ROBO GOALKEEPER

Akshay Salgond[1], Simran Tawar[2], Jagruti Tate[3], Mahima Nagrale[4], Deepthi Oommen[5]

[1,2,3,4]B.E. DEPARTMENT OF COMPUTER ENGINEERING, TERNA ENGINEERING COLLEGE, NERUL
[5]PROFESSOR, DEPARTMENT OF COMPUTER ENGINEERING, TERNA ENGINEERING COLLEGE, NERUL

Abstract - The 21st century is a century for robotics. Robots have long borne the potential to bridge the gap between the cybernetic world (the internet of things) and the physical world. As the most promising candidate to theme the next major industrial revolution succeeding the present third (digital) industrial revolution, robotics is set to play an ever increasingly important role in society for its influence in every aspect of life, including medicine and healthcare, building service, manufacturing, food production, logistics and transportation.

The goal of this project is to create a robot goalkeeper for the embedded control systems at Terna Engineering College. The problem the robot goalkeeper solves is to detect a ball, calculate the ball's position, estimate its trajectory and predict a future point to intercept the ball.

Keywords*:* Embedded Control System, Hellobot, Bitmap, Blob detection, Image grabber, Serial port communication.

1. Introduction

In the last decade robotics has grown from a simple vision to a reality where they exist in business and among individuals [1][2]. Robotics deals with the design, construction, operation, and use of robots, as well as computer systems for their control, sensory feedback, and information processing. These technologies are used to develop machines that can substitute for humans and replicate human actions. Empowering laborers also, doing some portion of their work to make the procedure increasingly proficient and more solid. In the field of robotics, when building up an independent system, one needs to organize all the robot subsystems, its activities and practices so as to have an assignment plan fit for accomplishing the ideal objective for the operator.

In a robot goal keeper, one of the elements is the goalkeeper which has particular challenging characteristics, different from the other teammates, when designing and coordinating the execution of a robot task plan. The main purpose of a goalkeeper, is to defend the goal from the kicks of the opponent teams which means the actuation area of a goalkeeper is always near the own goal. Track and follow the ball motion, intercepting the ball before reaching the goal, covering the goal and removing the ball from the goal area. In our project we are developing an algorithm which applies the constraint on the colour and shape of the ball, also differentiating it from the field using the image processing. We have a single vision system integrated in our project which captures the motion of the incoming ball. According to the trajectory of the ball the algorithm commands the bot to move to an appropriate position consequently defending the ball.

2. Scope

We are going to build a goalkeeper robot using embedded systems and with the help of concepts of image processing. The main moto of this goalkeeper robot is to prevent the ball from entering into the goal. The problem the robot goalkeeper solves is to detect a ball, calculate the ball's position, estimate its trajectory and predict a future point to intercept the ball. The robot will make use of different sensors and cameras to keep track of the trajectory of the ball and will try to be as accurate as possible in defending the ball. Following are the different objectives of this robot:

1. Target (ball) detection.
2. Always keep track of target, keeping visual contact.
3. Position and velocity determination of target.
4. Position determination of the goalkeeper.
5. Trajectory estimation and position prediction of the target.
6. Interception of target at predicted position.

3. Construction

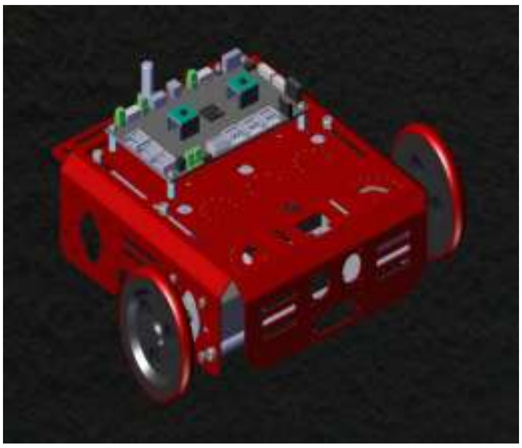


Fig.1 Hellobot Cover image

The Mechanical components used are:
Chassis, Wheel connectors, Castor wheel, Link connectors, Battery case.

The Electronic components used are:
Hello bot base board, Stepper motors, Remote control, Power cable.

