System Requirements

Group 4: Autonomous Foosball Table

Revision 1, October 26, 2015

|  |  |
| --- | --- |
| Yue(Ryan) Ma | 1147335 |
| Chenhe Li | 1158035 |
| Taha Hussain | 1200349 |
| Viktor Smirnov | 1154392 |
| Zi(Roland) Zhou | 1216063 |
| Alvin Li | 0957972 |

# 

# 

# **1 REVISIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0 | October 26, 2015 | Yue(Ryan) Ma,  Chenhe Li,  Taha Hussain,  Viktor Smirnov,  Zi(Roland) Zhou,  Alvin Li | Initial version |
| 1 | December 3, 2015 | Roland Zhou | updated constants table and input and outputs table. |

# 

# 

# **2 TABLE OF CONTENTS**

|  |  |
| --- | --- |
| [1 REVISIONS](#h.jjgxb2tbpw09) | 1 |
| [2 TABLE OF CONTENTS](#h.e0t9k2uxlrk5) | 2 |
| [3 INTRODUCTION](#h.qqlng31j1w5l) | 5 |
| [3.1 Purpose](#h.tqgzx53iy1wp) | 5 |
| [3.2 Scope of Work](#h.mnwfjnm3joor) | 5 |
| [3.3 Definitions and Abbreviations](#h.yoz5cm8sr9x7) | 6 |
| [3.4 References](#h.vj3sg6coel1h) | 6 |
| [4 NORMAL OPERATION](#h.8p1unbbaqz52) | 7 |
| [4.1 Description](#h.nb0fgiwua596) | 7 |
| [4.2 Normal Use Cases](#h.8jcgmr9rmmsz) | 7 |
| [4.2.1 Game is requested](#h.hm6bcnpoo6cl) | 7 |
| [4.2.2 Goal is scored](#h.fur0jlyd2h8j) | 7 |
| [4.2.3 Game is finished](#h.uvnriinz59rr) | 7 |
| [4.2.4 Difficult level setting](#h.l5zcqvny49rt) | 7 |
| [4.2.5 Emergency stop](#h.g7sksm1stsmf) | 8 |
| [5 SYSTEM DESCRIPTION](#h.aqokaxfndpqh) | 9 |
| [5.1 System Diagram](#h.7a94yc8wibvp) | 9 |
| [5.2 System Context Diagram](#h.f6v1zowcphw6) | 10 |
| [5.3 Functional Decomposition](#h.nnvb9holl5ns) | 12 |
| [5.4 System Variables](#h.p5ep63j3fzq8) | 14 |
| [5.4.1 Monitored Variables](#h.hf6s87qyooqg) | 14 |
| [5.4.2 Table 2. Monitored Variables](#h.p0ef6huu0oi) | 15 |
| [5.4.3 Controlled Variables](#h.tpjgr2p8rotx) | 16 |
| [5.4.4 Constants](#h.jaz3z2uteszo) | 17 |
| [6 REQUIREMENTS](#h.78jn0qpkq2vl) | 17 |
| [6.1 Functional Requirements](#h.h503we8bbny6) | 17 |
| [6.1.1 Mode Selection](#h.cgofemtev431) | 18 |
| [6.1.2 Gameplay mode detection](#h.nngxtn21ggmf) | 18 |
| [6.1.3 Player selection](#h.fc4lz084fyqv) | 18 |
| [6.1.4 Offense](#h.wib6cvezl1ri) | 19 |
| [6.1.5 Defense](#h.kqvfvlodyx8) | 20 |
| [6.1.6 Ball path prediction](#h.5e64ipgfbp8m) | 20 |
| [6.1.7 Emergency](#h.qk8ji9ysckdd) | 21 |
| [6.1.8 Waiting](#h.yrafk70jert) | 21 |
| [6.2 Non-Functional Requirements](#h.3iznujoy65is) | 21 |
| [6.2.1 System response time](#h.xvajw5bkre1a) | 21 |
| [6.2.2 Linear motion accuracy](#h.eietexo9ps6c) | 22 |
| [6.2.3 Maximum Linear motion speed](#h.lhc8wkk482a5) | 22 |
| [6.2.4 Angular motion accuracy](#h.jygwwvmohngp) | 23 |
| [6.2.5 Maximum ball kicking speed](#h.mwyorw3ajbyk) | 23 |
| [6.2.6 Ball detection accuracy](#h.ncx82cyqevw3) | 23 |
| [6.2.7 Foosman Angular position detection accuracy](#h.ukbwr1hd8mm9) | 24 |
| [6.2.8 Foosman linear position detection accuracy](#h.czfkburzpcka) | 24 |
| [6.2.9 Ball position detecting successful rate](#h.uuaq39j9dd0d) | 25 |
| [6.2.10 Ball blocking rate](#h.mft8qysejsbd) | 25 |
| [6.2.11 Unopposed goal score rate](#h.sg056h4gngwd) | 25 |
| [6.2.12 Total system cost](#h.w69vn7p14ep0) | 27 |
| [7 UNDESIRED SCENARIO HANDLING](#h.v7to0uj8vxrt) | 27 |
| [7.1 Foosball not moving in foosmen reachable area](#h.lptds4r738nr) | 27 |
| [7.2 Foosball not moving in user reachable area](#h.48tlkdhnvvh0) | 27 |
| [7.3 Ball jumps over the foosmen](#h.8pc1y4f590hx) | 27 |
| [7.4 Linear/Angular actuator not working](#h.aao554gm9d1j) | 28 |
| [8 REQUIREMENTS THAT ARE LIKELY TO CHANGE](#h.vsxivfqvgh6q) | 29 |
| [9 REQUIREMENTS THAT ARE NOT LIKELY TO CHANGE](#h.hx52puxhdeii) | 30 |
| [10 Appendix. Calculations](#h.liwrictkf2xl) | 29 |

# 

# 

# **3 INTRODUCTION**

## **3.1 Purpose**

The goal of the project is to design, develop and implement an affordable, safety autonomous foosball table.The foosball table shall meet the goals listed in the project goals document.

The purpose of this document is to define the high-level design and organization of the autonomous foosball system. This document should be used as the reference throughout the development cycle and a guide for validation and verification. It will also act as a guide for understanding the overall system.

## 

## **3.2 Scope of Work**

This project is constructing a retrofit/extension of a mini foosball table by employing software, electronic and mechanical methods. The entire design and implementation of mini foosball table is not within the scope of the project.

The system includes the following functionalities:

* Awareness of game start event
* Defence (blocking) against high-speed foosball
* Offence (foosball handling and shooting) where possible
* Pausing or stopping the game based on user input
* Edge case detection and handling
* Human Machine Interface

The following functionalities are deemed to be out of scope:

* Resetting the foosball after goal
* Resetting the foosball after dead ball (the foosball has completely stopped its motion and is not within reach of any player figure)
* Resetting the foosball stuck on user foosball men reachable area
* Detecting user spinning the rod

## **3.3 Definitions and Abbreviations**

|  |  |
| --- | --- |
| Term | Definition |
| foosball table | The autonomous mini foosball table |
| foosman | The players in the foosball table |
| ball | The foosball used in the table |
| user | The user playing the autonomous foosball table |
| row1 | System controlled foosball man nearest to user side on offense side rod |
| row2 | System controlled foosball man nearest to user side on defense side rod |
| enemy1 | User controlled foosball man nearest to user side on offense side rod |
| enemy2 | User controlled foosball man nearest to user side on defense side rod |

Table 1. Definitions and Abbreviations

## **3.4 References**

[1] “Foosball Rules” Internet: <http://www.foosball.com/learn/rules/>, Apr. 02, 2009 [Otc. 19, 2015]

[2] Michael Aeberhard, Shane Connelly, Evan Tarr, and Nardis Walker. fall 2007. Single Player Foosball Table with an Autonomous Opponent. Available: <http://www.eskibars.com/projects/foosball_robot/final_rpt.pdf>

[3] “Real-Time Foosball Game State Tracking”, Sven Bambach and Stefan Lee. [Otc. 19, 2015]

Available:<http://iu.svenbambach.de/foosball-tracking.pdf>

# **4 NORMAL OPERATION**

## **4.1 Description**

The system shall capture the position of the ball, predict the trajectory of the ball and use suitable strategies to arrange the positions and rotary velocity of the controlled foosman. The system should be able to deliver the ball to the player’s goal as well as preventing the ball from entering its own goal.

## **4.2 Normal Use Cases**

### **4.2.1 Game is requested**

The game is manually enabled by the user, either to start a new game or resume after a goal is scored. The system will first go to waiting mode and wait for the foosball to show in the field before initiating normal game behavior.

### **4.2.2 Goal is scored**

Each time the ball entering the goal is counted as a point. After a goal is scored, the system goes to waiting mode and waits for the user to retrieve the ball and put it in the field. The score has to be incremented manually by the user.

### **4.2.3 Game is finished**

The system will detect completion of the game, move the foosball player to the default position and goes to waiting mode.

### **4.2.4 Difficult level setting**

The system has three difficulty levels (“easy”,”normal”,”advanced”) and the default difficulty level setting is “normal”. User can change the difficult level of the system when the system is in waiting mode.

### **4.2.5 Emergency stop**

The system goes to emergency mode when unknown objects are present in the football field. Once in this mode, all the mechanical motions must stop within 500ms.

# **5 SYSTEM DESCRIPTION**

## **5.1 System Diagram**

Figure 1 illustrates that constant variable on the top view of foosball table; there is coordinate axis sign which demonstrate the positive direction of x and y axis. Contained constant variables: k\_length\_field, k\_goal\_length, k\_row2\_x\_max, k\_row2\_x\_min, k\_max\_linear\_range, k\_row1\_x\_max, k\_row1\_x\_min, k\_width\_field. (For detail about variables values see Table 4. Constant Table).

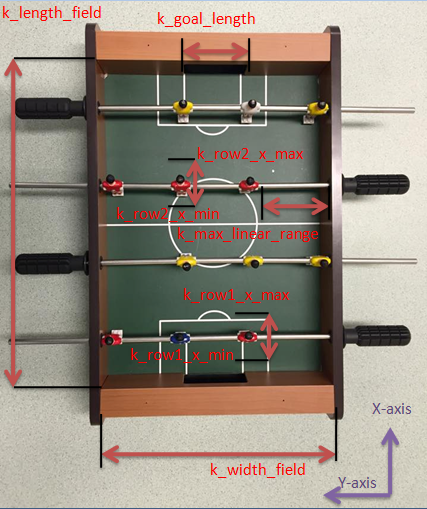


Figure 1

Figure 2 below demonstrates constant variables about each player and ball, contained variables are k\_foosman\_height, k\_foosman\_width, k\_ball\_diameter(For detail about variables values see Table 4. Constant Table).



Figure 2

## **5.2 System Context Diagram**

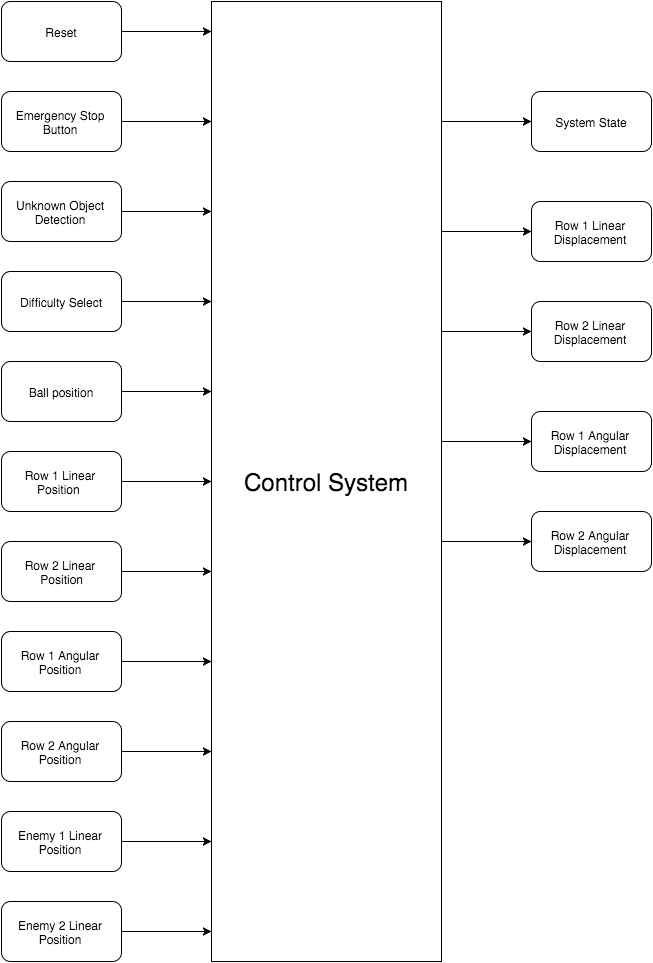
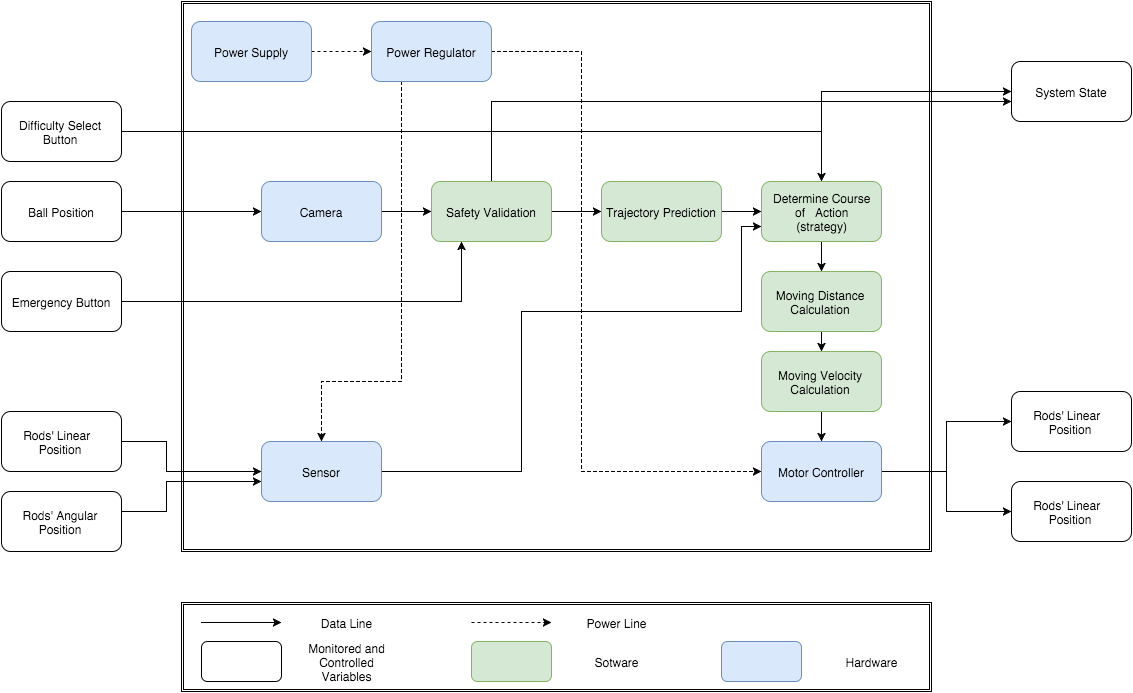


Figure 3. Context Diagram

## **5.3 Functional Decomposition**

(see Figure 4 below)





## **5.4 System Variables**

The variable names, descriptions/values and units are shown in the tables below. Variables names with “row1”, “row2”, “enemy1” or “enemy2” can be referred from “Table 1. Definitions and Abbreviations” in section 3.3 of this document.

### **5.4.1 Monitored Variables**

## 

|  |  |  |
| --- | --- | --- |
| **Variable** | **Units** | **Description** |
| m\_unknown\_object | Boolean | Other objects (e.g. human hand) show in the foosball table |
| m\_ball\_x | mm | Position of the foosball in the x-axis |
| m\_ball\_y | mm | Position of the foosball in the y-axis |
| m\_row1\_lp | mm | Linear position of the center foosman in row 1 |
| m\_row2\_lp | mm | Linear position of the center foosman in row 2 |
| m\_enemy1\_lp | mm | Linear position of the center foosman in enemy 1 |
| m\_enemy2\_lp | mm | Linear position of the center foosman in enemy 2 |
| m\_row1\_ap | degree | Angular position of the center foosman in row 1 |
| m\_row2\_ap | degree | Angular position of the center foosman in row 2 |
| m\_enemy1\_ap | degree | Angular position of the center foosman in enemy 1 |
| m\_enemy2\_ap | degree | Angular position of the center foosman in enemy 2 |
| m\_safety | Boolean | Safety shutdown button |
| m\_reset | Boolean | Reset button |
| m\_difficulty\_select | int | User requests difficulty level |
| m\_ball\_found | Boolean | Check if find ball in field |
| m\_goal\_user | Boolean | Check if user get goal |
| m\_goal\_robot | Boolean | Check if robot get goal |

Table 2. Monitored Variables

### **5.4.2 Controlled Variables**

|  |  |  |
| --- | --- | --- |
| Variable | Units | Description |
| c\_row1\_lp | mm | Linear position of row 1 |
| c\_row2\_lp | mm | Linear position of row 2 |
| c\_row1\_ap | degree | Angular position of row 1 |
| c\_row2\_ap | degree | Angular position of row 2 |
| c\_mode | N/A | System mode |
| c\_gameplay | N/A | gameplay mode |

Table 3. Controlled Variables

### **5.4.3 Constants**

|  |  |  |  |
| --- | --- | --- | --- |
| Constant | Value | Units | Description |
| k\_field\_length | 400 | mm | Length of the foosball field |
| k\_field\_width | 288 | mm | Width of the foosball field |
| k\_ball\_diameter | 28 | mm | Diameter of the foosball |
| k\_max\_linear\_range | 95 | mm | Linear movement range of the row |
| k\_goal\_width | 85 | mm | Width of the goal |
| k\_goal\_length | 40 | mm | Length of the goal |
| k\_foosman\_height | 75 | mm | Height of the foosman |
| k\_foosman\_distance | 85 | mm | Distance between the centers of two foosmen |
| k\_foosman\_width | 13 | mm | Width of the bottom part of the foosman |
| k\_cam\_width | 320 | pixel | Width of the frames |
| k\_cam\_height | 240 | pixel | Height of the frames |
| k\_edge\_x\_min | 20 | pixel | Minimum edge position in x-axis |
| k\_edge\_x\_max | 304 | pixel | Maximum edge position in x-axis |
| k\_edge\_y\_min | 18 | pixel | Minimum edge position in y-axis |
| k\_edge\_y\_min | 220 | pixel | Maximum edge position in y-axis |
| k\_row\_1 | 55 | mm | row one x-axis |
| k\_row\_2 | 255 | mm | row two x-axis |
| k\_enemy\_1 | 155 | mm | human controlling row one x-axis |
| k\_enemy\_2 | 357 | mm | human controlling row two x-axis |

Table 4. Constants

# **6 REQUIREMENTS**

## **6.1 Functional Requirements**

### **6.1.1 Mode Selection**

**Natural Language Description:** Determining the mode of the system. When the emergency button is pressed the system goes to emergency mode. When the ball is not on the foosball table, the system goes to waiting mode. When the ball shows in the table, the system goes to gameplay mode.

**Monitored Variables:** m\_ball\_x, m\_ball\_y, m\_reset, m\_safety, m\_unknown\_object

**Control Variables :** c\_mode

**Constants :** k\_field\_length, k\_field\_width

|  |  |
| --- | --- |
| condition | c\_mode |
| m\_safety | Emergency |
| ¬m\_safety⋀(m\_reset⋁m\_unkonwn\_object) | Waiting |
| ¬m\_safety∧¬(m\_reset⋁m\_unkonwn\_object)∧(m\_ball\_x≤k\_field\_length∧m\_ball\_x>0∧m\_ball\_x≤k\_field\_width∧m\_ball\_x>0) | Gameplay |
| ¬m\_safety∧¬(m\_reset⋁m\_unkonwn\_object)∧¬(m\_ball\_x≤k\_field\_length∧m\_ball\_x>0∧m\_ball\_x≤k\_field\_width∧m\_ball\_x>0) | Waiting |

### **6.1.2 Gameplay mode detection**

**Natural Language Description:** Determining the gameplay mode. When the ball is low speed in reachable area of system controlled foosmen, the gameplay is in offensive mode.

**Monitored Variables:** m\_ball\_x, m\_ball\_y, m\_ball\_speed

**Control Variables :** c\_gameplay

**Constants :** k\_row1\_x\_min, k\_row1\_x\_max, k\_row2\_x\_min, k\_row2\_x\_max, k\_row1\_y\_min, k\_row1\_y\_max, k\_row2\_y\_min, k\_row2\_y\_max

|  |  |  |
| --- | --- | --- |
|  | ((k\_row1\_x\_minm\_ball\_x)∧(m\_ball\_x k\_row1\_x\_max))⋁  ((k\_row2\_x\_min m\_ball\_x)∧(m\_ball\_x k\_row2\_x\_max)) | ¬(((k\_row1\_x\_minm\_ball\_x)∧(m\_ball\_x k\_row1\_x\_max))⋁  ((k\_row2\_x\_min m\_ball\_x)∧(m\_ball\_x k\_row2\_x\_max))) |
| m\_ball\_speed 0.5m/s | c\_gameplay==Offensive | c\_gameplay== Defensive |
| m\_ball\_speed > 0.5 m/s | c\_gameplay==Defensive |
|

### **6.1.3 Player selection**

For a particular offense or defense play the correct foosman has to be selected. Each foosman has its own workspace.A foosman is selected based on the trajectory of the ball and when its predicted path crosses with one of the friendly rows. The region of where this intersection takes place will decide which foosman will be chosen for defense. For offense whatever region the ball is in will lead to the selection of a foosman.During offense it is possible that the ball drifts out of one foosman's region, in that case the current offense play will end and a new one will begin with the payer whose region the ball drifts into.

### **6.1.4 Offense**

**Monitored Variables:** m\_ball\_x, m\_ball\_y, m\_row1\_lp, m\_row1\_lp, m\_row2\_lp, m\_enemy1\_lp,m\_enemy2\_lp

**Control Variables :** c\_row1\_lp, c\_row2\_lp, c\_row1\_ap, c\_row2\_ap

**Constants :** All

The offense requirements are divided in several levels. This has been done as we are still not completely sure what reaction time we will be able to obtain for the system.For these offense requirements we assume that:

* The system is in gameplay offense mode
* initialization has has been completed
* A player has been selected to perform the offense action

Level 0(basic requirements):

* Foosmen are able to shoot the ball towards the enemy side with the required rotational speed.
* Foosmen are able to detect when ball is not in front of it and is able to move itself in a position and orientation allowing for a shot.

Level 1(target requirements):

* Foosmen are able to pass the ball to the side to other foosmen. A pass is detected by other foosmen and is intercepted properly by rotating in the direction of the previous goal. This will let the ball roll in front of the foosmen creating an opportunity for a shot
* Foosmen are able to detect enemy players and shoot in between them by making use of passing

Level 2 (extra requirements):

* Foosmen are able to make use of specialized shot such as fast passing and instant shooting.
* Foosmen are able to hit the ball at an angle from the edge of the player

Shot completion detection

Shot completion will be detected when the particular foosman has reached the rotation angle that was assigned to it and is within tolerance of the angular position( k\_R1A\_tolerance). After a shot in any direction the mode will be switched to defense automatically as we can predict that the ball will be outside the friendly region. This will also prevent counterattacks from the enemy team.

### **6.1.5 Defense**

**Monitored Variables:** m\_ball\_x, m\_ball\_y, m\_row1\_lp, m\_row1\_lp, m\_row2\_lp, m\_enemy1\_lp,m\_enemy2\_lp

**Control Variables :** c\_row1\_lp, c\_row2\_lp, c\_row1\_ap, c\_row2\_ap

**Constants :** All

Theses defense requirements apply for the system in its defense mode. Like the offense requirements, the defense requirements are also divided into different level and have the following assumptions:

* The system is in gameplay mode
* Initialization process is completed
* Gameplay is in defense mode
* There is a foosman selected to perform the defense actions intended

Level 0(basic requirements):

* The foosmen are able to move to the predicted position of the ball to block it within sufficient time.
* The foosmen will turn 30 to 45 degrees towards the predicted direction of the ball and keep this orientation while performing the blocking action.

Level 1(target requirements)

* Ball prediction will be used out so players will align themselves with the position the ball is going to be when it reaches the row than instead of aligning with the ball. These prediction will include bounces off the wall.
* The foosman will not block and will move away from the ball’s predicted trajectory if the ball is travelling towards the enemy’s goal.
* Ball prediction should be taking place for only the rows involved. If the ball is in front of row 1 then row 2 should not worry about the ball and should revert to its default position.
* If the ball is not blockable then the Foosman will hit the ball through a linear motion towards the side instead of towards the goal.

### **6.1.6 Ball path prediction**

Path prediction should predict the path from the ball to one of the ends of the field. The intersection of the path of the ball with the friendly rods should be given as coordinates value for use with defense strategy and offense strategy. The path prediction should also predict deflections off the wall and collision of the ball with enemy players.

### **6.1.7 Emergency**

The system shall immediately stop all its mechanical and electrical operations.

### **6.1.8 Waiting**

The system shall move the foosball player to default position.

## **6.2 Non-Functional Requirements**

### **6.2.1 System response time**

|  |  |
| --- | --- |
| *Description:* | System response time is the amount of time the system takes to respond to user kicking the ball including image processing time to locate the ball and linear motion required to align the foosman with the ball. |
| *Rationale:* | The value is based on an estimated shot speed[2] of an average adult player and estimated time needed to located the ball[4]. For the approximation the full movement of the rod is assumed. Refer to Appendix[A]. |
| *Importance:* | 5 |
| *Unit:* | milliseconds (ms) |
| *Marginal Val:* | ≤150 |
| *Ideal Val:* | ≤15 |
| *Volatility:* | 3 |

### **6.2.2 Linear motion accuracy**

|  |  |
| --- | --- |
| *Description:* | The accuracy of moving the rod linearly to a desired position |
| *Rationale:* | The value is determined by the accuracy of the motors as well as the mechanical structure of the system. This is essential for blocking or kicking the ball. |
| *Importance:* | 3 |
| *Unit:* | millimeters (mm) |
| *Marginal Val:* | 0.5 |
| *Ideal Val:* | 0.1 |
| *Volatility:* | 2 |

### **6.2.3 Maximum Linear motion speed**

|  |  |
| --- | --- |
| *Description:* | The maximum linear speed of the system controlled rod |
| *Rationale:* | The value is determined by the motors,linear motion range and affects the response time significantly. This value is essential for blocking or kicking the ball. Refer to Appendix[B]. |
| *Importance:* | 4 |
| *Unit:* | meter/second (m/s) |
| *Marginal Val:* | ≥0.6 |
| *Ideal Val:* | ≥7 |
| *Volatility:* | 3 |

### **6.2.4 Angular motion accuracy**

|  |  |
| --- | --- |
| *Description:* | The accuracy of rotating the rod to a desired position |
| *Rationale:* | The value is determined by the accuracy of the motors as well as the mechanical structure of the system. This is essential for kicking the ball |
| *Importance:* | 3 |
| *Unit:* | degree |
| *Marginal Val:* | 2 |
| *Ideal Val:* | 0.5 |
| *Volatility:* | 2 |

### 

### **6.2.5 Maximum ball kicking speed**

|  |  |
| --- | --- |
| *Description:* | The maximum speed the ball can reach after system kicked the ball |
| *Rationale:* | The value is based on an estimated shot speed and reaction time of an average adult player[2], mass of the ball and dimensions of the foosball field. |
| *Importance:* | 4 |
| *Unit:* | meter/second (m/s) |
| *Marginal Val:* | ≥2 |
| *Ideal Val:* | ≥6 |
| *Volatility:* | 2 |

### **6.2.6 Ball detection accuracy**

|  |  |
| --- | --- |
| *Description:* | The accuracy of detecting the position of the ball |
| *Rationale:* | This is essential for ball path prediction and decision making which determines the ball blocking rate and goal score rate of the system |
| *Importance:* | 4 |
| *Unit:* | millimeters (mm) |
| *Marginal Val:* | 3.0 |
| *Ideal Val:* | 1.0 |
| *Volatility:* | 2 |

### **6.2.7 Foosman Angular position detection accuracy**

|  |  |
| --- | --- |
| *Description:* | The accuracy of detecting the angular position of the foosman |
| *Rationale:* | This is essential for making accurate blocks and shots. Foosman need to be at a specific angular to block the ball. |
| *Importance:* | 4 |
| *Unit:* | degree |
| *Marginal Val:* | 2 |
| *Ideal Val:* | 0.5 |
| *Volatility:* | 3 |

### **6.2.8 Foosman linear position detection accuracy**

|  |  |
| --- | --- |
| *Description:* | The accuracy of detecting the angular position of the foosman |
| *Rationale:* | This is essential for making accurate shots. |
| *Importance:* | 4 |
| *Unit:* | millimeters (mm) |
| *Marginal Val:* | 1.5 |
| *Ideal Val:* | 0.3 |
| *Volatility:* | 3 |

### **6.2.9 Ball position detecting successful rate**

|  |  |
| --- | --- |
| *Description:* | The ability of the system to detect the position of the ball when the ball is in the foosball table |
| *Rationale:* | This value is based on current prototypes with similar total system cost. |
| *Importance:* | 4 |
| *Unit:* | percentage (%) |
| *Marginal Val:* | 85 |
| *Ideal Val:* | 98 |
| *Volatility:* | 3 |

### **6.2.10 Ball blocking rate**

|  |  |
| --- | --- |
| *Description:* | The ability of the system to block human shots to prevent goal being scored |
| *Rationale:* | This value is based on current prototypes with similar total system cost. |
| *Importance:* | 4 |
| *Unit:* | percentage (%) |
| *Marginal Val:* | 70 |
| *Ideal Val:* | 85 |
| *Volatility:* | 3 |

### **6.2.11 Unopposed goal score rate**

|  |  |
| --- | --- |
| *Description:* | The ability of the system to score in the opponent’s goal when the goal is unopposed |
| *Rationale:* | This value is based on current prototypes with similar total system cost. |
| *Importance:* | 3 |
| *Unit:* | percentage (%) |
| *Marginal Val:* | 50 |
| *Ideal Val:* | 70 |
| *Volatility:* | 4 |

### **6.2.12 Total system cost**

|  |  |
| --- | --- |
| *Description:* | The cost of all parts and technologies needed to build this system |
| *Rationale:* | This value is based on researches on current available prototype of this system and parts needed for building such system |
| *Importance:* | 4 |
| *Unit:* | CAD ($) |
| *Marginal Val:* | 400 |
| *Ideal Val:* | 400 |
| *Volatility:* | 2 |

# **7 UNDESIRED SCENARIO HANDLING**

## **7.1 Foosball not moving in foosmen reachable area**

* Electrical components failure
* mechanical parts are stuck
* ball is stuck

The system will detect the event and stop all the actuators and goes to emergency mode. User will be notified by an error message when a fault occurs.

## **7.2 Foosball not moving in user reachable area**

* ball is stuck
* user is away

The system will stay in gameplay mode.

## **7.3 Ball jumps over the foosmen**

* strong shot from user
* strong shot from robot

The system will goes to waiting mode and reset foosmen to default position.

## **7.4 Linear/Angular actuator not working**

* user pulling or pushing the system controlled rods
* electrical components fail
* ball is stuck

If the the actuator does not work, the system will automatically go to emergency mode and a signal will notify the user that a fault has occurred.

# **8 REQUIREMENTS THAT ARE LIKELY TO CHANGE**

Since the detailed design are not completed yet, most of the performance related nonfunctional requirements are likely to change in the future. Below is the table of requirements that are likely to change.

|  |  |
| --- | --- |
| **Requirements Likely to change** | **Unstable reason** |
| k\_cam\_width,  k\_cam\_height | Ball or foosman location accuracy may larger than the marginal value. We may change it to vga resolution (640x480). |
| k\_edge\_x\_min,  k\_edge\_x\_max,  k\_edge\_y\_min,  k\_edge\_y\_max | Camera resolution may change to 640x480 due to the accuracy of location detection may larger than marginal value. |
| servo motor |  |
|  |  |
| Angular motion accuracy | Angular motion is powered by servo motor, fraction and force from ball can be various on different situation |
| Ball detection accuracy | Ball can be detected by camera after image processing, software detect ball based on image color and size; accuracy of ball detection can be affected by surrounding environment(for example brIghtness) |
| Foosman angular detection accuracy | Foosman angular detection will be analysis through image processing, the accuracy of Foosman angular detection is determined by surrounding environment (for example brIghtness) |
| Foosman linear position detection accuracy | Foosman linear position detection will be analysis through image processing, the accuracy of Foosman linear position detection is determined by surrounding environment (for example brIghtness) |
| Ball blocking rate | The rate of ball blocking cannot be determined by machine performance, user can have different skill to play; the rate of ball blocking is affected by user skill level |
| Unopposed goal score rate | Goal score function is based on AI offensive strategy analysis, but field ground is not ideally and field cannot be exactly same, error will be produced on different surface. |
| Total system cost | The budget might changed due to the change of hardware |

Table 5. Requirement That Are Likely to Change

# **9 REQUIREMENTS THAT ARE NOT LIKELY TO CHANGE**

Among the list of functional and nonfunctional requirements , most of the functional requirements are associated with the layout and behaviours of the system. We want to keep these requirements stable and stay away from changing these requirements in the future.

The nonfunctional requirements that are not likely to change are mostly restricted by the constraints so there are not much space for changes in the future. The table below contains the detailed lists of requirements that are not likely to change.

|  |  |
| --- | --- |
| **Requirements Not likely to change** | **Description** |
| Mode Selection requirements | These are functional requirements that describe the basic layout of system. |
| Offense requirements | These are among the functional requirements that set the basic behaviours of the system. |
| Defense requirements | These are among the functional requirements that set the basic behaviours of the system. |
| Emergency requirements | These are safety related requirements that should not be changed. |
| Maximum linear motion speed | This is restricted by the performance of the motors. It cannot be slower than 0.5m/s |
| Maximum ball kicking speed | This is restricted by the motors used. It cannot be slower than 4m/s. |
| Ball detection successful rate | 85% successful rate is the minimum acceptable performance |

Table 6. Requirements That Are Not Likely to Change

# **10 Appendix. Calculations**

[A]

To determine the system response time:

From the measurements made to the foosball table, we have the following data:

Minimum distance between two foosmen:

Max ball speed observed:

Therefore,

This value of 15 ms is for the best system response(likely to be changed later).

[B] Maximum Linear motion speed

k\_max\_lp = 85mm

For Marginal Case:

140ms out of the 150 ms response time is used to perform the linear motion.

Then,

For Ideal Case:

12ms out of the 15 ms response time is used to perform the linear motion.

Then,