Programming Cooperative Behavior

Corinne Alini, 2018

This project aims to adapt the behavior system of the robots on Bowdoin's RoboCup team with the ultimate goal of creating elaborate plays. The Northern Bites' robots currently operate as individuals and the focus of my work is to prepare the robots for cooperative actions or teamwork such as passing to a teammate. I am specifically working with the behavioral code in the Claims Transition module. The Claims Transition module uses a Boolean function that tells the robot whether or not to continue to pursue the ball. One of the most fundamental features of teamwork is identifying when to continue to chase the ball and when back off to let a teammate get the ball. However, it is difficult for robots to communicate their intentions with their own teammates when participating in drop-in games. Drop in games are a separate competition in which teams are formed of robots from various teams with the objective being cooperative play. Focusing on behavior during drop-in games provides a format to improve identification of the parameters necessary to determine claims transition. The purpose of my project this summer was to adapt the claims transition to improve robot function in drop-in games.

When adapting the Claims Transition module, I discovered that the robot’s messages when communicating in the drop in games lacked the teammates distance to the ball and ball bearing. During normal competition, we are able to control all the information in the message packet due to the fact that all the robots are part of the Northern Bites team. However, not all robots are from the Northern Bites in drop-in games. As it is not mandatory for all the teams to include such information in the drop-in message packet, the function in the Claims Transition module failed to run. My main issue was figuring out how to convert the information the robots receive in the drop-in message packet to work for our system. I needed to figure out how to make our current module work with few modifications in order to conclude whether or not the robot should continue going for the ball.

Working to calculate the data needed to compute the sigmoid function, I used the known information about the ball and the robot to determine the teammates distance to the ball and the ball bearing. First, I wrote the function to calculate the position vector b while utilizing the x and y components of the robot’s position relative to the origin. Next, the code computed the angle between the position vector and the robots heading by finding the difference between the heading and the ball bearing. Once the angle and position vector b were calculated, another function utilized the information to determine the ball’s x and y coordinates relative to the origin. The next challenge was to use the ball’s position to determine the teammate’s ball bearing and position vector m to the ball. I used the difference between the mates x and y component and the ball’s x and y component in order to obtain the x and y component of a new position vector m. After, the program calculates the position vector m by using Pythagorean theorem. The program also calculates the angle between the teammate and the ball by subtracting the heading from the ball bearing.

When we got to the competition I discovered that the new claims transition worked well in theory, but not in practice. The robot’s teammates sent the localization information to the robot. My calculations used this information to determine the missing data needed to run the claims transition. However, our teammates sent the incorrect localization information; therefore, the distances and heading in the actual games were throwing our drop-in player off. In the future, I believe a module to detect other robots would be beneficial to our success in the drop-in games. This module will allow us to ignore the information sent from other teams and solely rely on how far we believe the robot is from us. However, my program allowed us to discard some irrelevant data so we were able to still participate in the drop-in competition successfully and place 8th in the drop-in games.

Faculty Mentor: Professor Eric Chown
Funded by the Maine Space Grant Consortium