GOALKEEPER FOR ROBOCUP SPL

By

NEELAM UMBREEN

NUST201463113MSMME62114F

Supervised by: Dr. Yasar Ayaz

School of Mechanical and Manufacturing Engineering
National University of Sciences and Technology
H-12 Islamabad, Pakistan
May, 2016

GOALKEEPER FOR ROBOCUP SPL

A thesis submitted in partial fulfillment of the requirement for the degree of Masters of Science

In

Robotics and Intelligent Machines Engineering

By

NEELAM UMBREEN

NUST201463113MSMME62114F

Supervised by: Dr. Yasar Ayaz

School of Mechanical and Manufacturing Engineering
National University of Sciences and Technology
H-12 Islamabad, Pakistan
May, 2016

DECLARATION

It is hereby declared that this research study has been done for partial fulfillment of requirements for the degree of Masters of Sciences in Robotics and Intelligent Machines Engineering. This work has not been taken from any publication. I hereby also declare that no portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification in this university or other institute of learning.

Neelam Umbreen

Every breath of my life and drop of blood in my body is dedicated to my family to whom I owe everything, especially to my *FATHER*.

ACKNOWLEDGEMNT

"In the name of Allah the most Merciful and Beneficent"

First and foremost, praises and thanks to Allah, the Almighty, who is my Lord and who is my all who never let efforts go wasted. His blessings and guidance throughout made me complete this project successfully.

I don't have words to thank to my family to whom I owe everything, the ones who can never ever be thanked enough, for the overwhelming love and care they bestow upon me.

I would like to express my profound and sincere gratitude to my research supervisor, Dr. Yasar Ayaz, for giving me the opportunity to work under his supervision. His timely contribution, encouragement, kindness and valuable advices helped me shape my work into its final form.

Special thanks right from the depth of my heart to Zain Murtaza from SMME for his sincere and efficient contribution and assistance with various problems all the time.

I would like the whole hearted thank my friends for their prayers and moral support and especially to my beloved sister Ainee, for supporting me spiritually throughout writing this thesis and my my life in general.

Last but not the least; I would like to thank each and every person for supporting me throughout this research phase in any way.

Table of Contents

LIST OF FIGURES

ABSTRACT

RoboCup is amongst the most illustrious of international robotic competitions, held annually. Its Standard Platform League comprises of soccer competitions between teams of humanoid robots.

Robocup competition provides an interesting platform where intelligent systems are studied and developed to interact with real physical objects.

NAO robot from Aldebaran Robotics is the hardware platform for RoboCup SPL; while incorporates sonar sensors, joint encoders, cameras, 25 degree of freedom, foot pressure sensors, Wifi, Embedded Linux and Atom processor. The key objective of this project is to make a multi-agent system which can contribute towards the complete software module of robots which can play soccer with each other.

The robots operate fully autonomously with only local vision and highly constrained inter-robot communication, without any external intervention.

In real soccer goal keeper plays a totally different role from other players. Hence a novel idea is presented to improve the performance of goal keeper for Robocup SPL competition.

Chapter 1

INTRODUCTION

1.1 Motivation

In robotics, development of autonomous group of agents requires integration of different features of the agents to accomplish a certain goal for the group. For a mobile robot agent, all the information received from the different sensors, e.g., cameras , is understood using image processing, integrated for better perception of surrounding environment, e.g., to self-localize, and then performed an action to have a desired effect in the environment, e.g., using motion control and obstacle avoidance algorithms.

Above all is the most important key feature for a robot to achieve a desired goal, autonomy. The robot needs to integrate and coordinate all its subsystems, in order to achieve the desired goal.

1.1.1 The RoboCup Project

The RoboCup (Robot World Cup Initiative) project, [1], is a robotic competition which the main intention is to promote robotics and the research on artificial intelligence with the ultimate challenge stated as follows:

By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, comply with the official rule of the FIFA, against the winner of the most recent World Cup.

This robot soccer competition provides an enriched research oriented environment because it is a dynamic and partially known environment where each one of the agents should have a specific role and needs to coordinate its actions and behaviors to achieve a desired goal.

1.1.2 The RoboCup SPL

In the Standard Platform League standard hardware platform is used by all teams, so that more effort is put on software development rather than the robots mechanics[3] .The robots are fully autonomous; i.e., there is no external intervention by either humans or computers during the games.

Figure 1 *RoboCup SPL field*

1.1.3 The Goalkeeper

In a robotic soccer team, the goalkeeper has particular challenging characteristics that are different from the other teammates. The main purpose of a goalkeeper is to defend the goal from the kicks of the opponent teams that means the active area for goalkeeper always remains near its own goal. Besides this simple objective and limited area, the goalkeeper should have a perfect coordination amongst all its components such as: perception of ball and its tracking, intercepting the ball before it reaches the goal areal, covering the goal and clearing the ball from the goal area.

Therefore, these features of a robot soccer goalkeeper make its role important and distinct in a team.

1.2 Use Case Diagram for a Soccer Game

Use Case Diagram for a soccer player is shown in figure 1.2. Following is the description of actors, use cases and relationship between them.

USERS – Users are the external entities who can interact only when game is not being played actively.

Initialize players are positioned on the field automatically or manually, before the game play begins.

Start is an indication that players have occupied their positions and now game play can begin.

End means the user has the authority to end the game.

SERVER

Game Controller helps the server to transmit data to the player client that consists of information regarding or communication from other clients.

AI

Receive Message tells that the player client can receive data that is send by the server.

Send Message shows that the player client can communicate information to the server.

Parse Data indicates that the player client can parse the information it receives from the server.

Decide shows that the player can choose its next action based on the internal data and the information it receives from the server.

Act indicates that after selecting an action based on decision, the player client can implement the selected action. This might be done by communicating with the server or waiting for an event.

Kick involves two actions passing the ball to team mate and shooting the ball into opponent’s goal.

Run shows the movement of player into the field.

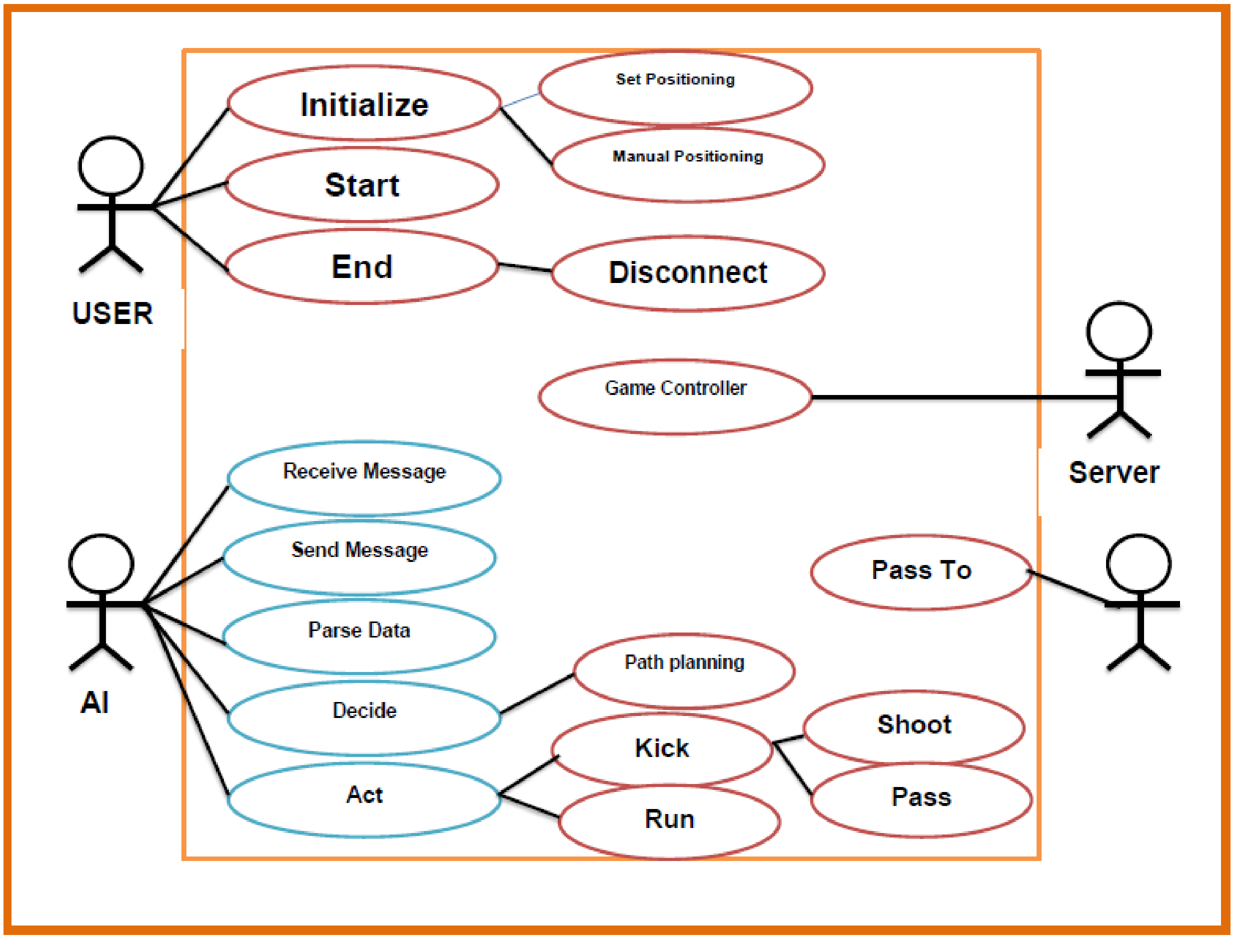


Figure 2 *Use Case Diagram for a Soccer Player*

Chapter 2

LITERATURE REVIEW

Although in recent decades numerous researches is conceded out in the field of humanoid robots, still in spite of all the researches there is no sufficient research about the right decision of the goalkeeper to hedge in order to defend goal. Some articles addresses relevant areas, such as how to lessen the damage to the robot in case of collision of robot with the ground [3], [4] or localization and vision of robots [2], [5], but a explicit discussion regarding the goalkeeper robots has only been found in [6].

Goalkeepers that exhibit Multi-behavior have been studied in the dominion of the RoboCup mid-sized league. Even though there challenges in Standard Platform League (SPL) and simulation league are distinctive that are not existent in the mid-sized league, the study of behavior control within the mid-sized league is still applicable.

Menegatti et al [18] implemented behavior and motion control for a goalkeeper using an ad hoc model. Their goalkeeper intercepts shots coming toward the goal by remaining in an arc in front of the goal. Work presented here was further improved by Lausen et al [15]. They used a hierarchical state machine comprises of 2 levels that synchronizes primitive tasks and behaviors. Complex motions in a task are approached by a non-linear algorithm.

Garcia et al [13] generated autonomous behavior using ethological inspired architecture and trailed SPL rules for demonstrating a humanoid goalkeeper. The behavior of goalkeeper is implemented using a premeditated structure for planning (self-positioning) and a reactive control mechanism for saving goal. They report being able to track the ball 100% of the time, positioning itself properly 84% of the time, and saving goals from 62% of the trajectories tested.

Bozinovski and Schoell [9] worked on an approach based on an evolving behavior model for the role of a goalkeeper. They used emotion based self-reinforcement learning algorithm and a learning demonstrated a curve as result from training experiments.

Birbach et al [7] used DLR's humanoid Rollin' Justin to implement a real time perception system able to catch ball. The system uses Multiple Hypothesis Tracker to tracks and predicts the trajectory of balls thrown towards the agent that uses Unscented Kalman Filters.

Adorni et al. [7] suggested a straight-forward method for a Middle-Size League (MSL) agent in the beginning of RoboCup. It retrieves images frame by frame and uses these images to calculate the current ball position. Then it uses a look-up table mechanism to calculate the position of ball relative to the agent. It then estimates ball position, motion direction and speed using two successive frames.

Research in soccer agents and robots has several aspects of gameplay such as efficient omnidirectional movement [17,16] and accurate perception [18].

