Timothy Sweet and Alex Ga

Introduction

This robot was designed following the subsumption architecture. there are four behaviors. The bottom behavior is obstacle avoidance, the second behavior is shooting the ball, the third behavior is seeking the ball, and the top behavior is wandering.

Behaviors

The obstacle avoidance behavior polls the right and left sonar. If an obstacle is detected on the left sonar the robot drives straight in reverse until the obstacle dissappears from view. If the right sonar detects and obstacle the robot rotates counter-clockwise until the obstacle dissapears from view. This behavior inhibits anything.

The ball shooter behavior detects when the ball is in the arm and attempts to shoot the ball towards the correct goal. This may include backhanding the ball. This sbehavior is inhibited by obstacle avoidance.

The seek ball behavior attempts to drive towards the ball such that it will be loaded into the arm. This behavior inhibits wandering.

The wandering behavior spins the robot counters-clockwise. this will almost always result in ball visibility.

The defence task positions the robot at a 45 degree angle to the shooter on the appropriate side of the field if it detects the ball in its peripheral vision. If it detects the ball directly in front of it, it turns the vehichle 90 degrees counter-clockwise and seeks the ball laterally with the arm open.

The penalty kick task positions the robot at a 90 degree angle to the ball and uses the arm and the robot to get the ball.

Design choices

There is minimal message passing between behaviors. All communication is done through the world. Since there is no explicit subsumption architecture, motor control inhibitions are past through a global array.

Design challenges

Since the NXT crashes when sensors are polled too quickly we implimented many layers of abstaraction between the behaviors and sensor polling functions, to allow call delays and custom safecall implimentation. This allowed us to impliment the defined sensor data when we could not use the IR ball.

Design Strengths

The biggest advantage our robot had was probably its speed. During the competition, the outcome of the ride was usually decided in fewer than ten seconds. When our robot succeeded it was usually due to being able to reach the ball before the opponent. Another advantage our robot had over the competition was its throwing arm, which allowed us to maneuver the ball past the enemy robot quickly, and usually before they had a chance to react. One final merit of robot was its alternative start button mechanic, which allowed our robot to start the game immediately and not waste time on initialization. This mechanic proved to be popular and effective enough to appear on at least two other robots.

Problems Encountered and Possible Solutions

During the competition, our robot developed a habit of bearing right while driving; this was likely due to our wheels not spinning synchronously. We had encountered this problem before, and we had fixed it by coding in the correct power ratio to each of the motors, but what we did not consider was that the ratio would change depending on the charge of the battery. We noticed that as the charge on the robot diminished, this problem of bearing right was mitigated, so battery charge was very likely the problem. To fix this, we should have tested our robot under the same conditions we would have during the final competition. If we had done this, we would have been able to correct our ratio and the robot would have driven in a straight line.

One other problem we encountered was that our robot was much too light compared to some of the robots it competed against. On several occasions during the competition, we found our robot being carried by the opposing robot traveling the opposite direction. To fix this we should have either weighed our robot down (but lose velocity and acceleration) or improved our obstacle avoidance algorithm, which may have caused our robot to become less aggressive.