Fig.2 LOGITECH C310 HD WEBCAM

4. Methodology

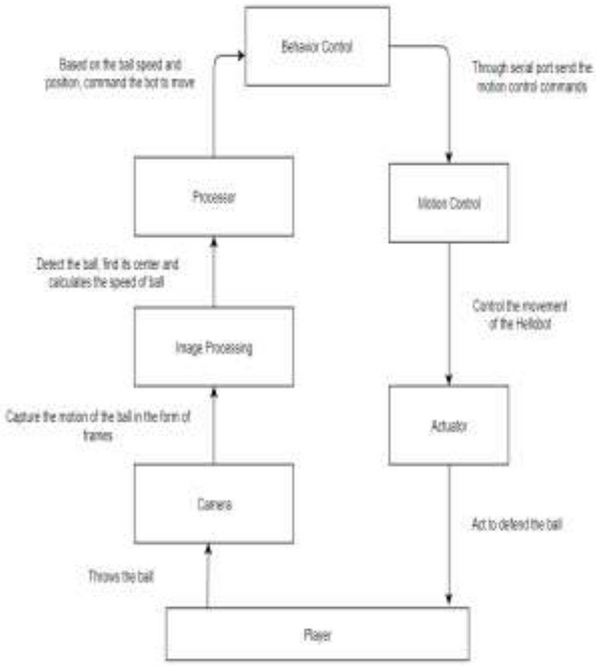


Fig.3 Software Architecture

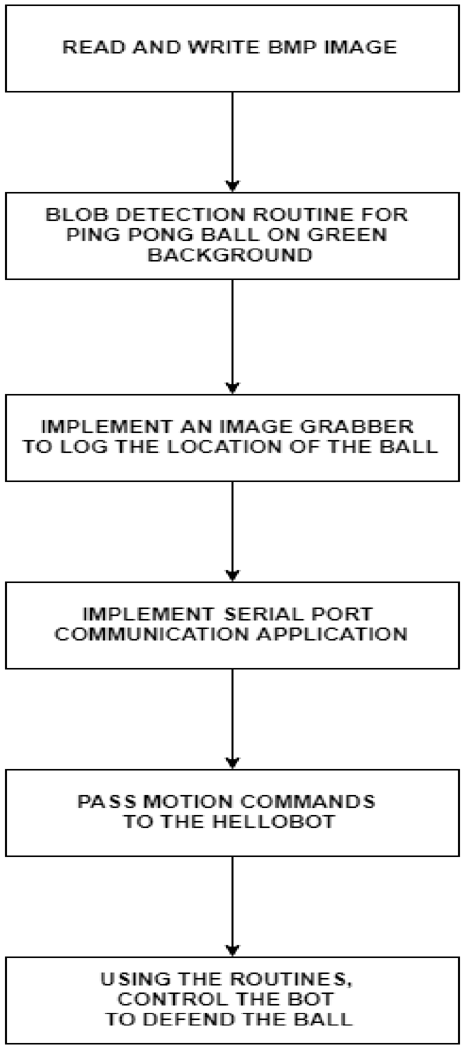


Fig. 4 Flow Diagram for Robo Goal Keeper

Task 1: Read and write bitmap images

In this task we are developing an algorithm which reads the bitmap images along with their header and also writes the data to another file to create the exact copy of the bitmap image.
Algorithm:

Step 1: First read the input bmp file using file input.

Step 2: Then verify if the header of the bmp file is correct or not.

Step 3: If the header is correct, then store the pixel values of the input image.

Step 4: Copy paste the header and pixel data to a new file using file output.

Step 5. Now check if the output image is an exact copy of the input image or not.

The Concept of padding:

Bitmap image format stores the information of pixels bytes in the multiple of 4 bytes. Hence while copying the pixels of an input file to an output file, you have to make sure that the number of bytes in a row is not multiple of 4, you have to pad the required number of bytes with 0.

The formula that can be used to calculate the number of padding bytes required is as follows:

Padding bits= 4 - (Width_Of_Image * 3) % 4) % 4

Task 2: Implement a blob detection routine for ping pong ball on green background

In this task we are developing an algorithm to detect the blob in the image of the ping pong ball on green background which will help us to find the position of the ball (centre of gravity of the ball) when it is in motion.
STEP 1: We are capturing the image in RGB format and then converting it into Grey scale format.

Formula to convert RGB into Grey scale:

Grey = (blue * 0.3) + (green * 0.59) + (red * 0.11)

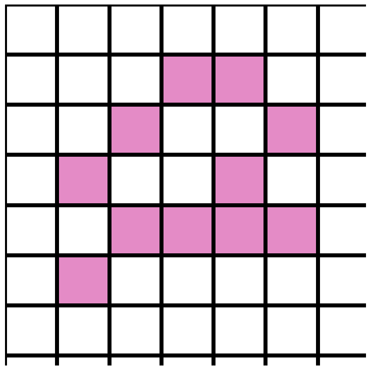
STEP2: Definition of an 8 connected component

A Set of black pixels, P, is an 8 connected component (or simply a connected component) if for every pair of pixel pi and pj in P, there exists a sequence of pixels pi, …., pj such that:
a) All pixels in the sequence are in the set P i.e. are black, and
b) Every 2 pixels that are adjacent in the sequence are 8-neighbors.

NOTE
All 4-connected patterns are 8-connected i.e. 4-connected patterns are a subset of the set of 8-connected patterns. On the other hand, an 8-connected pattern may not be 4-connected.

Example of 8-connected pattern:

The diagram below is an example of a pattern that is 8-connected but not 4-connected:



We are using the recursive grassfire algorithm to detect and store the blob.

