Red Ball Kicking by NAO

Kaustubh Bhowmick[i], Prachi Purbia[ii], Rajat Arya[iii], Sachin Choubey[iv] and Dr. Tapas Badal[v]

***Abstract* - NAO, a humanoid robot is being extensively used at RoboCup, an international robotic soccer competition. Different universities around the world take part in it with their own NAO and their own algorithms to make NAO play soccer. In this paper, we have demonstrated the navigation system for the NAO that seeks to track a red ball, navigate to it, and kick it to the goal post. We have approached the problem in two ways - (1) one is similar to penalty kick and (2) another is kicking the ball kept anywhere (range of 1.5-2.0 m) in a frontal hemispherical plane with respect to the robot. The proposed method uses data acquired using the 2 cameras mounted on the robot to estimate the position of the ball with respect to the robot and then to navigate towards it. Thereafter, it will take image of the goal post using its upper camera and adjust its angular position with to the goal post. Finally, it will take image of the ball from its lower camera and adjust its position to kick the ball with the right or left leg whichever is nearer to the ball.**

***Index Terms* - RobotCup, Detection, Tracking, Kicking**

I. INTRODUCTION

Robotics is an ever-increasing division in Engineering and Computer Science. As we begin the 21st Century, technology is growing by leaps and bounds, and such exponential rates are often overwhelming [3]. The paper aims to align with those of the official RoboCup Federation, an international organization, registered in Switzerland, to organize international effort to promote science and technology using soccer games by robots and software agents [4].

Autonomous navigation and tracking require appropriate modelling of environment [1]. Remarkable progress has been done over the years and there is an ongoing research every year. For this project we utilize NAO, a complete humanoid robotics platform created by Aldebaran Robotics. NAO has 2 cameras mounted on his head which gives an enlarged field of vision, making it able to see in front of him or at his feet. His extensive capabilities due to the variety of sensors that he has plus its easy-to-use SDK makes him an excellent choice to fuse computer vision and motion dynamics into one single project. In this project, we will enable NAO to perform a low-level control of motion as well as use his cameras for streaming images and do a further processing of the video for tracking a target object. This concept has been widely used in a popular competition called RoboCup that happens every year, where robots play soccer among themselves [1].

The details of the NAO used in the project is as shown in the fig. 1. We used a polymer sheet for NAO to walk for secure movement of NAO for testing purpose. NAO will locate the ball, move to the ball, change walking speed as approaching objects, align and aim for kick, and kick the ball to the goal post. We are using a Red Ball, of 20 cm diameter and using a rectangular black box (50 cm X 40 cm) as goal post.

Figure 1: NAO Robot Details

II. RELATED WORK

As RoboCup takes place every year from 1998 (2008 with NAO) there is plethora of work going on in the direction of increasing the probability for a positive penalty. In [1] authors have used Reinforcement Learning to increase this to 85%.

Apart from its stability becomes a major player when balancing on one leg. The stability is classified into two types: -

A) Static

In static kicking there is no continues feedback to the body from camera. Once the decision is taken it is not reconsidered.[2] and [1] discuss keyframe based approach using Bezeir Curves which we have also followed.

B) Dynamic

Dynamic kicking involves taking PID (proportional integral differential) feedback controllers or using the Nao's Gyroscopic feedback to decide for itself whether the next keyframe arriving after the present one is stable or not. The papers describing this are [3] and [4].

The observations provided by the lightweight sensors which typically have to be used with humanoids are rather noisy and unreliable. As a result, accurate navigation, which is considered to be mainly solved for wheeled robots, is still a challenging problem for humanoid robots. Numerous papers are written every year addressing this problem, especially using NAO [5]. Based on these publications, we modified some ideas to create our own set of procedures which is explained in the next section.

