Robocup 2021

JuniorSoccer simulation

Team name : Hero Boys

NimTajiki [1*], Arian Ramianpour[1],Mohammad Hossain Sedghi [2],
Amir Mehdi Momeni [2]

[1]Allame Helli 2 High school, Tehran, Iran
f_daemi@yahoo.com

Abstract

With the introduction of RoboCup simulator in the Junior Football league, we started working on this simulator. After getting acquainted with the software environment and programming, we implemented various algorithms and strategies. Using the data received from the simulator and using the coordinates and robots, we were able to implement various algorithms so that one of the robots as a goalkeeper and other robots play the role of defender and attacker. After the team approved the code sent by the team, we worked more seriously. We followed and we were able to get better conditions to participate in RoboCup 2021.

Keywords: software, programming, robot, simulator,strategies,Webot

1- Amir Mehdi Momeni
2- Arian Ramianpour
3- Nima Tajiki
4- Mohammad Hossain Sedghi

1 Introduction

HeroBoys Team from Allame Helli 2 High school in Tehran, Iran, we worked on real light weight soccer robots for a year and followed the hardware design and mechanics of soccer robots as well as robot programming professionally. With the introduction of RoboCup simulator, we decided to try this simulator. After installing the webot software and running the junior the junior soccer sim program on it, we became interested in this simulator and decided to participate in RoboCup 2021, so we sent the initial code to confirm our team in the Asian region, and after Confirmation our team is currently preparing for the World RoboCup. We are programming, making TDP by dividing tasks among team members.(figure_1)

