Software Requirement Specifications

Embedded Systems Lab

CS 308 Project

MINI GOLF ROBOT

Neeraj Toshniwal (09005039)

Vaibhav Gupta (09005019)

Arup Kumar Pal (09005024)

Shrikant Nagori (09005040)

Table of Contents: **Page No.**

Introduction:

The purpose of this document is to present a detailed description of the Mini Golf Robot. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for the developers of the system.

Definitions, Acronyms and Abbreviations:

Terms	*Description*
Bot / Robot	The electromechanical machine that will operate according to our program.
Software requirement specifications	The document all the functionalities of our robot and the environment along with all the constraints under which it will work.
Mechanical Arm	The robot is equipped with an indigenously developed mechanical hand that it uses to provide a desired impulse.
IDE	Integrated Development Environment.
SDK	Software Development Kit.
VGA	Video Graphics Accelerator.
API	Application Program Interface.
Visual Studios	IDE for the SDK used.
AVR Studios	IDE for 'C' programming that will run the bot.
DirectX	DirectX is a collection of API for handling multimedia related tasks, especially game programming, on Microsoft platforms.
IR	Infra Red

Aim of the Project:

The aim of the project is to design a golf playing robot.

Given a hole (marked by a flag) and an orange coloured ping pong ball (instead of the heavy original golf ball) on a flat surface, our robot will try to put the ball into the hole.

Environment for the Mini Golf:

***1**. Our golf court is a plane surface.*

The bot doesn't have enough degrees of freedom in its movement and constructing a mechanism to take into account the undulations of the surface will be very challenging. We are focusing mainly at the game completion part rather than the intricacies of details in the robot's movement.

***2**. The golf balls are simulated using light weight orange coloured ping pong balls.*

The purpose of using light weight balls is to reduce the force that has to be exerted to move them. Using lightweight balls helps in minimizing power requirements and also reducing the workload of the mechanical arm without sacrificing much on the simulation of the game.

***3**. There shouldn't be any other objects of colour similar to that of the golf balls.*

The golf court setting is such that the background as perceived by the camera doesn't fail to capture the orange coloured balls.

***4**. There is a hole on the plane which is marked by placing a thin white flag against it.*

The flag will be used to locate the position of the hole from a distance.

5. *The Sharp IR sensor has got a blind spot of certain distance (about 10cm).*

The ball should not be kept in this region.

Hardware Requirements:

1. Firebird V ATMEGA2560 (Cost=Rs.16,875.00) as the basic skeleton on which we will implement the golf-playing bot.

- IR proximity sensors already attached to Firebird.

2. Zig-bee (XBee 802.15.4 OEM RF module 2.4GHZ) is used for communicating between the firebird and the PC (Cost = Rs.1665).

3. Wireless camera is used for taking snapshots. As part of this TV tuner card is also used (for connecting to PC) (Cost=Rs.1500 for camera + Rs.2800 for tuner card).
Specification:
· Image Pickup device - 1/3 1/4 inch CMOS
· TV system - PAL /CCIR NTSC/EIA
· Definition - 380 TV lines
· Scan frequency - PAL/CCIR:50Hz NTSC/EIA:60Hz
· Min illumination - 3 LUX
· Output power - 50mW 200mW
· Output frequency - 900MHz 1200MHz
· Power supply - DC 6V – 12V

4. Servo motors to rotate the camera and parts of the mechanical arm.

5. Mechanical Arm will be attached to the firebird robot to provide the required impulse to move the ball by a desired distance. The arm will be developed by us as per the requirements and updates regarding the same will be posted later.

Software Requirements:

- AVR Studio 4
- OpenCV for Image Processing.

Functionality of our bot:

The functions implemented by the bot to play the mini golf are as follows:

Find the ball:

The bot will detect ping pong balls placed randomly on the surface around it by rotating in its position and by using images from its camera.

Move towards the ball:

The bot after locating the ball aligns itself in the direction of the ball. This is ensured by keeping the vision of the camera perpendicular to the axis of the back two wheels of firebird. Then starts moving towards the ball until it is asked to stop at a particular predefined distance (say x) using the proximity IR sensors at the front.

Positioning itself to hit the ball:

First the bot turns 90 degrees to the right in its place and then rotates the camera 90 degrees to the left.

Now it rotates around a circle with the ball as the centre and “x” (as defined above) as the radius. From the images captured henceforth, it tries to locate its position where the flag is just behind the ball. This ensures that all the three are collinear with the ball in the middle.

It then uses the IR sensors and visual input from the camera to move within a predetermined “y” distance from the ball and in proper alignment for hitting it.

Hit the ball with appropriate force:

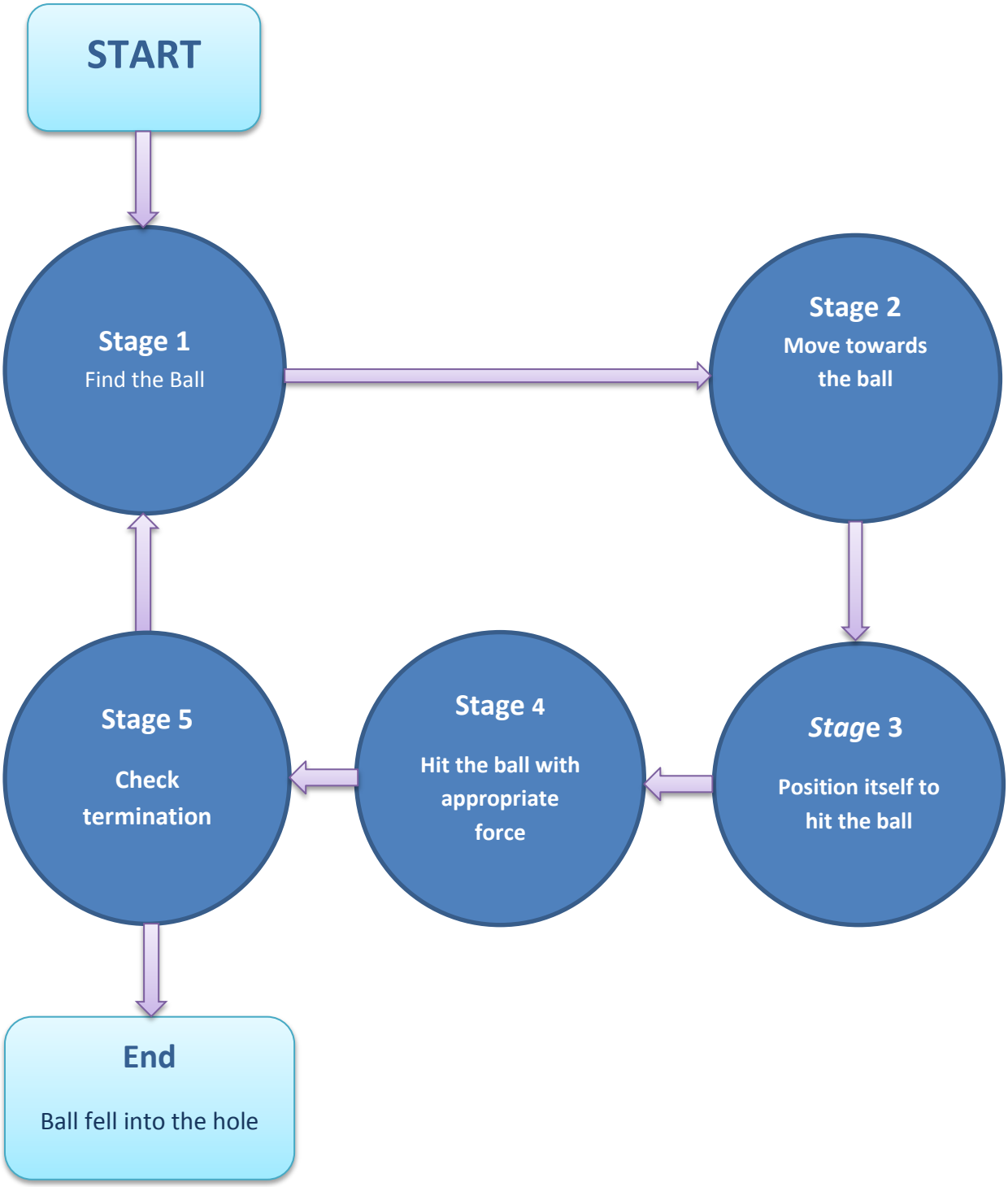
First it calculates the distance of the flag from the ball using the perceived height of the flag in the visual image and knowledge of the true height of the flag.

Then it computes the desired force to be exerted from the knowledge of the distance, weight of the ball and other miscellaneous data like friction of the surface, strength of the arm etc already known to it for tuning purposes.

Check termination:

The bot after hitting the ball checks whether the hit has been successful using images from the camera. If it finds no ball present on the surface a 360 degree rotation in its place it declares success and terminates. Else if it finds the ball somewhere, it again starts from the step 1.

State Diagram:



Conclusions:

We are developing this mini golf playing bot which can be improvised to operate in undulating terrain with better hardware (sensors and locomotion both). The underlying concept however will remain the same.