In the grass fire algorithm, we first check if the current pixel is of the blob's grey scale color or not and if it is then first add it to the blob structure and then move forward searching for neighboring pixels to be of the same color or not.

If they match the color then add them in the blob structure too and resume to keep checking for the neighboring pixels using 8 connectivity.

Now while checking for these pixels if we come across a pixel that is already visited then do not add this pixel into the blob structure and continue with the recursive process.

STEP 3: NOISE REDUCTION

After detecting the connected blobs in the image we have to differentiate between the desired blob and the noise. To get an image with just the desired blob we need to eliminate the noise.

In order to achieve that, we have set two criteria that cancel the noise, those are:

1) If the color of the other blobs is different than the color of the desired blob then, change the different colored blobs to the color of the background that is white. (the colors mentioned here are in greyscale format)

2) After checking the first condition if we find a noise of the same color as the blob then we check for the second condition that is if the size of the other blobs is less than the threshold value (Here the threshold value is set as the number of pixels in the desired blob), then those blobs which are smaller in size are considered to be noise and are removed using the same process.

Step 4: Find the centre of the blob detected

When we come across the first blob in the image, the centre of the first row of the detected blob is calculated and then we traverse vertically downwards from that centre to find the total number of pixels in the blob vertically. Now we find the centre of those vertical pixels. This is how the centre of the blob is estimated.

Task 3: Implement an image grabber and log the location of ball in real time in the console.

In the previous task as we have found the position of the ball, now we are developing an algorithm to capture the trajectory of the ball using the image grabber and log the location of the ball in real time.

Task 4: Working of Image grabber and calculating time and distance for speed.

The below task is to send the motion commands to the robot in order to move it to the respected position and to defend the ball from getting into the goal position. The image grabber will capture the images, which is in the form of frames and we are calculating the change in the position of the ball between the consecutive frame of images which we are getting from the camera. And also we are able to find the difference between the time of capture of those images. So by doing this we are calculating the speed of the ball and we keep calculating this for each second that passes, by what distance the bot needs to move and eventually by covering that much distance the bot defends the ball.

Formula for calculating speed is
Speed = Distance / Time

Task 5: Implement a serial port communication application which connects with hellobot.

As we have found the correct position of the ball in real time now it's important to command the hellobot to move to an appropriate position. For this we are developing an application on pc to send the commands to the hellobot to move.

Task 6: Define the protocol for sending motion control commands to hellobot.

As we had developed an application in the previous task we are developing a protocol which will bind our application to the hardware (hellobot) and will send the motion control commands to it.

Task 7: Using the routines developed in task 3,4 and 5, control the bot to defend the goal.

According to the motion control commands sent by the protocol to the hellobot and thereby controlling it to move to an appropriate position to defend the ball.

5. Conclusions

Proposed method can be implemented into an application of colored ball tracking system on the robot. This proposed system will perform the task of robo goalkeeper in the real time. The task of the goalkeeper is to detect the ball as soon as the player throws it. System is able to detect the presence of the ball by a specific color in the greyscale color model. Ball position coordinates can be detected by the system using 8-cell connectivity for goalkeeper robot soccer to block the ball.[3]

In our formulation of the problem the goalkeeper moves left and right in the goal line in order to defend the ball from entering the goalpost. The control strategy is derived based on geometrical rules and the robot goalkeeper is controlled in the linear velocity according to the ball speed and direction. In our case we are using grids which are nothing but horizontal and vertical lines which divides the ground in small sections and based on that we are commanding the robot to move depending on the grid where the ball is located.[5]

This paper presented a goalkeeper task of defending the penalty shootouts. Some good results were achieved, mainly, on the goalkeeper task performance. The results are satisfying, but further work needs to be done, namely to improve the reaction speed, to use a variable intercept line and include different behaviors.[4]

6. Reference

1. J.K. Archibald Dept. of Electrical. & Computer. Eng., Brigham Young Univ., Provo, UT, USA James K. Archibald received the B. S. degree in mathematics from Brigham Young University, Provo, Utah, in 1981 and the M. S. degree in computer science in 1983 and the Ph. D. degree in computer science in 1987, both from the University of Washington. His research interests include multiagent and embedded systems.
2. Indra Adji Sulistijono Politeknik Elektronika Negeri Surabaya (PENS) · Graduate School of Engineering Technology.
3. Arif Rahman to detect the presence of the ball by a specific color in the HSL color model. Ball position coordinates can be detected by the system in 9-Cellss coordinate for goalkeeper robot soccer to block the ball. (https://www.researchgate.net/publication/265852164_Colored_Ball_Position_Tracking_Method_For_Goalkeeper_Humanoid_Robot_Soccer).
4. https://fenix.tecnico.ulisboa.pt/downloadFile/395142130507/CarlosMartins_Article.pdf
5. Belkhouche, F., & Belkhouche, B. (n.d.). On the control of a wheeled mobile robot goalkeeper. IEEE Conference on Robotics, Automation and Mechatronics, 2004. doi:10.1109/ramech.2004.1438022
url to share this paper: