Technical University of Crete
Electrical and Computer Engineering

Autonomous Agents
INF 412
1st Laboratory Exercise
Report

Παπαματθαιάκη Ηλέκτρα-Δέσποινα (2018030106)

1. The purpose of the exercise

This laboratory exercise was designed with the intention of getting us familiar with the Webots simulator as well as the provided code and understanding how the Nao robots react to it.

2. Environment

I worked with the "robotstadium_nao_vs_robotis-op2" environment, initially featuring five Nao robots and five opponent robots(1 goalkeeper and four players each). To simplify the world, I retained only one Nao robot to attempt scoring. Moreover, the Nao's camera is programmed to identify the colors of the ball and goals. This is the reason that we modified the textured background and textured background light settings from "stadium" to "mountains." This adjustment aligns better with the camera's color recognition programming.

3. Adjustments

The adjustments that are presented below were implemented in the provided code in order to make the Nao robot player score a goal.

3.1 The shooting Motion

In the existing "nao_team_1" code, the robot's movements were influencing the motion of the ball. Specifically, when employing the forwards motion while the ball was positioned in front of the robot, the ball responded by moving accordingly. Within the motions directory, there is a "shoot" motion. The initial modification involved implementing this "shoot" motion into the code, as following:

```
public class FieldPlayer extends Player {

  . . .
   //* add new Motion for shooting *//
   private Motion shootingMotion;

. . .

   public FieldPlayer(int playerID, int teamID) {

. . .
```

```
//* add new Motion for shooting *//
shootingMotion = new Motion("../../motions/Shoot.motion");
```

In the following section, a code snippet of where the shootingMotion was implemented, will be presented.

3.2 The function run()

The subsequent modifications were applied to the 'run()' function. In particular, during the ball-searching phase, I observed that when the ball is in close proximity to the robot, it continues searching without success. What I thought was that simply stepping back once was not enough to locate the ball. As a result, an extra step backward was added.

```
@Override public void run() {
  step(SIMULATION_STEP);

  while (true) {

    runStep();

    getUpIfNecessary();

    while (getBallDirection() == NaoCam.UNKNOWN) {
      System.out.println("searching the ball");
      getUpIfNecessary();
      if (getBallDirection() != NaoCam.UNKNOWN) break;
      headScan();
      if (getBallDirection() != NaoCam.UNKNOWN) break;
      playMotion(backwardsMotion);
      //* added one step backwards*//
      if (getBallDirection() != NaoCam.UNKNOWN) break;
      playMotion(backwardsMotion);
      if (getBallDirection() != NaoCam.UNKNOWN) break;
      headScan();
      if (getBallDirection() != NaoCam.UNKNOWN) break;
      turnLeft180();
    }

    . . .

    }
```

After running the simulation multiple times, I noticed that the shooting motion executes correctly when the ball's direction is approximately -0.15. Additionally, I found that adjusting the first part of the code related to "short distance" to 0.22 instead of 0.3 improves its performance. Afterwards, I added a part in which Nao exclusively initiates a shot when the goal direction is zero because this way it is easier to hit the target. If the direction is not zero, itt simply moves forward in order to find the right angle to attempt the shot again. The modifications are shown in the following part:

```
  @Override public void run() {

. . .

      //* changed to 0.22 from 0.3 based on observations from
simulation *//
      if (ballDist < 0.22) {
        System.out.println("short distance");

        if (ballDir < -0.15)
          playMotion(sideStepLeftMotion);
        else if (ballDir > 0.15)
          playMotion(sideStepRightMotion);
        else if (goalDir < -0.35)
          turnLeft40();
        else if (goalDir > 0.35)
          turnRight40();
          //* only shoot when goal direction is zero and ball is close
enough *//
        else if (ballDist<0.15 && goalDir == 0.0)
          {System.out.println("shooting !!!");
          playMotion(shootingMotion);}
         //* go forwards if goal direction is zero to get close to the
goal*//
        else if (goalDir == 0.0)
          playMotion(forwards50Motion);
        else{
          System.out.println("I'm not gonna shoot yet!!!");}
        }

. . .
}
```

4. Conclusions and observations

The current code implementation still poses a challenge for Nao in locating the ball when it is in close proximity to its body. The struggle is evident during every simulation I run. While the code performs well in many simulations, it does not consistently achieve the goal in all instances. Moreover, after the goal, it finds it difficult to locate the ball again. Additionally, I observed that when Nao falls forward, the implemented function to stand up is ineffective. Despite my attempts to resolve the issue using foot sensors, I was not able to achieve a successful solution.