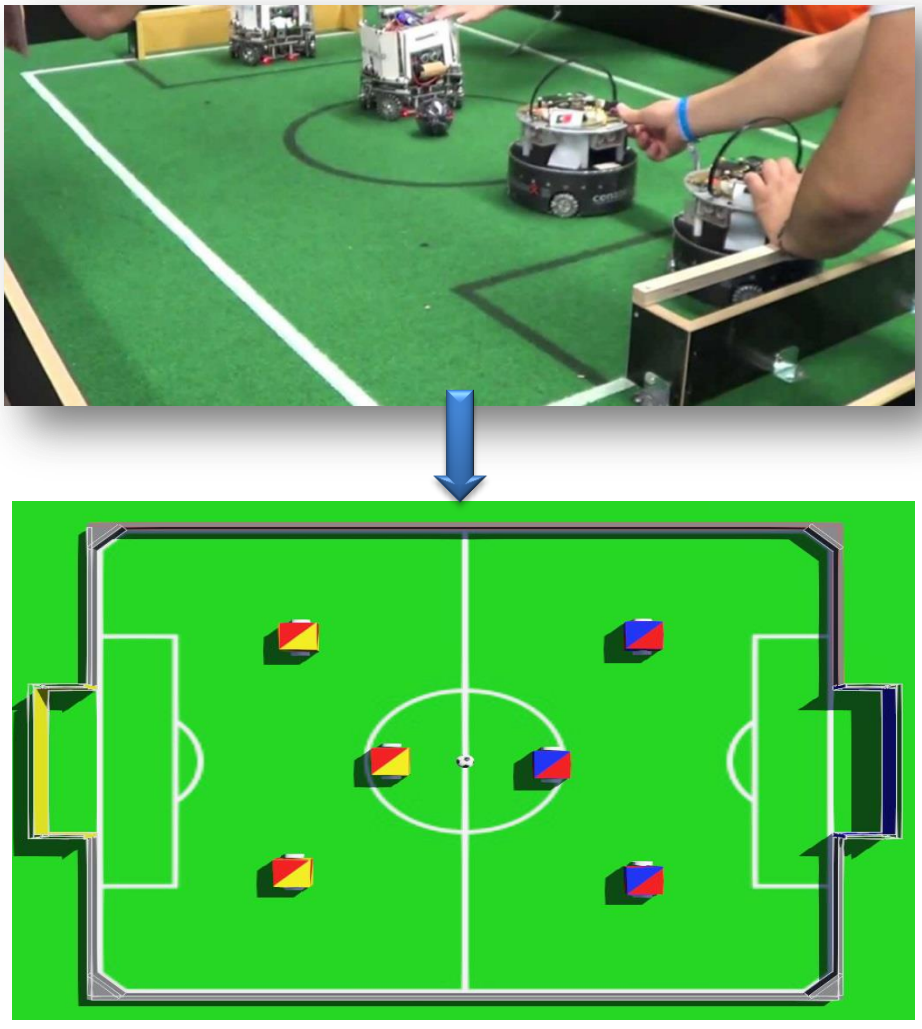

Figure_1: RoboCup simulator

2 Software

Webot is a software which all real robots can be simulated and simulator of that robot can be designed. The junior football robot simulator is also designed with webot software and there are 3 robots for playing football and each team must have these 3 robots. Write the game program webot software allows us to make changes in programming on robots and evaluate the performance of our robot in the game with the opposing team. The programming language of robots is Python, and using this. The language of different algorithms can be implemented on robots.(figure_2)

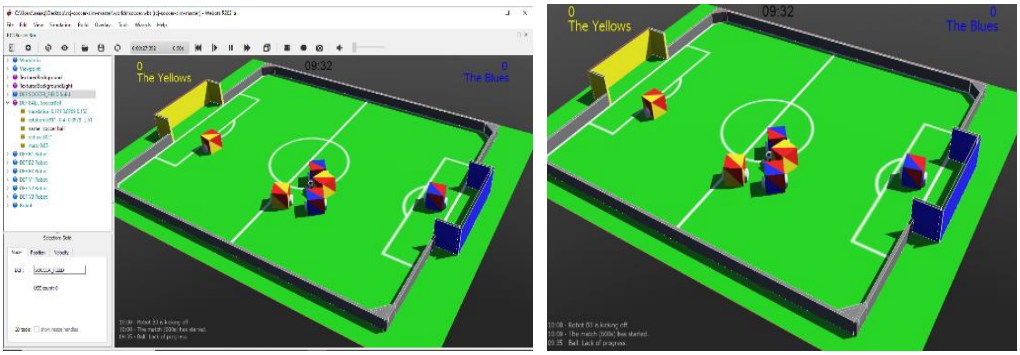

Figure_2: Webot software

Webot software provides us with robot and ball data using Python commands and defining different functions. These data show the position of the robots and ball on the ground relative to the center of the ground. They are important to chase the ball.(figure_3)

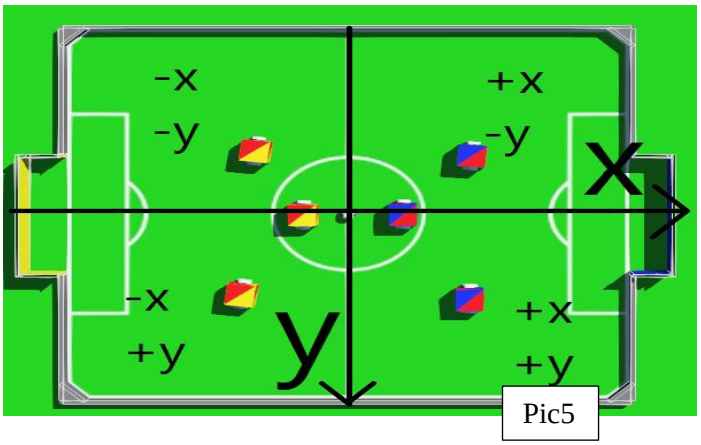

Figure_3: ball and robots data

3 Algorithm

Finding the ball and determining its direction is the first and most important point in the game. We have used 4 directions to move the robot easier and more accurately. Our robot is rated from zero to 360 degrees so we will have 4 directions as shown below.(figure_4)

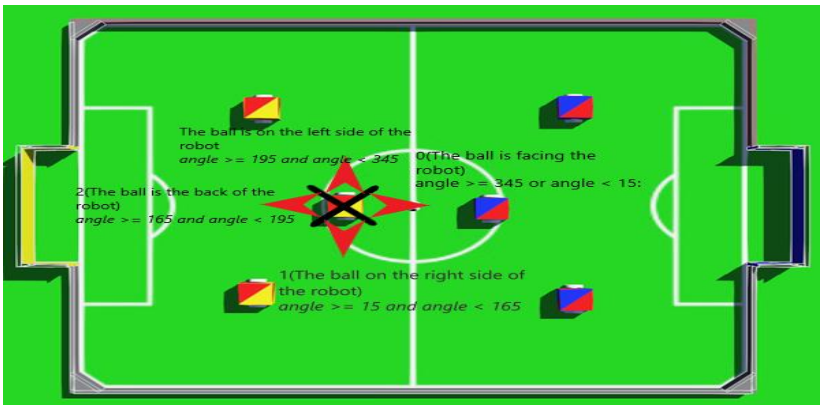

Figure_4: Finding the ball

We also calculated the distance between the ball and the robot so that we could programing the best and most accurate movements for the robot.(figure_5)

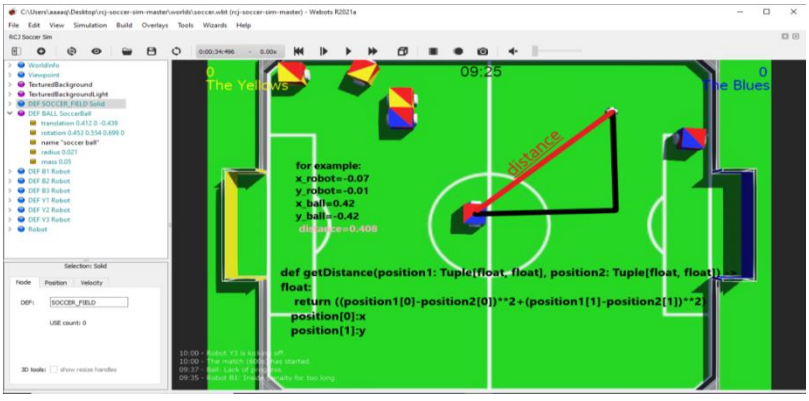

Figure_5: ball distance between the ball and the robot

The program for this section is located in the utils file.

Relevant codes:

```
def getDirectionBy4(angle: float) -> int:
    if angle >= 345 or angle < 15:
        return 0
    elif angle >= 15 and angle < 165:
        return 1
    elif angle >= 165 and angle < 195:
        return 2
    elif angle >= 195 and angle < 345:
        return 3
def getDistance(position1: Tuple[float, float], position2:
Tuple[float, float]) -> float:
    return   ((position1[0]-position2[0])**2+(position1[1]-
position2[1])**2)**0.5
def correctAngle(angle: float) -> float:
    if angle < 0:
        angle = angle % -360+360
    else:        angle%=360    return angle
```

Once the direction of the ball is determined, we must move towards the ball with proper accuracy and speed. We programmed the appropriate movement algorithm using the direction and position of the ball on the playing field.Using the “newDataProccessing” function, we receive information about the position of the ball and the robot, and by giving that information to the “move” function, we transfer the robot to the robot position.

Relevant codes(Part of the code):

```
def newDataProccessing(self):
self.rawData = self.get_new_data()
self.position=self.rawData[self.name]['x'],
self.rawData[self.name]['y']
self.lastBallPosition=self.ballPostion
self.ballPostion = self.rawData['ball']['x'],
self.rawData['ball']['y']
self.distances = {}
for i in ROBOT_NAMES:
self.distances[i] = utils.getDistance(
(self.rawData[i]['x'],self.rawData[i]['y']),(self.rawData[
'ball']['x'], self.rawData['ball']['y']))
def move(self, position):
x,y = position
angle = utils.correctAngle(self.get_angles(
{'x': x, 'y': y}, self.rawData[self.name])[0])
direction = utils.getDirectionBy4(angle)
value = self.distance/0.15*1.5
if value < 1:
value = 1
if angle < 90 or angle > 270:
if direction == 1:
self.left = -10
self.right = -value
```

4 Game strategy(The role of robots)

Attacker: This robot has the task of following the ball within a certain range and moving towards the opponent's goal.

Goalkeeper: This robot has the task of following the ball in a certain range according to the Y axis.

Shooter: This robot is programmed for simultaneous use in defense and attack.(figure_6)

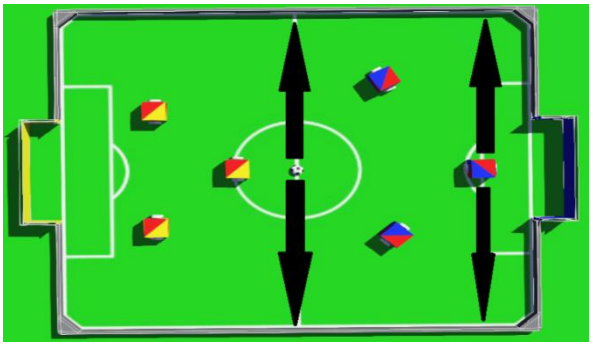

Figure_6: game strategy

5 Strategy:

At the beginning of the game, the number one robot goes to our goal. Robots two and three move towards the ball.(figure_7)

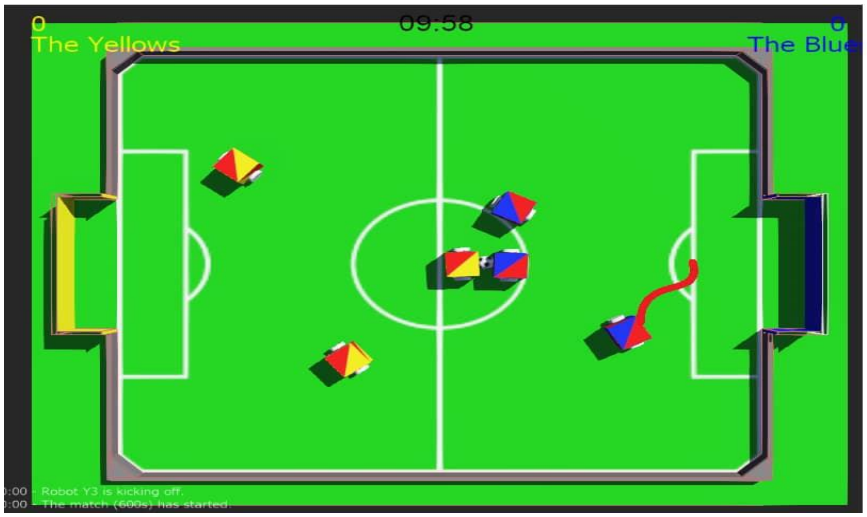

Figure_7: game strategy

Relevant code(goal keeper):

```
if self.role=='goalkeeper':
if self.position[0]<0.65:
self.move((0.7, 0))
else:
if self.orientation>170 and self.orientation<190:
if self.ballPostion[1]>self.position[1]:
self.left=-10
self.right=-10
elif self.ballPostion[1]<self.position[1]:
self.left=10
self.right=10
else:
self.stop()
else:
if self.orientation>180 and
self.orientation<360:
self.left=-5
self.right=5
else:
self.left=5
self.right=-5
```

When the ball is placed next to the walls or corner points, the number two robot distances itself from the other robots involved in the ball location and after a few seconds (if the ball remains fixed) moves to the neutral point in the middle of the field.(figure_8)

Figure_8: game strategy

Relevant code(shooter):

```
if self.role=='shooter':
self.checkLackOfProgress()
if self.ballLackOfProgress:
self.move((0.2, 0))
if 0.15<self.position[0]<0.25 and -0.05<self.position[1]<0.05:
if 100>self.orientation>80:
self.stop()
elif 90>self.orientation>0 or 360>self.orientation>270:
self.left=-5
self.right=5
else:
self.left=5
self.right=-5
else:
if self.ballPostion[0]<0.55:
self.move(self.ballPostion)
else:
self.stop()
```

6 References:

- https://drive.google.com/file/d/1WdsB8ayPtIh3qTKui0jDZdmWI-ShMa8/view
- https://github.com/RoboCupJuniorTC/rcj-soccer-im/archive/refs/heads/master.zip
- https://www.python.org/downloads
- https://github.com/
- https://www.avrfreaks.net
- http://www.hpinfotech.ro