III. METHODOLOGY

For NAO to perform any task it needs to be trained using Choregraphe software having various modules for various task. We used pynaoqi, pyhton-sdk having support for python libraries for connecting and performing certain tasks. We have provided two cases how NAO would kick the ball which are as follows:

(1) Penalty kick can be decomposed into the following steps (as shown in figure 2)

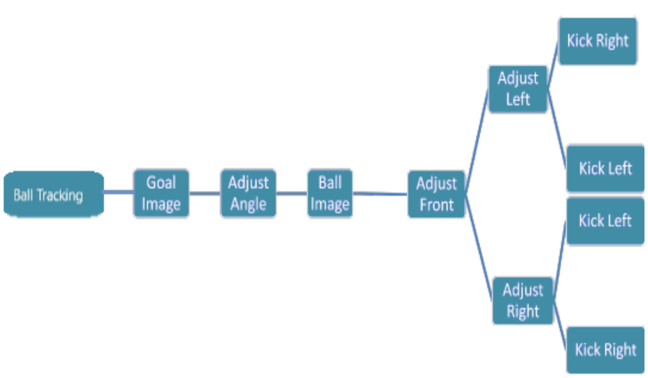

Figure 2: Penalty Kick Wireframe

Here, NAO would first look for the ball and will track it. After reaching to the ball (8 cm away), it would find the center of the black box meant as goal post and would adjust its direction with respect to goal. Again, it would look for the ball and adjust its position to kick with right or left leg whichever is nearer to the ball.

A) Ball Tracking

For detecting the red ball, we have used an API of Pynaoqi (ALTracker) where it is predefined to Nao what a red ball is. Also, we set parameter of the ball i.e. diameter, its position with respect to Nao, and the distance from the ball where it will stop.

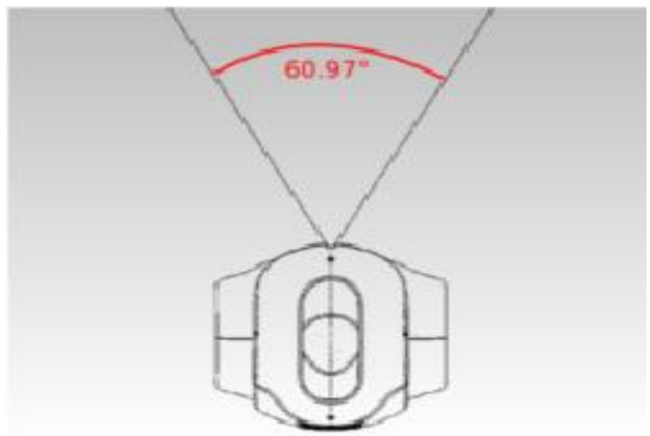

Figure 3: Top View of NAO Showing its Field of View

B) Goal Detection

For goal detection the RGB image derived from Nao's forehead camera is used. It is first converted to HSV (Hue Saturation Value) space and then morphological image processing operations (dilation followed by erosion are performed on it). Finally, the center of goal is found from it.

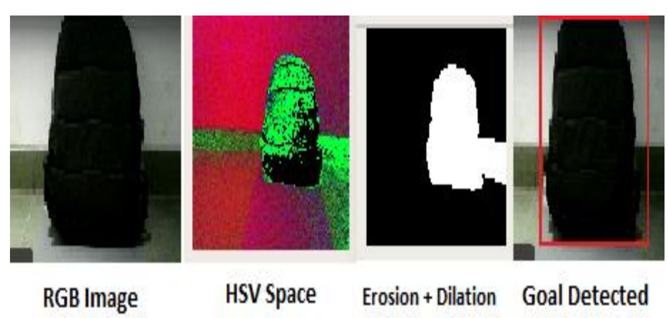

Figure 4: Goal Detection Images

C) Adjusting Angle

In figure 5, we can see how Nao changes its body orientation towards the goal. We have curated a very simple model (following the KISS- Keep it Simple Stupid principle) [6]. It assumes that the ratio of pixels from a point to other in an image is roughly equal to the ratio of their distances in the same image i.e.