Human goalkeepers predict the direction of the ball using visual cues in the shooter's body posture [11], before the shot has been performed. However, in the framework of the RoboCup 3D soccer competition, goalkeeper agents depend on merely on the ball trajectory after shooting due to the restraints (ex. noisy perception, limited computational resources) posed by perceiving the opponent robot's posture.

In spite of copious research into the various challenges posed by the robot soccer context, work done on the goalkeeper agent is limited to defining simple rules that increase the chances of saving a goal. Moreover, goalkeepers use the same methods for perception as other field player agents.

Goalkeeper has a unique role as it is the only player agents allowed to purposely dive and block the ball using their hands. Diving is the most popular goalkeeper skill present in the competition, but decision about how and when to activate this feature depends upon predefined rules[18].

Chapter 3

HARDWARE PLATFORM: NAO

3.1 NAO's Description

Nao is a humanoid robot. Height of Nao is 58 cm, it weigh 4.3 Kg. The Nao robot carries a full computer on board with an x86 AMD Geode processor at 500 MHz, 256 MB SDRAM, and 1 GB ash memory running an Embedded Linux distribution. It uses a 6-cell Lithium-Ion battery. It uses an IEEE 802.11g wireless or a wired ethernet link which provides about 45 minutes of continuous operation and communicates with remote computers.

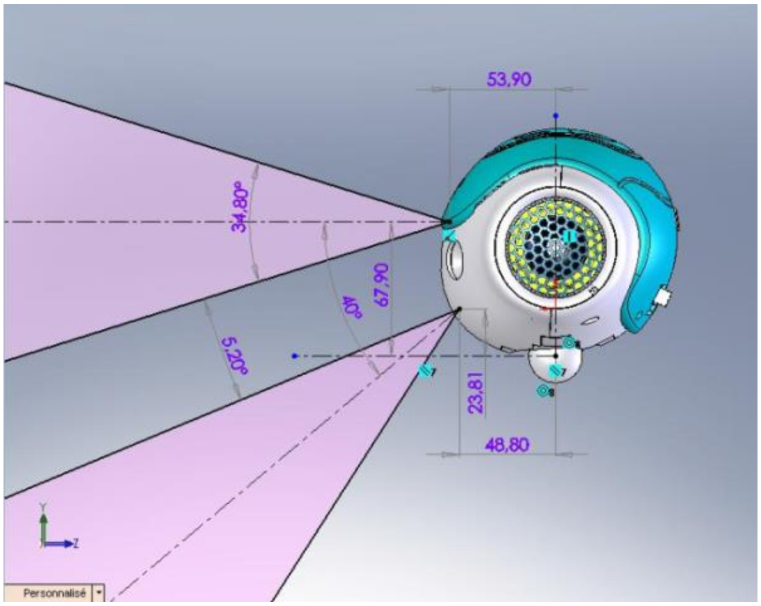


Figure 3 *Nao's field of View*

3.1.1 Cameras

In the NAO V3, the cameras are mounted on the head of robot therefore they provide non-overlapping views. At a time only one robot is active and the view can be transferred from one to the other almost instantaneously.

3.1.2 Connectivity

There are three means for Connecting to a robot.

1) The NAO robot, can be connected directly or sharing the same network, through a wired ethernet link.
2) An IEEE 802.11g wireless card is available for connection
3) Aldebaran provides a serial cable, which comes into play when we want to debug the robot, and read messsages concerning the Nao's procedures.

3.1.3 Audio, Sensors, Various

It has a pair for stereo audio perception. Nao sense obstacles in front of it, using two ultrasonic sensors placed on its chest. Information about its instantaneous body movement is provided by the torso IU (inertial unit).

Finally, force applied to the feet is delivered by an array of resistors, while joint position at each time is recorded by encoders on all servos. Information about feet collisions with obstacles is measured by two bumpers on the feet. The Nao robot has a total of 21 degrees of freedom.

Stereo loudspeakers and a series of LEDs complement its motion capabilities with auditory and visual actions.

3.1.4 NaoQi

NaoQi is the framework that acts as a interface between the robot and high-level languages, such as C, C++, and Python. NaoQi offers a distributed programming and debugging culture. Its architecture consists of modules and brokers that allows local and remote module to be run on the robot.

Chapter 4

FEATURES OF A GOALKEEPER

4.1 VISION

4.1.1 Corners Detection

The corner detection done is based on the information of the convexity defects found in the image from the contours points and the convex hull information.

4.1.1.1 Convexity defects

To understand what convexity defects are we first have to understand about contours points and their corresponding convex hulls.

Contours

Contours are closed boundaries around the binary image blobs they are made by connecting necessary number of points to contain all detected binarized blob.

Convex Hull

As the name indicates convex hull is the convex shape that contains all of the contour points.

Convexity Defects

Convexity defects correspond to all those areas which are part of the convex hull but not contained in the original contour from which the convex hull was made. The depth of the convexity defect is the maximum perpendicular distance from the line between the start and finish of the defect to any point within the defected portion.

In the following image the blue color shows the contours made around the detected hand, the yellow color shows the convex hull around the contour hand and the black arrows shows the convexity defects depth in the respective convexity defects. The portions which are not inside the contour but are inside the convex hull are the convexity defects.

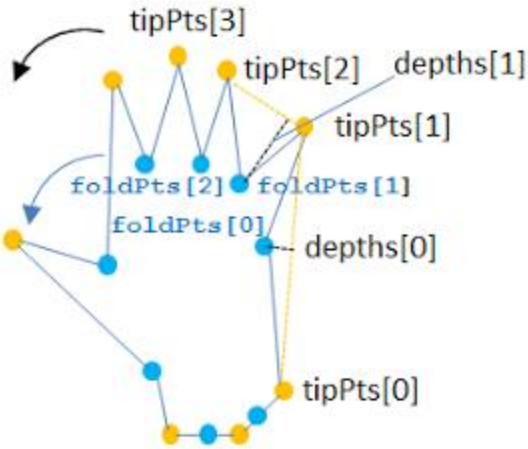


Figure 4 *Contour (Blue), Convex Hull (Yellow), Convexity Defects and their Depth (Black)*

4.1.1.2 Method explanation

The method chosen has the following steps in its algorithm.

1. Input coming from the robot.

2. Extract the field portion from the image.

3. Take inverse perspective transform.

4. Binarize the transformed image for white color.

5. Make contours around the detected white color blob

6. Find convex hulls of all the blobs

7. Draw corner points (circles) on the convexity defects areas if its depth is greater than a certain threshold.

8. Classify corners into L, T or X type based on the number of lines detected on the square whose pixels are traversed around the corner points.

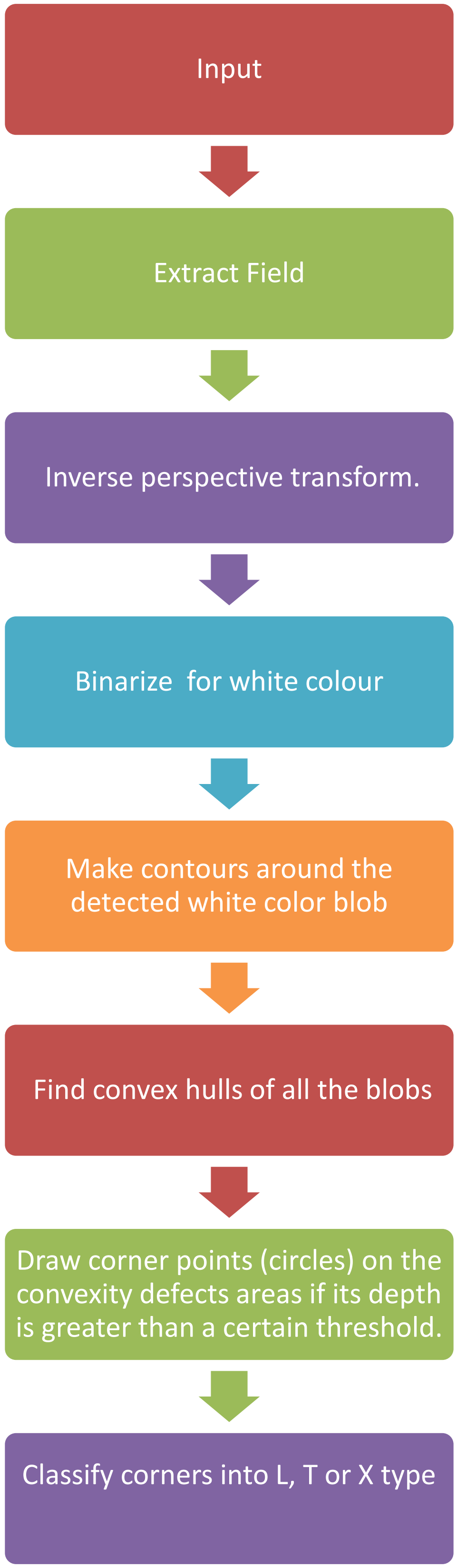


Figure 5 *Steps involved in corner detection*

The major algorithm steps are showing in the following image

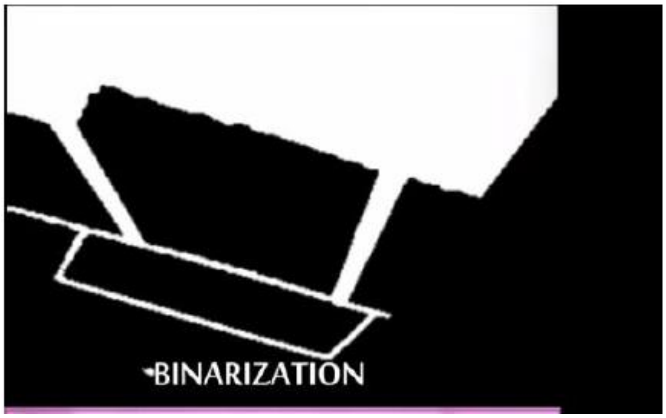


Figure 6 *Binarization*

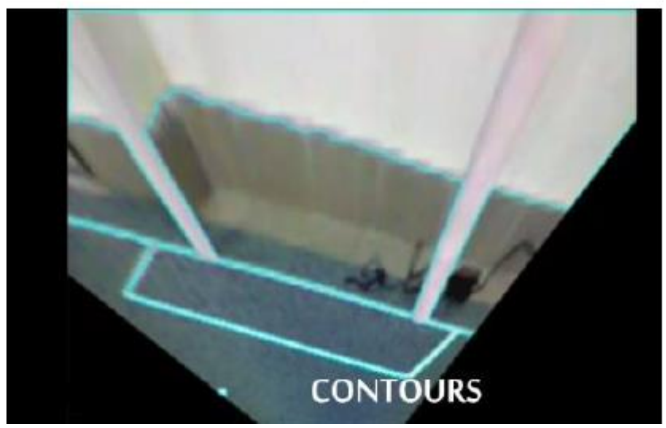


Figure 7 *Contours*

Figure 8 *Convex Hulls*

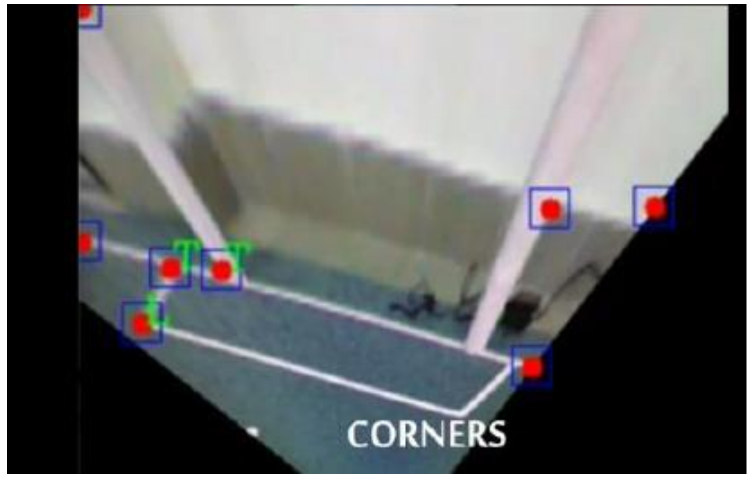


Figure 9 *Corners*

4.1.2 Goal Post Detection

We have found the goal post base points based on the topological structure of the goal post contours.

4.1.2.1 Topological structural analysis

We have approximated the shape of the goal post up to a level where there is only one base point at max for each leg. Otherwise we had more than one point on each legs of the goal post which resulted in getting the two base points on the same leg. Also we have utilized the direction of the legs of the goal post so that correct results are found even when tilted image comes as an input.

4.1.2.2 Method explanation

The method chosen has the following steps in its algorithm.

1. Input coming from the robot.

2. Extract the field portion from the image.

3. Binarize the goal post by taking yellow color from the image.

4. If the color blob exceeds a certain threshold for area then go to the next step, otherwise take another image for goal post detection.

5. Approximate the shape of the found contour.

6. Now get the first base point of the goal post by finding the lowest point of the goal post approximated contour.

7. Now find the direction that the first leg of the goal post points towards.

8. Find the other leg of the goal post by find the direction opposite to the first leg of the goal post.

9. If the other leg is found then go to the next step, otherwise take another image as input

10. Find the other base point of the goal post by getting the end point of the other leg of the goal post.

4.1.3 Field Extraction

Here convex hulls and field height histograms are utilized.

4.1.3.1 Downsampling, height histograms and convex hulls

The field extraction is done through downsampling the binarized image so that the unfilled areas present in the binarized image are smoothed out. Then we extract the outer boundaries of the field through convex hull made with the help of the height histogram.

4.1.3.2 Method explanation

The method chosen has the following steps in its algorithm.

1. Input coming from the robot.

2. Binarize the green color found on the image.

3. Downsample the image 3 to 4 times and smooth the image with each downsampling.

4. Now threshold the image and make a binary image so that we can have a clear representation of the green blob found in the image.

5. Make the height histogram of the field boundaries from the bottom up.

6. Include the height points and the bottom two corner points of the image and make a convex hull of all the points.

7. Extract the portion of the image contained in the convex hull to get the extracted field.

4.2 LOCALIZATION

4.2.1 Distance perception

4.2.1.1 Available hardware:

For distance perception available hardware that we can use is:

- Monocular cameras
- Ultrasonic sensors

Below we discuss the viability of each sensor as input.

Ultrasonic sensors:

These sensors send a ultrasonic pulse and measures the time of flight by capturing the echo. This allows it to calculate the distance to an object. Currently the land marks that are being detected are plane corners and field lines so these sensors cannot be used as feasible sensors for measuring distance to landmark that we are detecting through our perception module.

Monocular cameras:

Nao has two monocular cameras with none overlapping field of views as below:

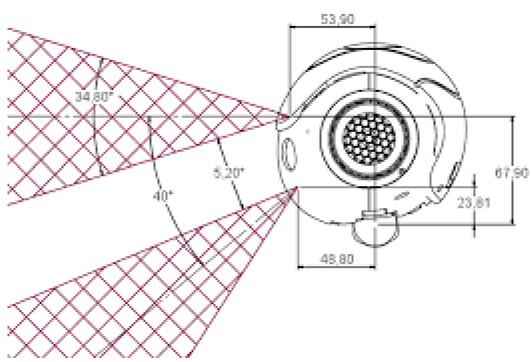


Figure 10 *NAO Robot Cameras*

In order to perceive depth using the camera we can employ the techniques of stereo vision. These techniques can be employed using determined patterns of camera motion in order to create a virtual stereo camera but unfortunately this technique becomes too computationally expensive therefore we resort to another technique that solely uses monocular camera.

As our landmarks are on ground plane so we can utilize the assumption that each point we see in image pixel lies on the ground with this assumption we drive equations to solve for the local coordinate of the object we see in the image.

4.2.2 Land Mark Hypothesis

Once a feature point is detected on the image it has to be correlated with a landmark on the map. But we have a problem here as same feature point correlates to multiple landmarks on the map.

Feature Points	Land Marks
L corner	Four outer corners, four penalty area corners
T corner	Four penalty area corners
X corner	Circle crossing on the line

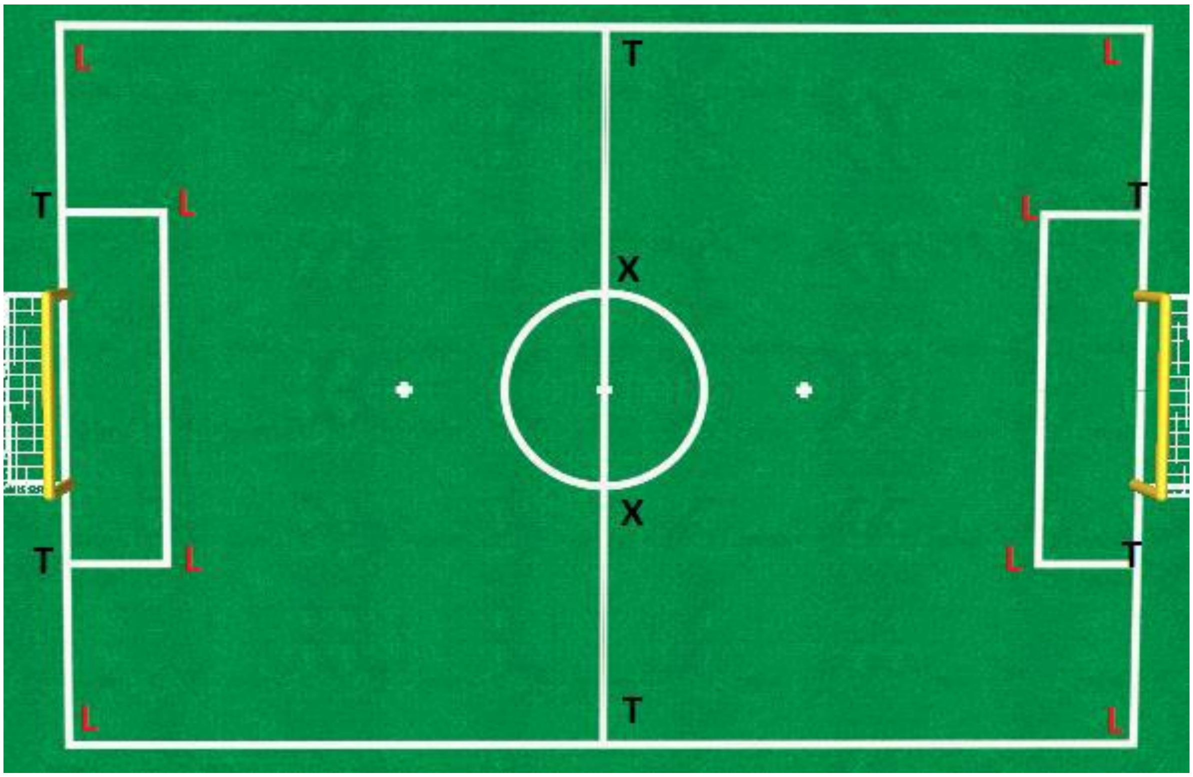


Figure 11 *Field Corners in Robot Field*

But before these feature points can be used they must be uniquely related to a specific land mark. For this simple heuristic is used.

Based on the current pose of the robot we predict which land mark is most plausible i-e generates least error and thus we choose that land mark to be associated with given feature point.

We project the given feature point onto world coordinates using this projected point and the detected type of the feature point we measure the distance between all the possibilities and the project point then we choose the one with least distance i-e least error.

4.2.3 Camera Model Calibration and Correction

For camera calibration we used checker board pattern and algorithms from opencv to determine the focal length for pinhole camera model.

4.2.4 Sensor Model

Once we take the measurements of local coordinates of each feature point and have mapped it to a unique land mark. We need to translate each reading to a probabilistic measure.

Given inputs:

- Local coordinates of feature points
- State/pose of each particle

Required Output:

- Probabilistic measure of correctness of the given state

Adopted procedure:

We convert the mapped landmark world location to local coordinates according to the given pose then

Then probability of correctness is given by:

$p1=(0,1-x2/x1,variance)*gauss(0,(y1-y2)/x1,variance)$

$where\ gauss\ is\ gaussian\ function\ with$

$mean=0$

$x2=x\ local\ coordinate\ of\ the\ observed\ land\ mark$

$x1=x\ local\ coordinate\ of\ the\ mapped\ land\ mark$

$variance=varaince\ of\ gaussian\ function$

4.2.5 Odometry Model

Its purpose is to update the particle state according to the motion of the robot. For odometric measurement we use the naoqi api for robot position using this we determine the delta in the movement and use them to update the state of the particle with some added Gaussian noise.

4.2.6 Particle Filter

We have used augmented particle filter approach as mentioned in [1]. The algorithm is given below:

$Algorithm\_Augmented\_MCL(X_{t-1}, u_t, z_t, e)$: $X_t'(\emptyset), X_t(\emptyset)$
    static w_{slow}, w_{fast}
    for $m = 1$ to M do
        $x_t^{[m]} = motion\_update(u_t, x_{t-1}^{[m]})$
        $w_t^{[m]} = sensor\_update(z_t, x_t^{[m]}, e)$
        $X_t' = X_t' + (x_t^{[m]}, w_t^{[m]})$
        $w_{avg} = w_{avg} + M^{-1} w_t^{[m]}$
    $w_{slow} = w_{slow} + \alpha_{slow}(w_{avg} - w_{slow})$
    $w_{fast} = w_{fast} + \alpha_{fast}(w_{avg} - w_{fast})$
    for $m = 1$ to M do
        with probability $\max\{0, 1 - w_{fast} / w_{slow}\}$ do
            add new pose to X_t
        else
            draw $i \in \{1,\ldots,N\}$ with probability $\propto w_t^{[i]}$
            add $x_t^{[i]}$ to X_t
return X_t

This extension of the base particle filter algorithm keeps track of the overall weighting of the distribution over time.

4.2.7 Inverse Perspective Transform

As it can be seen in the image below it is very difficult to detect feature points in the original image.

Figure 12 *Image before inverse perspective transform*

Therefore we applied the inverse perspective transform to change the perspective of the image, resulting into the image given below:

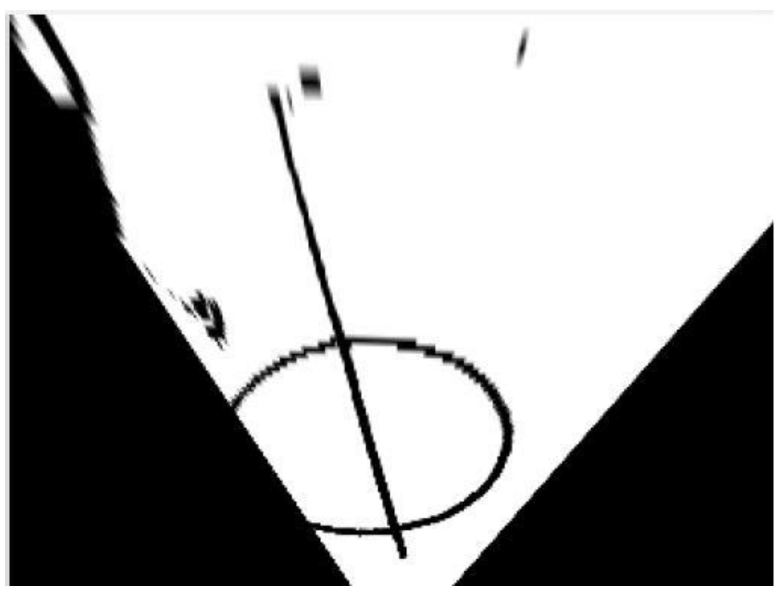


Figure 13 *Image after inverse perspective transform*

This allows us to detect the feature points very easily as can be seen above. To make this transformation we take four corner points of the original image and then find their local coordinates. Then these local coordinates are rotated to adjust in the coordinate frame of the image. Then they are scaled to fill the dimensions of the original image. Now we have two set of coordinate and we need to find a perspective transformation that maps one onto and other.

