Train a robot to block incoming ball
BLG456E Project Proposal

*By **FOSFOS**:*

Furkan Aksın 150130701

M. Faruk Yazıcı 040100112

Ozan Özyeğen 150120115

Sema Karakaş 150120003

Sercan Bayındır 150130104

Table of Contents

Introduction

The robot problem that we are expecting to solve in this project is the blocking of a ball by a goalkeeper robot in the Gazebo platform. This topic is especially exciting for us because we believe humanoid robots and its human like actions are more interesting subjects to work on. The main approach will be the track of balls action and move the humanoid robot accordingly in order to block to ball. The ROS and Gazebo environments will be used in this project. In this report, general information about the project is given such as background amd methods. The schedule that we are planning to follow is also given in this report.

Background

Our main inputs will be the locations of ball, keep and the robot. The robot will track the balls actions. Then, it will decide which direction it should walk and walk sideways.

In order to achieve this, there are some basic issues that must be solved. First, creating and animating the ball model in simulation is very important. If the ball moves too fast, the robot can not block the ball. Thus, the ball must created accordingly. We may need to give state and update the position of the ball. However, if this approach is taken, it can cause crash on contact of the robot. Another approach that can be used is to use a ramp and use its slope to move the ball.

The robot will see the ball and decide how it should move to the ball. It can move sideways or walk straight to the ball. Or it can fall sideways to block the ball.

ROS on Gazebo simulator will be used in the project. The speed of the ball in Gazebo is a problem which is explained above.

From the Nao ROS Wiki[1], footstep_planner package will be used for moving the robot. naoqi_pose package will be used for collapse simulation. For visualization of the robot, naoqi_dashboard package will be used.

Methods

When the robot gets information from the RGBD camera, the point cloud will be analyzed, the position of the ball will be extracted. Then, the robot will make decision according to speed and position of the ball. It will have several options such as:

- Stand still and not move

- Walk sideways
- Turn and walk to intersect the ball
- Walk sideways and collapse

These are the 4 main options that the robot will have. In order to implement all these scenerios, a footstep planning has to be created for each.

Our quantitave measure of success will be the number of the shoots that can be blocked by the robot from 10 different angles. We will create a histogram chart which includes the angle, number of shoots tried, and the success (block of the shoot by the robot).

Schedule & Outcomes

A summary of tasks and their delivery times can be found in Table 1. Below that the tasks are detailed.

Table 1: Main tasks and responsibilities

Task	Responsible Person	Completion date	Amount of time expected (hours)	Share of marks (%)
WP1.1: Localization of the ball from point cloud data	Sercan Bayındır	30 Nov	16	10%
WP1.2: Simulation and modelling of the ball	Furkan Aksın	30 Nov	16	10%
WP1.3: Walking simulation	Ozan Özyeğen	30 Nov	22	10%
WP1.4: Collapse simulation	Sema Karakaş	30 Nov	12	10%
WP1.5: Modelling of the keep	Faruk Yazıcı	30 Nov	16	10%
WP2.1: Speed measurement from ball position data	Sercan Bayındır	26 Dec	12	10%
WP2.2: Decision algorithm	Faruk Yazıcı	26 Dec	22	10%
WP2.3: Turn head towards ball simulation	Furkan Aksın	26 Dec	12	10%
WP2.4: Collapse simulation integration	Sema Karakaş	26 Dec	12	10%
WP2.5: Walking simulation integration	Ozan Özyeğen	26 Dec	12	10%
Bonus: Movement of the robot will be parallel to the keep	Ozan Özyeğen	26 Dec	12	+10%

WP1.1: Localization of the ball from point cloud data

Camera feauture will be used. Point cloud data will be used as input.

Completion will be determined by rate of succesfull localization of the ball.

WP1.2: Simulation and modelling of the ball

Ball model will be created. Simulation method options will be tried.

Completion will be determined by smoothness of the ball movement.

WP1.3: Walking simulation

Footstep planning will be used.

Completion will be determined by number of walk options.

WP1.4: Collapse simulation

Body Posing package will be used.

Completion will be determined by the time that collapse simulation takes and the range it can reach.

WP1.5: Modelling of the keep

Keep model will be created. Realistic model will be created.

Completion will be determined by similarity to a real keep.

WP2.1: Speed measurement from ball position data

Speed of the ball will be determined from position data list. Position data mined with WP1.1 will be used.

Completion will be determined by true speed value calculated.

WP2.2: Decision algorithm

Decision mechanism will be created. The robot decide how to move using the data gathered with WP1.1 and 2.1.

Completion will be determined by the rate of blocked balls.

WP2.3: Turn head towards ball simulation

Body posing package will be used. Position data from WP1.1 will be used.

Completion will be determined by tracking of the ball with head.

WP2.4: Collapse simulation integration

Collapse simulation node will be integrated with decision mechanism.

Completion will be determined by the right behavior of the robot.

WP2.5: Walking simulation integration

Walking simulation node will be integrated with decision mechanism.

Completion will be determined by the right behavior of the robot.

Bonus: Movement of the robot will be parallel to the keep

Walking simulation will always be paralel to the keep.

Completion will be determined by the right behavior of the robot.

Conclusion

The robot will make decisions based on the data gathered from the RGBD camera. From this information, it will calculate the direction and speed of the ball. Then it will decide how it should move to block the ball. It can stand still, walk or collapse. The successful blocks will be used to determine the success rate of the robot.

Reference List

[1]"nao - ROS Wiki", *Wiki.ros.org*, 2016. [Online]. Available: http://wiki.ros.org/nao. [Accessed: 01-Nov- 2016].