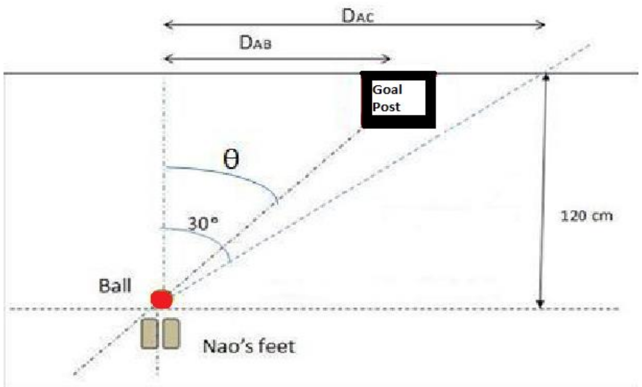

Figure 5: Calculation of angle to be adjusted

$P_{AB}/P_{AC}=D_{AB}/D_{AC}$

D_{AC} is found out by using the field of view angle of upper camera. As D is fixed at 120 cm and the Finally, as the distance to goal is fixed i.e. in our case equal to 120 cm. We can find θ¸ simply by

$\text{Tan } \theta = D_{AB}/120$

C) Ball Detection

After the angle is adjusted NAO's HeadPitch angle is changed from 0 to 29.5 degrees so that it can look down completely. The ball is then detected in the same way as goal, the difference is only that the image is taken from NAO's Lower Camera. Figure 6 describes the same.

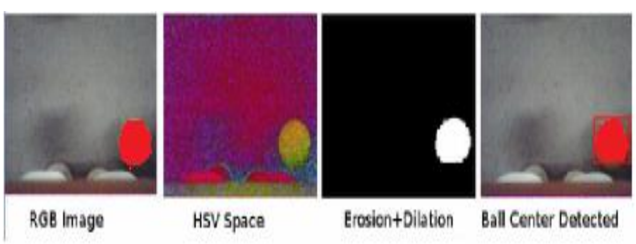

Figure 6: Ball Detection Images

D) Adjust Position

Looking at the position of the ball, NAO adjusts its position by moving front left or right. According to figure 6 which NAO decides to walk in front and NAO needs to move right. The number of steps taken are a function of the abscissa and coordinate of ball in image.

D) Kick

We have adopted a Bezier curve interpolation between keyframes for kicking the ball. Below in figure 7 you can see the 3 keyframes (can be understood colloquially as "screenshots"). Each of the three keyframes consists of different HipPitch, KneePitch and AnklePitch angles. Here instead of using a motion engine as in [1] we have manually found out these frames by checking stability of NAO in various positions.

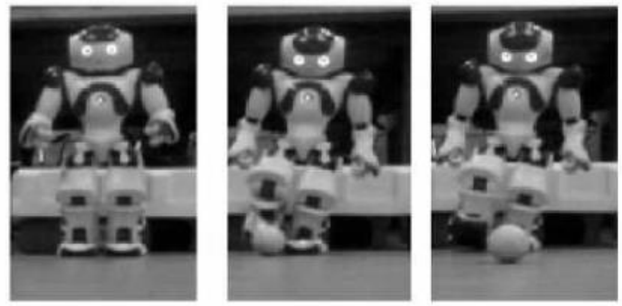

Figure 7: Keyframes for Kicking the ball

(2) Kicking from anywhere in frontal hemispherical plane can be decomposed into the following steps (as in figure 8)

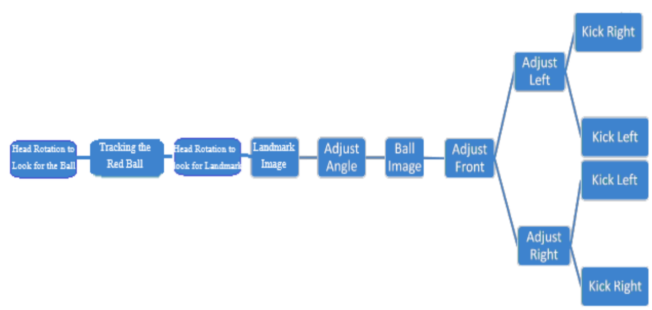

Figure 2: Kicking from Anywhere Wireframe

Here, NAO would first look for the ball by turning its head in a semi-circle until it finds the ball and as soon as it will find the ball it will start tracking it. After reaching to the ball (8 cm away), it would again turn its head in semi-circle and find landmark image as in figure 8 which is predefined symbol for NAO to recognise as landmark. Thereafter, it would find the center of the black region on landmark image and it would adjust its direction with respect to the symbol (as explained above). Again, it would look for the ball and adjust its position to kick to the landmark symbol. Thus, methodology is similar to as in penalty kick except the head rotation which is as follows:

Figure 9: Landmark Symbol as Goal Post

A) Head Rotation to locating for the ball

NAO will rotate its heads 90 degrees in one direction in both directions until it sees the ball. In this whole process as it sees the ball it will start tracking the ball and moving towards it.

B) Head Rotation to locating for the landmark symbol

NAO will rotate its heads 90 degrees in one direction in both directions until it sees the landmark. As soon as it sees the symbol it will adjust itself towards.

IV. RESULTS AND DISCUSSIONS

The Percentage success for the penalty kick through our implementation is coming out to be 55-60 %. And in case of kicking from anywhere in frontal hemispherical plane percentage success to kick the ball to the goal post comes around 30-35 %.

It is very difficult to develop this project in the Choregraphe software, we had to use the pynaoqi, python-sdk for connecting and performing various tasks with NAO. And we wrote codes [vi] using eclipse IDE for designing our navigation system and for kicking the red ball.

V. CONCLUSION

As of now we have implemented only static kicking i.e. if the robot once take decision to kick the ball, it will then irrationally kick it even if the ball is displaced from the position when the decision was taken. Dynamic kicking involves taking the feedback from Gyroscope and balancing the robot based on that. It could be a path for future work on it.

Also, right now, we have only one kind of kick i.e. no side kick etc. We can possibly find keyframes for maybe, a few more kick types and apply reinforcement learning [1] which will make the decision process rational and improve the results from 60% hit rate to 85% as stated in [1].

ACKNOWLEDGMENT

We are very thankful to all the jury at Bennett University and leadingindia.ai for providing us NAO robot, all necessary resources as well as suitable place and time to complete our project.

REFERENCES

[1] Hester, Todd, Michael Quinlan, and Peter Stone. “Generalized model learning for reinforcement learning on a humanoid robot." In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 2369-2374. IEEE, 2010.

[2] Czarnetzki, Stefan, Soren Kerner, and Daniel Klagges. “Combining key frame-based motion design with controlled movement execution." In RoboCup 2009: Robot Soccer World Cup XIII, pp. 58-68. Springer Berlin Heidelberg, 2010.

[3] Xu, Yuan, and Heinrich Mellmann. \Adaptive motion control: Dynamic kick for a humanoid robot." In KI 2010: Advances in Artificial Intelligence, pp. 392-399. Springer Berlin Heidelberg, 2010.

[4] Muller, Judith, Tim Laue, and Thomas Rofer. “Kicking a ball-modelling complex dynamic motions for humanoid robots." In RoboCup 2010: Robot Soccer World Cup XIV, pp. 109-120. Springer Berlin Heidelberg, 2011.

[5] Master's Thesis of Tomas Gonzalez Sanchez, Department of Computer Science and Mathematics, Universitat Rovira I Virgili, September 2009, 64-82

[6] Lovish and Rahul, Penalty Shootout by Aldebran NAO https://cse.iitk.ac.in/users/cs365/2013/submissions/~lovishc /cs365/project/index.html

[i] B. Tech (CSE), Galgotias University, MP
[ii] B. Tech (CSE), Mody University, Rajasthan
[iii] B. Tech (CSE), JECRC, Jaipur, Rajasthan
[iv] M. Tech (MN), IIT Kharagpur, West Bengal
[v] Asst. Prof. at CSE Dept at Bennette University, Noida

[vi] https://github.com/rajat1478/Redball-Kicking-By-Nao-Humanoid-Robot-