4.3 DEFENCE

Three different motions were created for the Nao humanoid goal keeper robot. These motions are designed to tackle different ranges. These motions are described as follows:

4.3.1 Full body dive

This motion is designed to cover as large distance as possible .In this motion full length of Nao's body is used

4.3.2 Planting on ground

In this motion, the robot falls down on its hips and plants his self in front of goal to stop the ball. This will be a medium range motion.

4.3.3 Squatting

This is a non-impact motion. In this motion the robot remains on its feet and suat in front of goal. This will be a short range motion.

Chapter 5

GOALKEEPER BEHAVIOUR

5.1 ARCHITECTURE

A goalkeeper's behavior is implemented using different components, each one providing different feature of a goalkeeper. For instance, there is a component that is used for ball detection another for defending the goal etc.

Components are organized into different level or hierarchies to implement behaviors of a goalkeeper. Tasks that require several components are triggered in a hierarchical tree. Every non-active component can be activated through the request of an active component.

5.1.1 States and Transitions

Each component consists of states and transitions that help the states to switch between each other.

At a time only one state can be active. Every time a state is executed the same code of the active state will be executed .Transitions from the currently active state are predicates that returns Boolean value to switch to another state. State diagram for a goalkeeper's behavior is shown in figure 5.1

5.1.1.1 Pseudo Code for Sate Transitions

$[CurrentState\ StateTransition]$⟹NewState

- $[\quad - \quad , \ start \quad]$⟹ [SeekingGoal]
- $[SeekingGoal\ , \ goalFound]$⟹ [Positioning]
- $[Positioning, \ inPosition]$⟹ [SeekingBall]
- $[Positioning\ , goalLost]$⟹ [SeekingGoal]
- $[SeekingBall, \ ballFound]$⟹ [Defending]
- $[Defending\ , \ ballAtFeet]$⟹ [ClearingBall]

- $[Defending\ ,\ ballNear] \Longrightarrow$ [MovingToTheBall]
- $[Defending\ ,\ ballLost] \Longrightarrow$ [SeekingBall]
- $[\text{MovingToTheBall}\ ,\ \ ballNear] \Longrightarrow$ [ClearingBall]
- $[\text{MovingToTheBall}\ ,ballLost] \Longrightarrow$ [SeekingBall]
- $[\text{ClearingBall}\ ,\ ballCleared] \Longrightarrow$ [SeekingGoal]

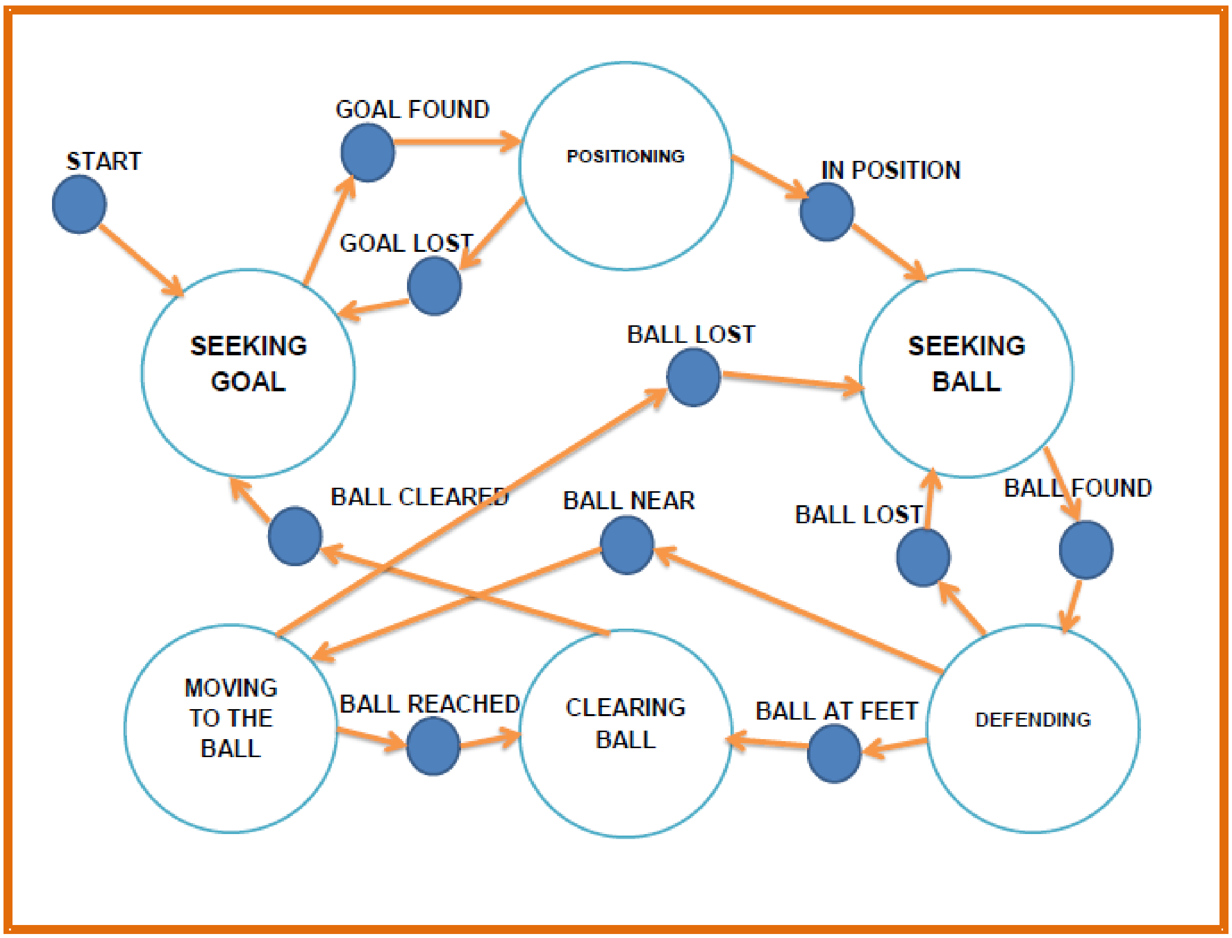


Figure 14 *State Diagram for a Goal Keeper Behavior*

States Dependencies

- Task related staes are grouped into same controller i.e

Goal Detector

- Positioning
- Seeking Goal

Searcher

- Seeking Goal
- Positioning
- Seeking Ball
- Moving to the ball
- Defending
- Clearing Ball

Ball Detector

- Seeking Ball
- Moving to the Ball
- Defending

5.2 DECISION MAKING FOR A GOALKEEPER

A goalkeeper has a simple purpose i.e., to defend the goal from the opponent kicks, but it needs to exhibit a richer behavior by showing strong coordination between its different actions in order to have an important role in the team.

Goalkeeper robot has a significant role in the league of humanoid so, its wrong decision and incorrect jump can cause changes in the result of a competition.

The role of goalkeeper robot in the game is vital as a good goalkeeper in the team can save lots of balls. For saving the goal the goalkeeper first determine the direction of ball then jumps toward it and tries to maintain its position for a while in order to make sure that the goal is saved. After that it gets up and throws the ball away and then moves towards the center of its goal using localization. Steps required for saving the goal are shown in figure 5.2

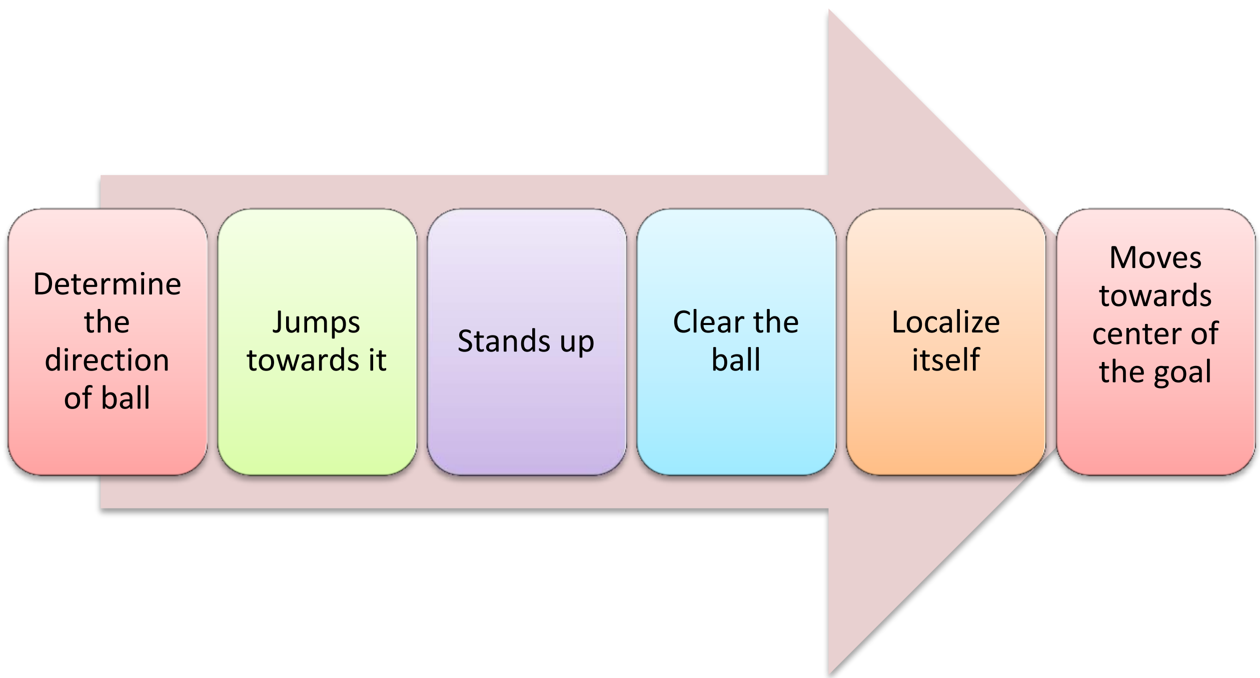


Figure 15 *Steps required for saving the goal*

Above mentioned steps require a substantial amount of time which Humanoid robots lack due to currently low speed. So if the robot makes an incorrect decision and jumps without any ball approaching the goal or jumps in the wrong direction, it will lose considerable time of the game.

5.2.1 Heuristic Based Decision Making

Taking right decision, in limited time, for a goalkeeper is of significant importance as it plays a vital role in winning a game. Heuristic based decision-making is used for goalkeeper that uses following heuristics for making a right decision:

- Possession of ball
- Ball coordinates
- Ball speed
- Point at which ball collides with goal
- Time it hits the goal

5.2.1.1 Possession of ball

An important feature of humanoid robots is that they are autonomous. This means that they have to make all decisions by themselves so using low powered CPUs, it difficult for robots to have high level of image processing. So currently object detection in the field is done by using different colors. For this purpose each object on the field is given a specific color for differentiation.

To reduce and minimize the dependency of object detection on the light of surrounding environment HSI space is used.

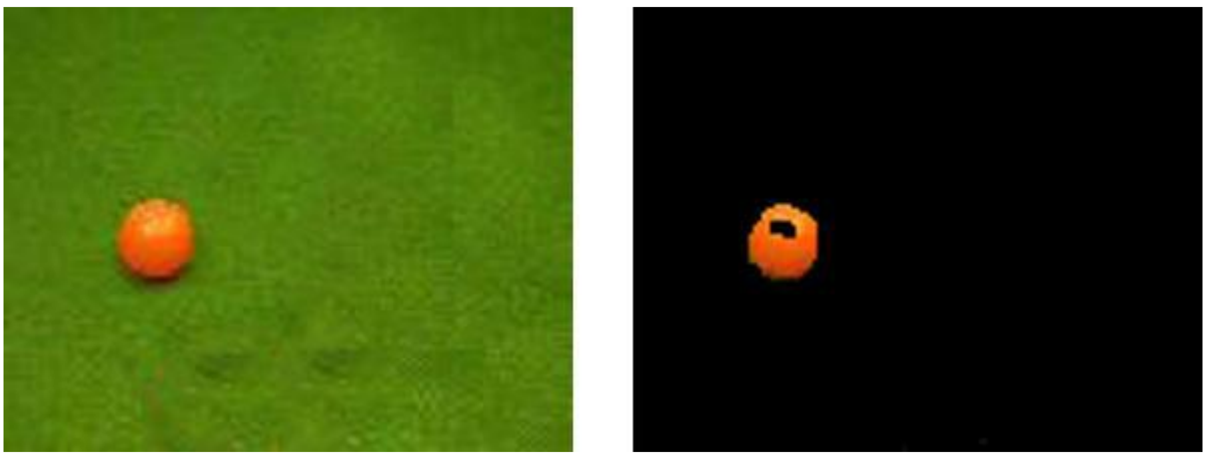


Figure 16 *Left: Image of the ball on the ground, taken by the robot camera. Right: the processed image by Ball Detection Algorithm*

5.2.1.2 Determining Coordinates of Ball

As the camera is along the ball, coordinates of the ball could be determined through the angle of horizontal and vertical motors. For this, we assume that the robot is in the coordinates (0, 0). The distance of ball from the goal, in the *Y* axis can be obtained as follows:

$$Y = H \tan(\theta_v) \qquad (1)$$

In Figure 5.2.2, *H* is the height of robot, *Y gives* the distance of from the goal in the *Y* axis and θ_v is the angle of vertical motor.

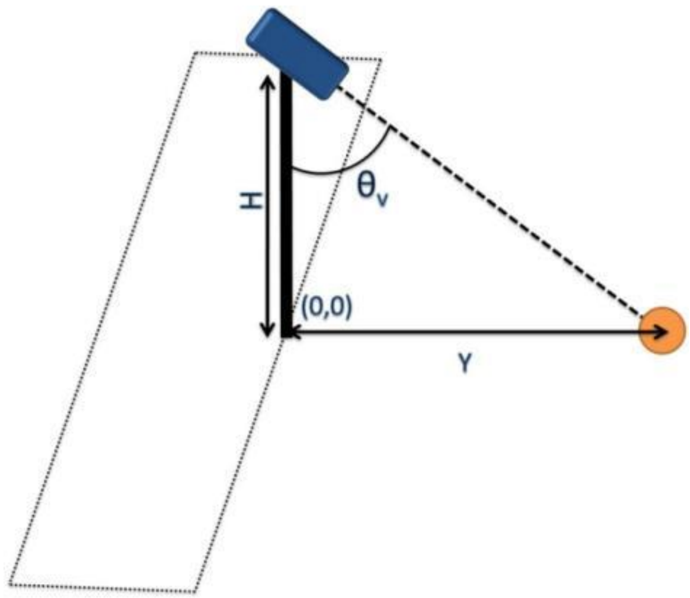


Figure 17 *Side view of the robot*

Figure 5.2.3 shows the upper view of robot where θ_h is the horizontal angle of the motor. In this figure horizontal distance of the ball from the robot is shown by X and the angle of horizontal motor is represented by θ_h . Figure 5.2.4 exhibits lateral view where θ_h is the horizontal motor angle is and the vertical angle is θ_v. Thus the coordinate (x, y) of the ball can be calculated at any time.

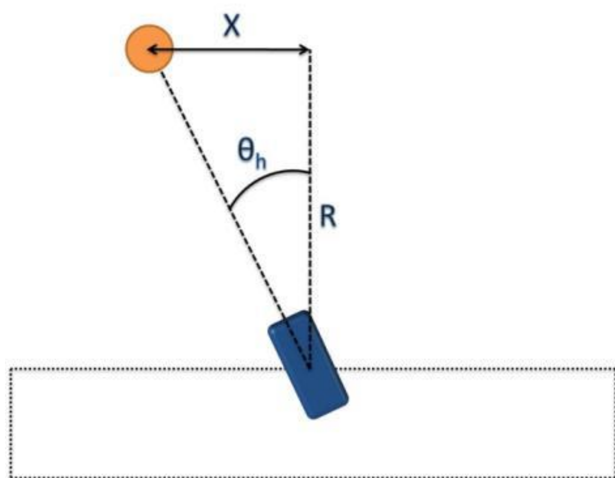


Figure 18 *Upper view of robot*

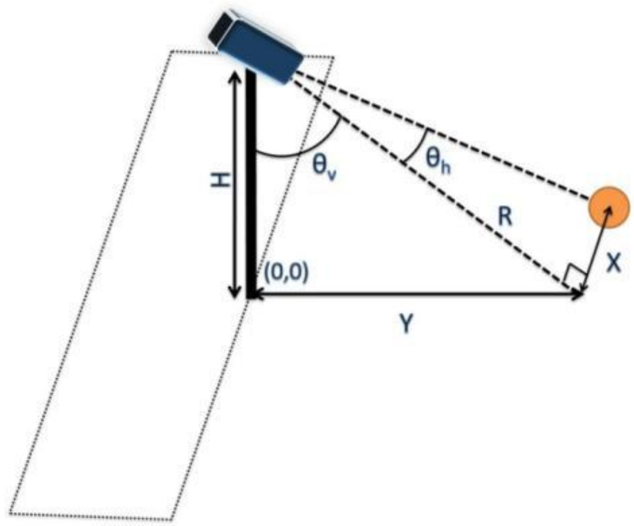


Figure 19 *Side view of the robot*

Following relations are used to calculate the horizontal distance of the ball from the robot:

$$X = R\tan(\theta_h) \qquad (2)$$

$$R = \frac{H}{cos\,(\theta_v)}$$

Therefore:

$$X = \frac{H\,tan\,(\theta_h)}{cos\,(\theta_v)} \qquad (3)$$

5.2.1.3 Ball Speed and Acceleration

The next important heuristics used are speed and acceleration of the ball movement. They are calculated using relative values. Speed is computed by using two points whereas acceleration requires three points for its calculation, which is calculated as below:

$$V_y = Y_{old} - Y_{new} \qquad (4)$$

$$a_y = V_{old} - V_{new} \qquad (5)$$

5.2.1.4 Collision Point of Ball with Goal

This heuristic can help the goalkeeper to choose most appropriate jump amongst different predefined type of jumps. Each new image frame obtains the new coordinates of the ball. Linear equation that the ball moves on can be gained by using the previous coordinates and the new coordinates of the ball.

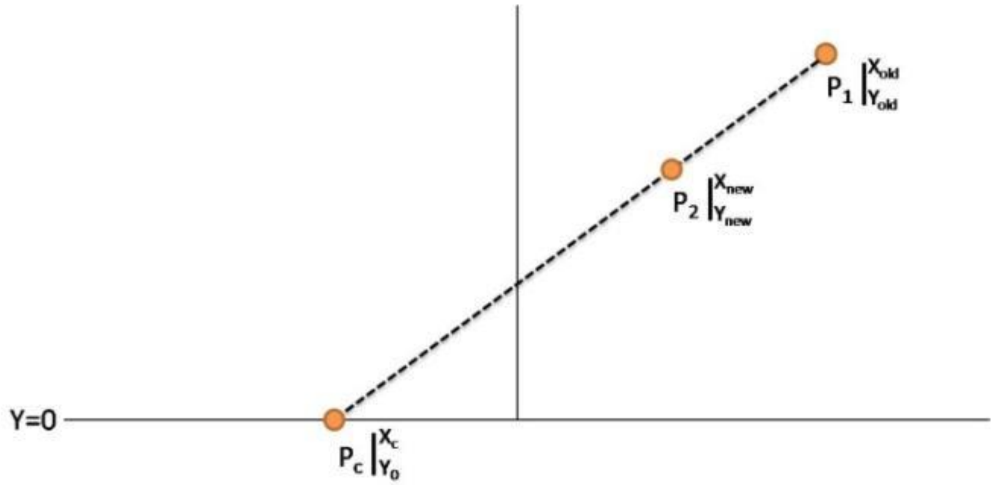


Figure 20 *Linear relationship for movement of ball*

Calculating the coordinates of the ball in two points P1 and P2 the collision point P_c can be calculated.

By crossing the acquired line with goal line (where y = 0) and using below equations collision point of the ball with the goal can be determined.

$$Y - Y_{new} = m(x - x_{new});$$

$$m = \frac{y_{old} - y_{new}}{x_{old} - x_{new}} \qquad (6)$$

$$x_{contact} = \frac{-y_{new}}{m} + x_{new} = \frac{(x_{new} - x_{old})y_{new}}{y_{new} - y_{old}} + x_{new} \qquad (7)$$

5.2.1.5 Time it hits the goal

If the ball distance from the goal is less than the distance that ball transverse before stopping, the ball will hit the goal and the goalkeeper should attempt to save the goal.

Chapter 6

IMPLEMENTATION AND RESULT DISCUSSION

6.1 VISION

6.1.1 Implementation

All of the algorithms are implemented in OpenCV and C++. Also remote modules are used for running the modules on PCs, instead directly running them on the robots processor.

6.1.2 Results

The results are discussed below:

6.1.2.1 Ball detection

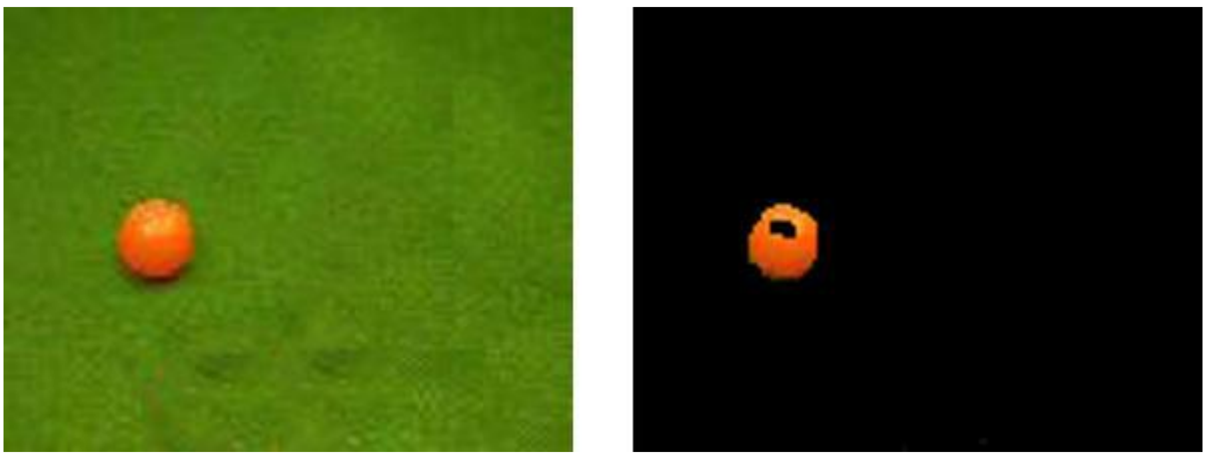


Figure 21 *Results of Ball Detection*

6.1.2.2 Corners detection

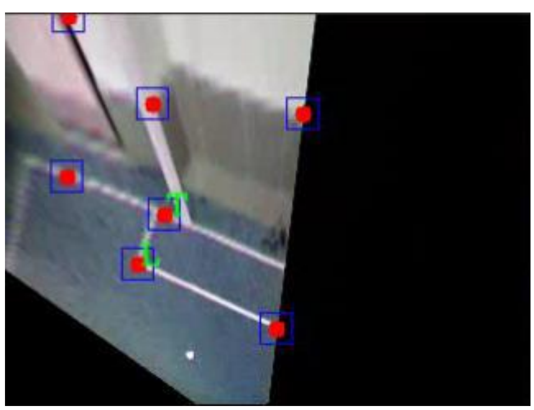


Figure 22 *Results of Corner Detection*

6.1.2.3 Field extraction

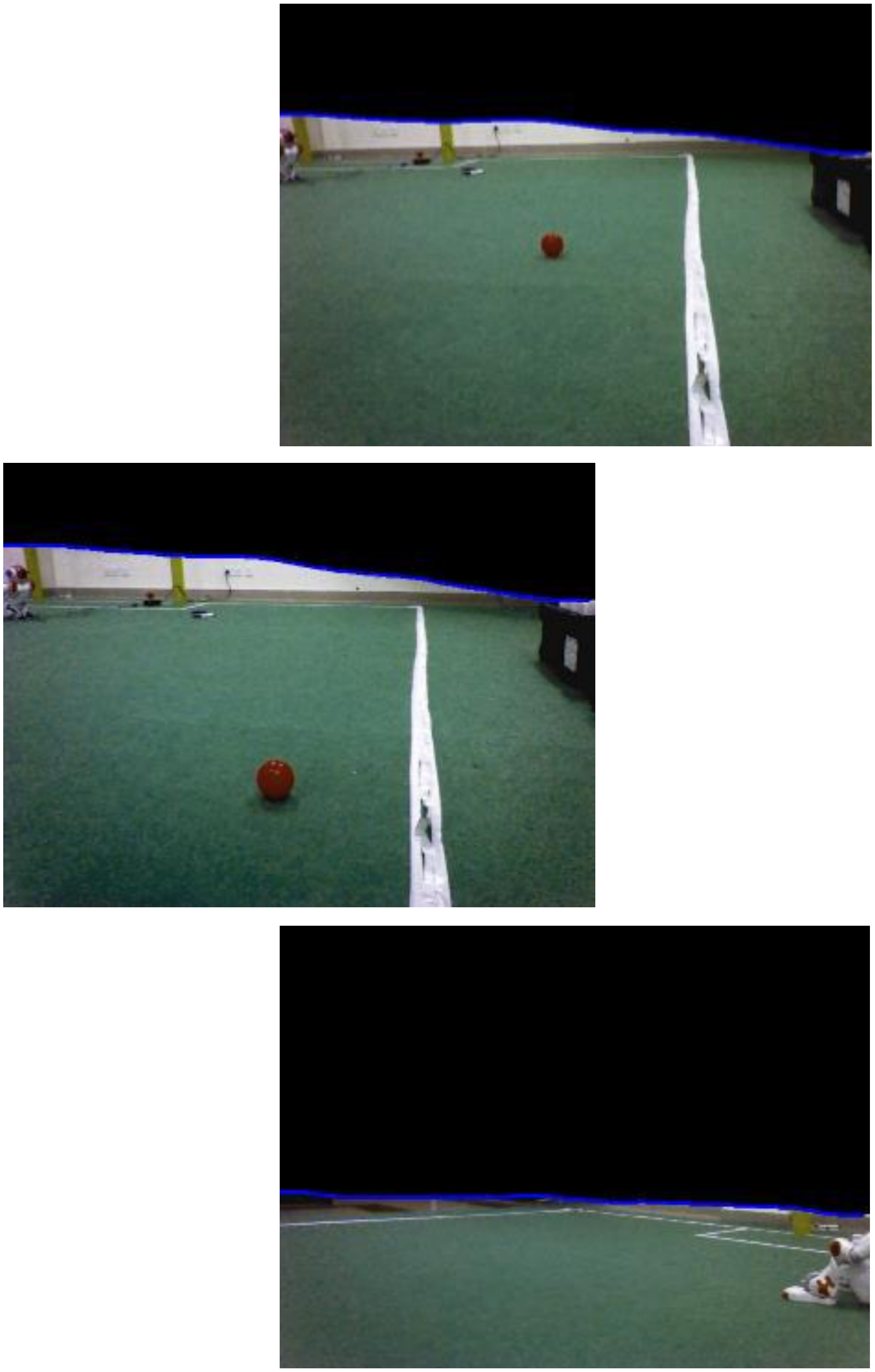


Figure 23 *Results of field extraction*

6.2 LOCALIZATION

6.2.1 Implementation

Modules have been used for implementation of localization. Each module has its own entity as a class.

6.2.2 Results:

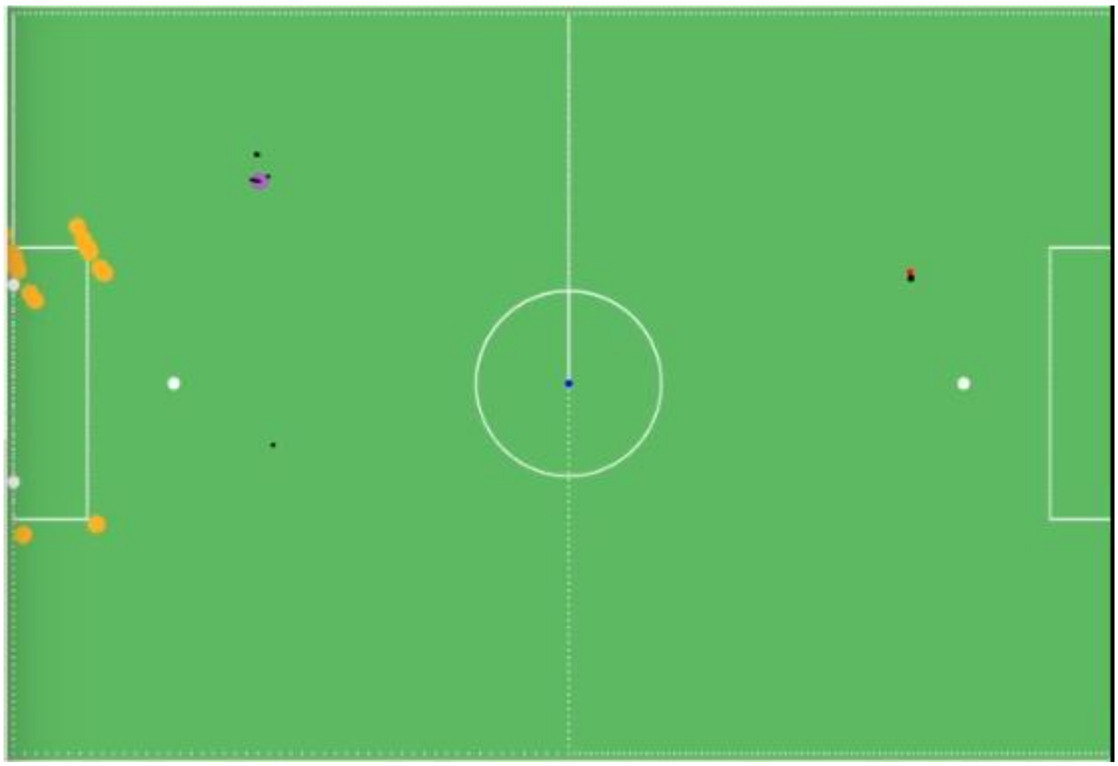


Figure 24 *Localization results*

6.3 DECISION MAKING

6.3.1 Implementation

A goalkeeper's behavior is implemented using different components, each one providing different feature of a goalkeeper.

Each component is made up of states and transitions that help the states to switch between each other, using a Boolean return value.

6.3.2 Results:

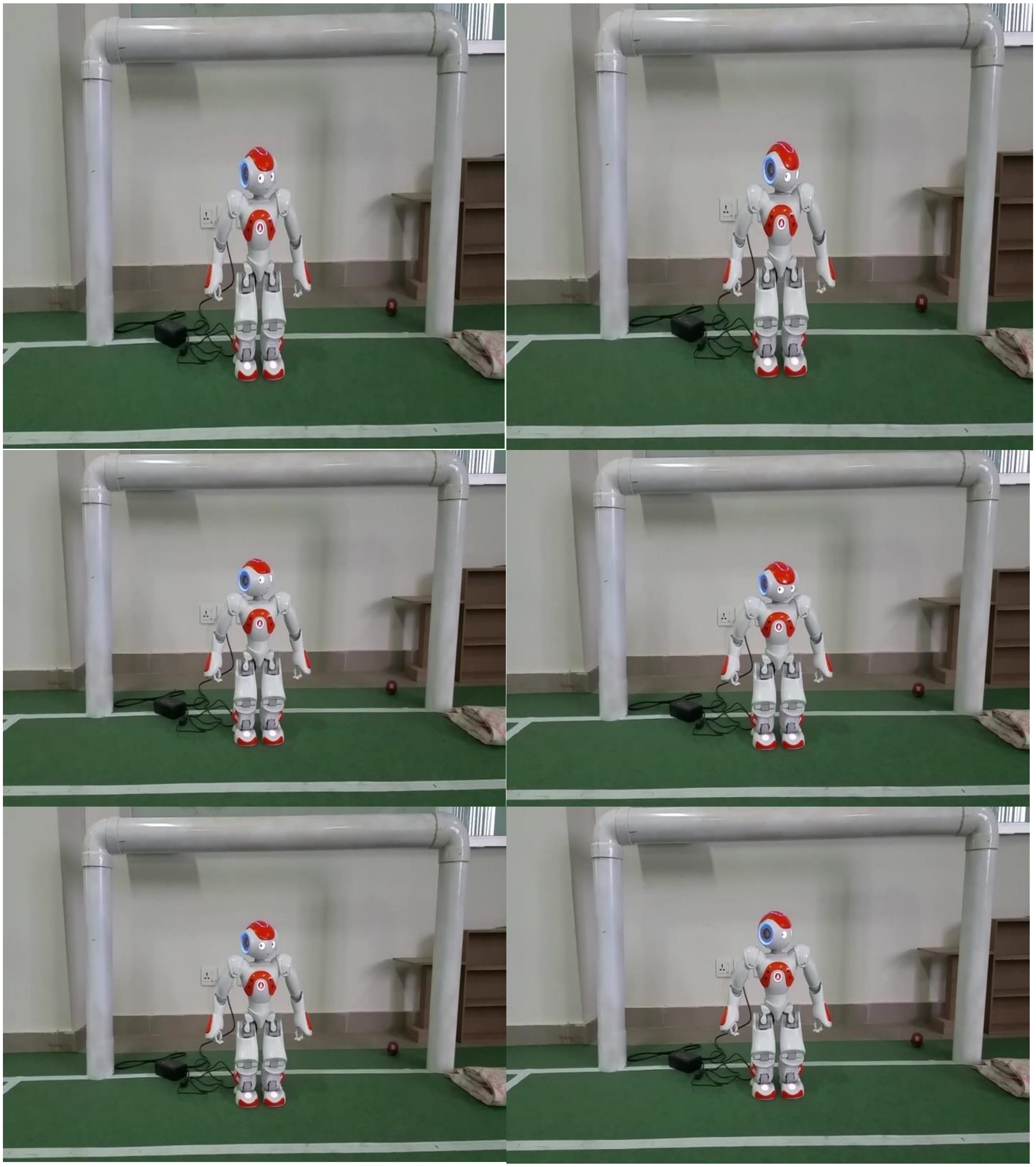


Figure 25 Seeking Ball State

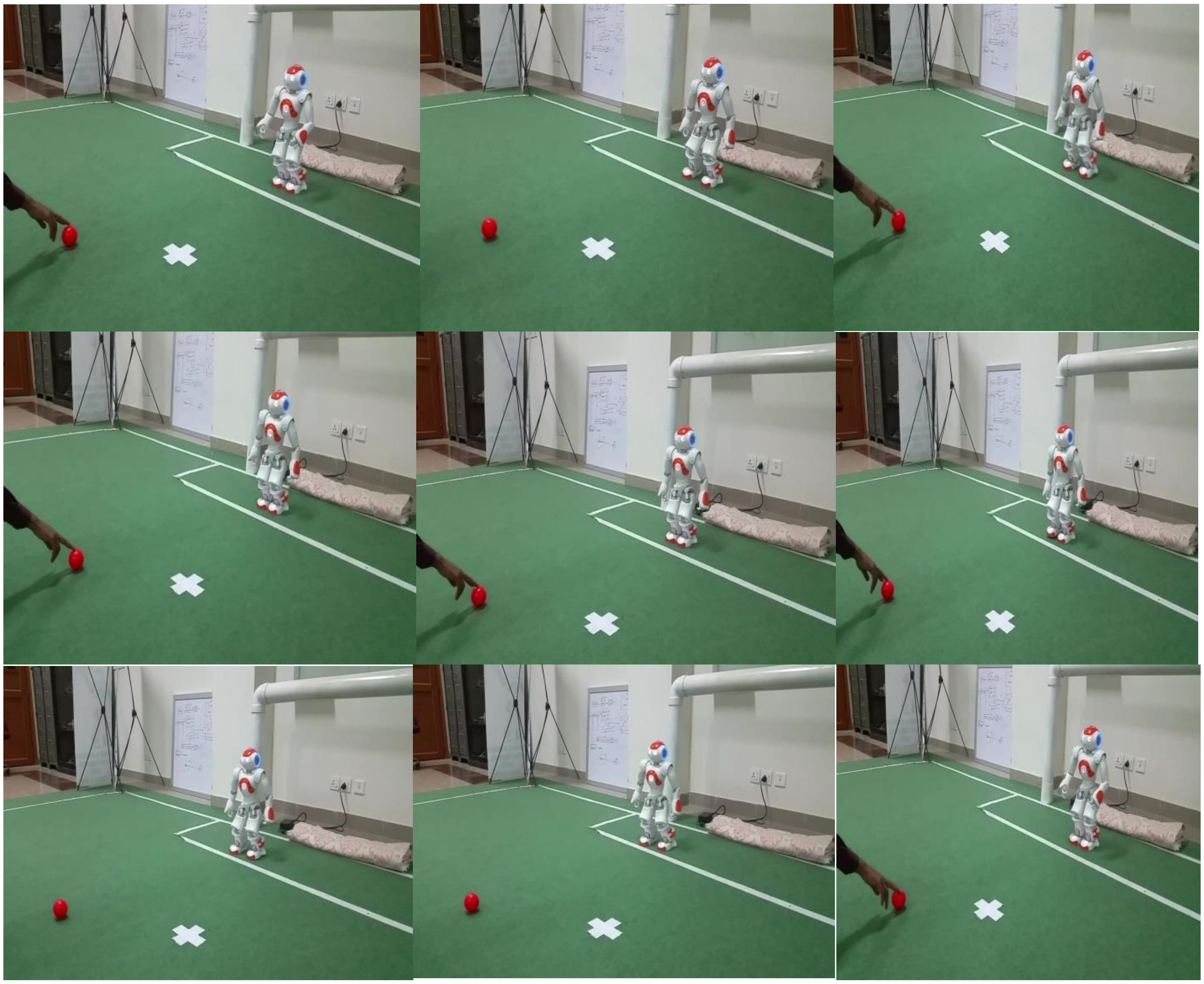
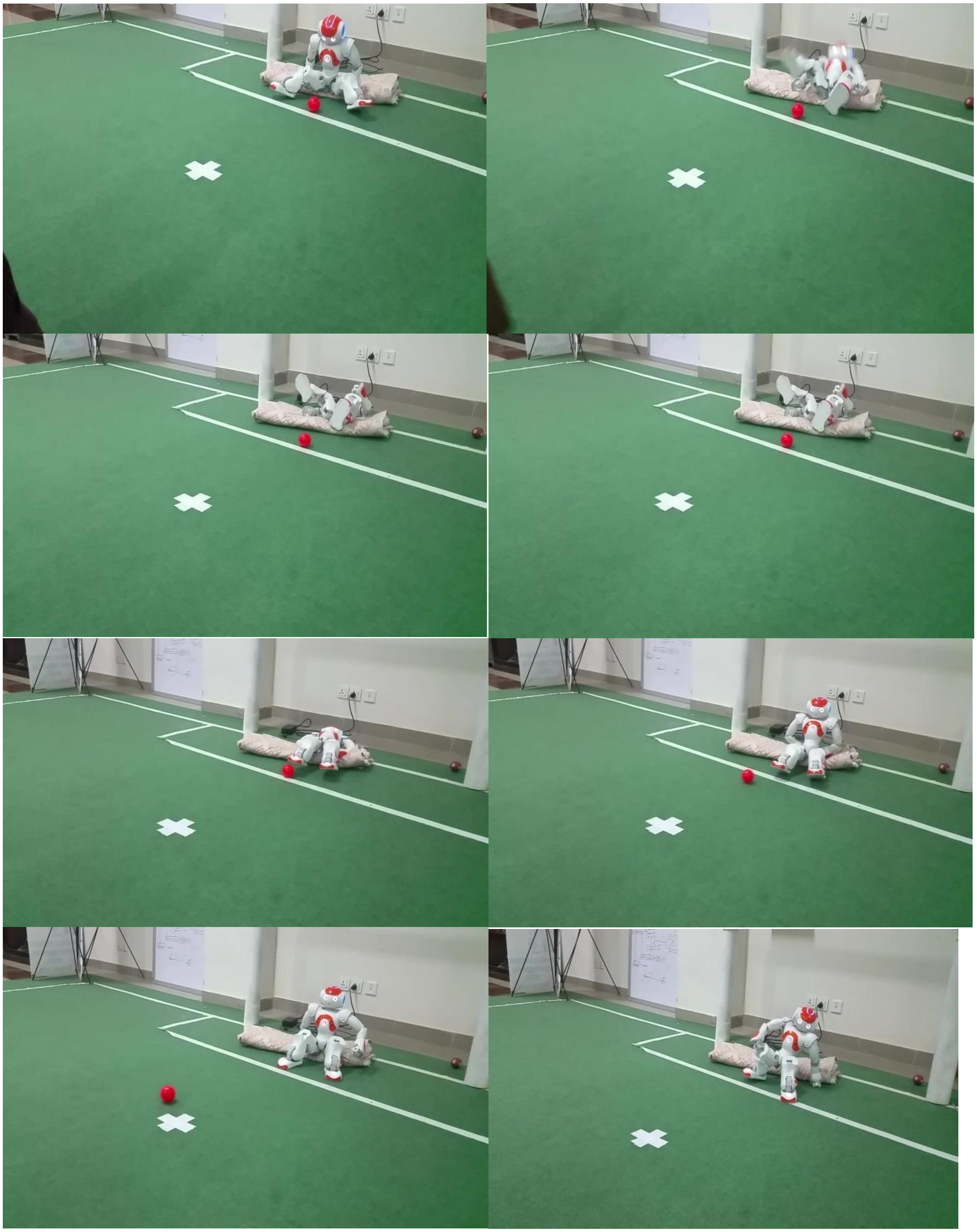


Figure 26 *Positioning*



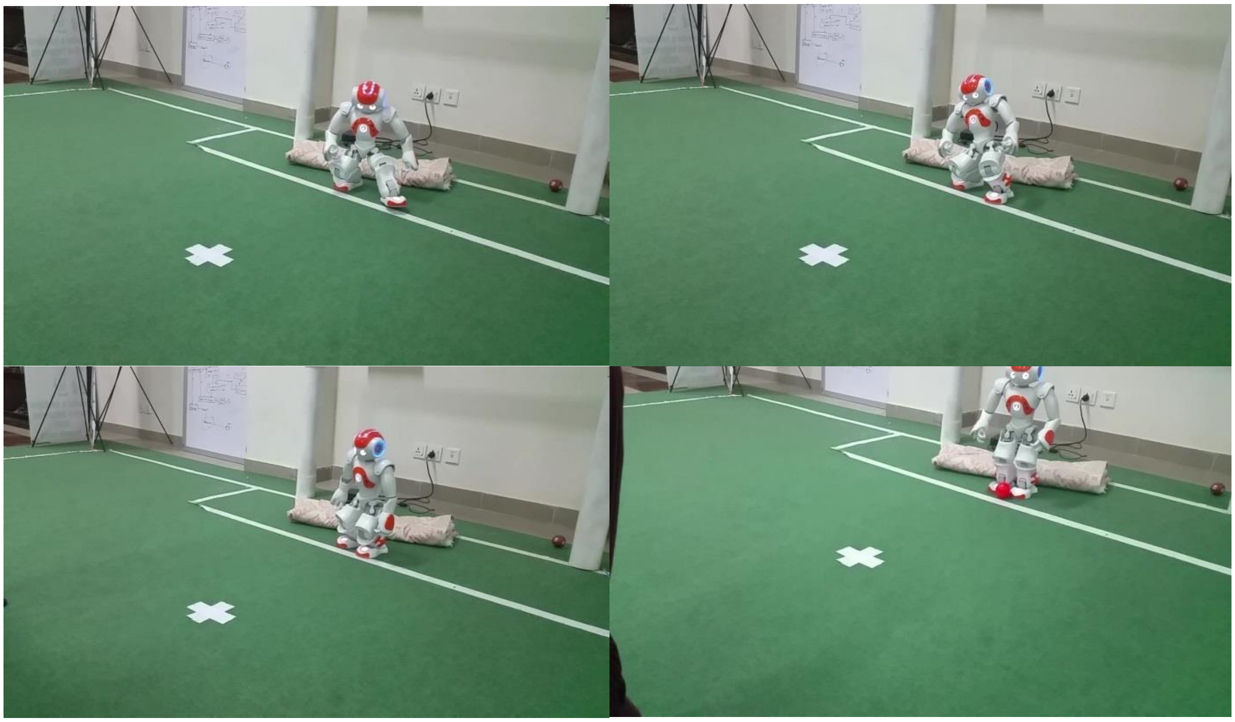


Figure 27 *Defending*

Chapter 7

CONCLUSION AND FUTURE RECOMMENDATIONS

In this research a goalkeeper for Robocup SPL is developed and a heuristic based decision making approach is being presented to improve its performance. Heuristics used for its decision making includes detection of the ball on the ground, finding out its coordinates, speed and direction of the ball and finally calculating when and where the ball hits the goal.

Some of the suggestions for the future work are as below:

- There are still some immaturities in the vision module of this project. The results sometimes give false positives which really mess up the localization results. Some algorithms needs to be implemented to make Vision results robust and more accurate.

- Localization needs improvement in developing new methods to allow only true positives to be taken up as readings to in cooperate into the filter. One such method is to use optical flow. Other improvements include developing sensor models for lines and while goal post. Besides this it requires improvement in cooperating sensor models for using information sent from other robots.

- The architecture can be developed further to incorporate a layer to improve perception, making it more robust. Furthermore, deployment of code and debugging for multiple robots is an area which will need future attention.

- Decision making depends upon finding and chasing the ball. If the motor in the head of the robot could be fast enough and the imaging and image processing rate be adequate we can highly trust on the correct results.

BIBLIOGRAPHY

[1] Sridharan, M., Kuhlmann, G., Stone, P.: Practical vision-based Monte Carlo localization on a legged robot. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA) (2005).

[2] Minakata, Hayashibara, H., Ichizawa, Y., Horiuchi, K., Fukuta, T., Fujita, M., Kaminaga, S., Irje, H., Sakamoto, K., Chiba, H.:A method of single camera robocup humanoid robot localization using cooperation with walking control. In: 10th IEEE International Workshop, pp. 5--55. Trento (2008).

[3] Ruiz-del-Solar, J., Palma-Amestoya, R., Marchant, R., Parra-Tsunekawa, I., Zegers, P.: Learning to fall: Designing low damage fall sequences for humanoid soccer robots. In: science, Robotics and Autonomous Systems, Vol. 57, No. 8, pp. 796--807, (2009).

[4] Ruiz-del-Solar, J., Palma-Amestoy, R., Vallejos, P., Marchant, R., Zegers, P.: Designing Fall Sequences That Minimize Robot Damage in Robot Soccer. In: RoboCup 2008, LNCS, vol 5399, pp. 271—283. Springer, Berlin / Heidelberg (2009).

