection with the computer before running the code.
4. Open both of the MATLAB “m” files containing the vision system and controller algorithm in separate MATLAB windows. By default the “m” files are linked to MATLAB software and double clicking the file opens it, but if you want to open it from the software select the home tab and click on the open button and then select the file you want to open.



Click on the Open button and select the project file

Select the HOME tab

1. If same hardware is used then there is no need to configure anything just hit the “RUN” button on the “EDITOR” tab to start the program.



Select the EDITOR tab

Click on the Run button to start the program

##### If different webcams are used then change default the name of webcam to the name of webcam attached to the computer.

 

Change the default name of webcams in code line # 2

##### The program Run for 10 minutes after that it stops as the duration of one match is set to 10 minutes. To start a new match simply click on the “Run” button as mentioned in the step “5”.