[5] Ude, A., Shibata, T., Atkeson, C.G.: Real-time visual system for interaction with a humanoid robot. In: Robotics and Autonomous Systems, Vol. 37, pp. 115--125, (2001). International Journal of Advanced Science and Technology Vol. 34, September, 2011 26

[6] Wilken, T., Missura, M., Behnke, S.: Designing falling motions for a humanoid soccer goalie. In: a workshop of the 2009 IEEE-RAS Intl. Conf. On Humanoid Robots (Humanoids 2009), pp. 79-84, France (2009).

[7]. Adorni, G., Cagnoni, S., Mordonini, M.: Landmark-based robot self-localization: a case study for the RoboCup goal-keeper. In: Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on. pp. 164{171. IEEE (1999)

[8] Birbach, O., Frese, U., B□auml, B.: Realtime Perception for Catching a Flying Ball with a Mobile Humanoid. In: 2011 IEEE International Conference on Robotics and Automation. pp. 5955{5962 (2011)

[9] Bozinovski, S., J□ager, H., Sch□ol, P.: Engineering goalkeeper behavior using an emotion learning method. In: Proceedings of the RoboCup Workshop, KI99: Deutsche Jahrestagung f□ur K□unstliche Intelligenz. pp. 48{56 (1999)

[10] Curtin, R.R., Cline, J.R., Slagle, N.P., March, W.B., Ram, P., Mehta, N.A., Gray, A.G.: MLPACK: A scalable C++ machine learning library. Journal of Machine Learning Research 14, 801{805 (2013)

[11] Diaz, G.J., Fajen, B.R., Phillips, F.: Anticipation from biological motion: the goalkeeper problem. Journal of experimental psychology: human perception and performance 38(4), 848 (2012)
[12] Faragher, R., et al.: Understanding the basis of the Kalman _lter via a simple and intuitive derivation. IEEE Signal processing magazine 29(5), 128{132 (2012)
[13] Garc__a, J.F., Rodr__guez, F.J., Fern_andez, C., Matell_an, V.: Design an evaluation of RoboCup humanoid goalie. Journal of Physical Agents 4(2), 19{26 (2010)
[14] King, D.E.: Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research 10, 1755{1758 (2009)
[15] Lausen, H., Nielsen, J., Nielsen, M., Lima, P.: Model and behavior-based robotic goalkeeper. In: RoboCup 2003: Robot SoccerWorld Cup VII, pp. 169{180. Springer (2004)
[16] Liu, J., Liang, Z., Shen, P., Hao, Y., Zhao, H.: The Walking Skill of Apollo3D{ The Champion Team in the RoboCup2013 3D Soccer Simulation Competition. In: RoboCup 2013: Robot World Cup XVII, pp. 104{113. Springer (2014)
[17] MacAlpine, P., Barrett, S., Urieli, D., Vu, V., Stone, P.: Design and Optimization of an Omnidirectional Humanoid Walk: A Winning Approach at the RoboCup 2011 3D Simulation Competition. In: AAAI (2012)
[18] MacAlpine, P., Collins, N., Lopez-Mobilia, A., Stone, P.: UT Austin Villa: RoboCup 2012 3D simulation league champion. In: RoboCup 2012: Robot Soccer World Cup XVI, pp. 77{88. Springer (2013)
[19] Mackworth A (1993) Computer vision: system, theory, and applications, Chapter 1. World Scientific Press, Singapore
[20] Chiou J, Wang K (2008) Application of a hybrid controller to a mobile robot. Simul Model Pract Theory 16(7):783–795
[21] Kim J (1997) Special issue about the first micro-robot world cup soccer tournament, mirosot. Robot Auton Syst 21:137–205
[22] Matsubara H, Asada M, Kitano H (2002) History of RoboCup and prospects for RoboCup-2002. J Robot Soc Jpn 20(1):2–6

[23] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in International Conference on Computer Vision & Pattern Recognition, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, June 2005, pp. 886–893.

[24] T. Rofer, T. Laue, J. M¨uller, A. Burchardt, E. Damrose, A. Fabisch, F. Feldpausch, K. Gillmann, C. Graf, T. J. de Haas, A. H¨artl, D. Honsel, P. Kastner, T. Kastner, B. Markowsky, M. Mester, J. Peter, O. J. L. Riemann, M. Ring, W. Sauerland, A. Schreck, I. Sieverdingbeck, F. Wenk, and J.-H. Worch, “B-Human Team Report and Code Release 2010,” 2010, only available online: http://www.b-human.de/file download/33/ bhuman10 coderelease.pdf.

[25] H. P. Moravec and A. Elfes, “High resolution from wide angle sonar,” in Proceedings of the 1985 IEEE International Conference on Robotics and Automation, vol. 2, St. Louis, MO, USA, 1985, pp. 116–121.

[26] S. Lenser and M. Veloso, “Visual sonar: Fast obstacle avoidance using monocular vision,” in Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, NV, USA, 2003.

[27] J. Hoffmann, M. J¨ungel, and M. L¨otzsch, “A vision based system for goal-directed obstacle avoidance,” in RoboCup 2004: Robot Soccer World Cup VIII, ser. Lecture Notes in Artificial Intelligence, D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, Eds., vol. 3276. Springer, 2005, pp. 418–425.

[28] D. Wilking and T. R¨ofer, “Real-time object recognition using decision tree learning,” in RoboCup 2004: Robot Soccer World Cup VIII, ser. Lecture Notes in Artificial Intelligence, D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, Eds., vol. 3276. Springer, 2005, pp. 556–563.

[29] T. Laue and T. R¨ofer, “Integrating simple unreliable perceptions for accurate robot modeling in the four-legged league,” in RoboCup 2006: Robot Soccer World Cup X, ser. Lecture Notes in Artificial Intelligence, G. Lakemeyer, E. Sklar, D. G. Sorrenti, and T. Takahashi, Eds., vol. 4434. Springer, 2007, pp. 474–482.

[30] U. Kaufmann, G. Mayer, G. Kraetzschmar, and G. Palm, “Visual robot detection in robocup using neural networks,” in RoboCup 2004: Robot Soccer World Cup VIII, ser. Lecture Notes in Artificial Intelligence, D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, Eds., vol. 3276. Springer, 2005, pp. 262–273.

[31] Reid, S. ; Goodwin, R.; Haigh, K.; Koenig, S.; O'Sullivan, J.; Veloso, M. (1997). Xavier: Experience with a Layered Robot Architecture. Agents '97, 1997.

[32] Stoytchev, A.; Arkin, R. (2000). Combining Deliberation, Reactivity, and Motivation in the Context of a Behavior-Based Robot Architecture. In Proceedings 2001 IEEE International

Symposium on Computational Intelligence in Robotics and Automation. 290-295. Banff, Alberta, Canada. 2000.

[33] Arkin, R. (1989). Motor Schema Based Mobile Robot Navigation. The International Journal of Robotics Research, Vol. 8, No. 4, 92-112 (1989).

[34] Calvo, R.; Cañas, J.M.; García-Pérez, L. (2005). Person following behavior generated with JDE schema hierarchy. ICINCO 2nd Int. Conf. on Informatics in Control, Automation and Robotics. Barcelona (Spain), sep 14-17, 2005. INSTICC Press, pp 463-466, 2005. ISBN: 972-8865-30-9.

[35] Cañaas, J. M.; and Matellán, V. (2007). From bio-inspired vs. psycho-inspired to etho-inspired robots. Robotics and Autonomous Systems, Volume 55, pp 841-850, 2007. ISSN 0921- 8890.

[36] Lenser, S.; Bruce, J.; Veloso, M. (2002). A Modular Hierarchical Behavior-Based Architecture, Lecture Notes in Computer Science. RoboCup 2001: Robot Soccer World Cup V. pp. 79-99. Springer Berlin / Heidelberg, 2002.

[37] Saffiotti, A. ; Wasik, Z. (2003). Using hierarchical fuzzy behaviors in the RoboCup domain. Autonomous robotic systems: soft computing and hard computing methodologies and applications. pp. 235-262. Physica-Verlag GmbH. Heidelberg, Germany, 2003.

[38] Gómez, A.; Martínez, H.; (1997). Fuzzy Logic Based Intelligent Agents for Reactive Navigation in Autonomous Systems. Fitth International Conference on Fuzzy Theory and Technology, Raleigh (USA), 1997.

[39] Herrero, D. ; Martínez, H. (2008). Embedded Behavioral Control of Four-legged Robots. RoboCup Symposium 2008. Suzhou (China), 2008.

[40] Loetzsch, M.; Risler, M.; Jungel, M. (2006). XABSL - A pragmatic approach to behavior engineering. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006), pages 5124-5129, Beijing, October 2006.

[41] Akin, H.L.; Meriçli, Ç.; Meriçli, T.; Gökçe, B.; Özkucur, E.; Kavakhoglu, C.; Yildiz, O.T. (2008). Cerberus'08 Team Report. Technical Report. Turkey, 2008.

[42] Chown, E.; Fishman, J.; Strom, J.; Slavov, G.; Hermans T.; Dunn, N.; Lawrence, A.; Morrison, J.; Krob, E. (2008). The Northern Bites 2008 Standard Platform Robot Team. Technical Report. USA, 2008.

[43] Phillips, M. & Veloso, M. (2009). Robust Support Role in Coordinated Two-Robot Soccer Attack, In: RoboCup 2008, LNAI 5399, Iocchi, L, Matsubara, H., Weitzenfeld, A. & Zhou, C., pp. 235-246, Springer-Verlag, 3-642-02920-5, Germany.

[44] Work, H.; Chown, E.; Hermans, T.; Butterfield, J. & McGranaghan, M. (2009). Player Positioning in the Four-Legged League, In: RoboCup 2008, LNAI 5399, Iocchi, L, Matsubara, H., Weitzenfeld, A. & Zhou, C., pp. 391-402, Springer-Verlag, 3-642- 02920-5, Germany.

[45] Zickler, S. & Veloso, M. (2009). Playing Creative Soccer: Randomized Behavioral Kinodynamic Flanning of Robot Tactics, In: RoboCup 2008, LNAI 5399, Iocchi, L, Matsubara, H., Weitzenfeld, A. & Zhou, C., pp. 414-425, Springer-Verlag, 3-642- 02920-5, Germany.

[46] Acosta Calderon, C.A.; Zhou, C.; Yue, P.K.; Wong, M. & Mohan, R.E. (2007). A Distributed Embedded Control Architecture for Humanoid Soccer Robots. Proceedings of Advances in Climbing and Walking Robots, pp. 487-496, Singapore, July 2007.

[47] Acosta Calderon, C.A.; Mohan, R.E. & Zhou, C. (2007). A Humanoid Robotic Simulator with Application to RoboCup, Proceedings of IEEE Latin American Robotic Simposium, Mexico, November 2007.

[48] Behnke, S.; & Stueckler, J. (2008). Hierarchical Reactive Control for Humanoid Soccer Robots. International Journal of Humanoid Robots, Vol 5, No 3, September, pp. 375-396.

[49] Coelho, A.L.V.; Weingaertner, D. & Gomide, F.A.C. (2001). Evolving Coordination Strategies in Simulated Robot Soccer. Proceeding of the 5th International Conference on Autonomous Agents, pp. 147-148, 1-58113-326-X, Montreal, Canada, May-June 2001.

[50] Friedmann, M.; Kiener, J.; Petters, S.; Thomas, D. & von Stryk, O. (2006). Modular Software Architecture for Teams of Cooperating, Heterogeneous Robots. Proceedings of IEEE International Conference on Robotics and Biomimetics, pp 24-29, Kunming, China, December 2006.

[51] Risler, M. & von Stryk, O. (2008). Formal Behavior Specification of Multi-robot Systems Using Hierarchical State Machines in XABSL. Proceedings of AAMAS08-Workshop on Formal Models and Methods for Multi-Robot Systems, Estoril, Portugal, May 2008.

[52] Halpern JY (2005) Reasoning about uncertainty. MIT Press, Cambridge.

[53] Fan B, Pu J, Liu G (2009) Multi-agent DecisionMaking Based on Evidence Reasoning. In: 2009 International joint conference on artificial intelligence. IEEE, pp 70–73.