Autonomoboro
Lab07: Futball

Bandith Phommounivong, Terence Henriod

December 16, 2014

Abstract

A discussion of the design and performance of this year's 1st place autonomous "soccer" robot, Gadnuk, Breaker of Worlds.

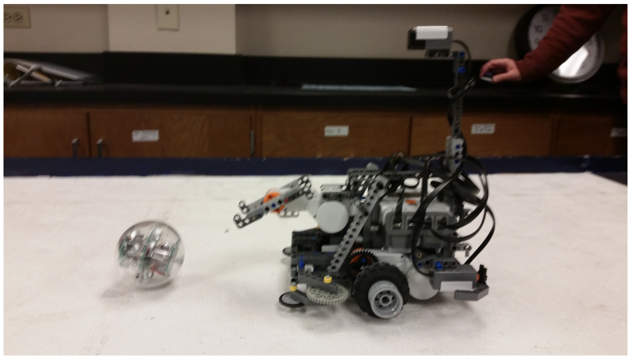

Figure 1: A winning robot!

1 Robot Design

This lab's design tried to remedy many faults of the designs of previous weeks. This was done by eliminating as much "hardware bloat" as possible while reinforcing all the key points of the hardware and simplifying the software. Simplicity and robustness was the key to success of this design.

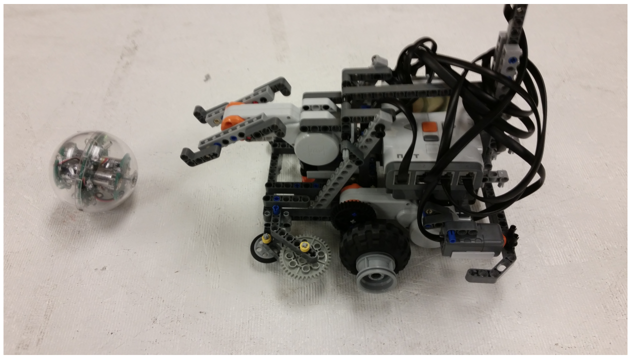

Figure 2: Gadnuk, Breaker of Worlds

1.1 Hardware Configuration

To achieve our design goals all sensors were removed unless absolutely necessary. The multiplexer was removed and only two touch and one infra-red sensor was used. The castor ball was also removed because it used too many pieces and added weight. A kicker was added to provide a bit more momentum to the ball and to confuse "naive" ball tracking algorithms. The robot was also geared for increased speed to be quicker than the competition. Lastly, wheels were add to each edge of the robot to allow it to not get wedge.

1.1.1 Sensors

Only three sensors were used because of the computational limitations of the NXT. Additional sensors, we learned from previous lab made the robot unreliable. This lab required only the use of an IR sensor to detect the ball so by removing other sensors we avoid all the complexity of having to consider all those inputs in our algorithm. Touch sensing was added to reliably detect when the robot had backed into a wall. This was not necessary but it helped tremendously when the robot got into an unknown state. Finally, a compass sensor was used to help the robot orient itself appropriately and detect when the robot was facing the wrong direction.

1.1.2 Caster

The caster ball was removed because parts were needed to add the kicker. A single part, the tip of a Lego beam, with minimum friction was added to the rear in its place. This part was added to to reinforce the binding of the NXT brick but it served a dual purpose of replacing the caster wheel. This was possible due to the smooth surface of the soccer arena and the smoothness of the plastic Lego piece used as a rear support.

1.1.3 Kicker

A kicker was added because of the old soccer adage, "the ball is faster than the man." The most common scenario was that one robot would beat the other too the ball. However, beating the other robot alone would not guarantee that it would score. The robot has to be able to impart its momentum and send the ball going in the desired direction. A simple collision would often be enough but sometimes the ball would stick to the front of the robot. Adding a kicker would ensure the ball never "sticks" and guarantees momentum transfer one way or another. When momentum is transfer the adage becomes true. The ball moved faster than the ability for most robots to track. This is because hitting the ball ahead of our robot forces our opponent to relocate the ball thus confusing them. If we had used only a ram, we had the potential of making the situation easier for our opponent because the robot would be holding the ball in front of the opponent and which ever robot had the higher torque would have likely won the ball. The kicker prove especially useful against "naive" solutions that did not deal with scenario where the ball was behind an opponent's robot.

1.1.4 Gearing

Our design was geared using 38t and 20t for a ratio of 10:19 nearly 1:2 which placed Gadnuk among the fastest robots. We chose to gear for speed because we wanted to beat other robots to the IR ball at the start of a round; further, because our robot was one of the heaviest so we had to increase the gearing to drive at similar speeds.

1.1.5 Wheels

Wheels were attached to all edges of the robot to allow it to roll off walls. This was particularly useful when the ball stop right in front of a wall and our robot came at from an angle. Our robot was able roll off the wall on most instances. The one instance it was not able was when the touch sensor was suppose to detect our robot backing up but failed to.

1.1.6 Gears

The gears of the robot did not sit flush. So axelrod of the wheels were warped and that caused our robot to wobble from left to right. This was an issue because a better solution to mount the gears was not found.

1.1.7 "Four Wheel Drive"

The axelrod being warped caused the wheels to fall off the robot when driving. To solve this issue an extra long axelrod was fitted. Attached to the rod were four wheels, two on each side. This prevented the wheels from popping out but now the axelrod would pop off so three clips were added to each axle along with a key pin between the two axles to hold everything in place.

1.2 Software Design

The software design featured the same "pub-sub" architecture as our team had used in previous projects, but also an incredibly simple controller architecture.

1.2.1 Publish-Subscribe Architecture

As it was with previous designs, a publish-subscribe pattern was used. That is to say, each sensor and the controlling logic ran in their own threads. Sensor readings were "published" by storing their data in sensor-data-typed structures. The controller thread "subscribed" to the data by polling the updated structures as appropriate. To save system resources, sensor threads could be deactivated when reading from a particular sensor was no longer necessary.

1.2.2 Regular Game-play Controller

For regular game-play, the controller alternated between two behaviors:

- Charge the Ball
  The robot would turn to face the ball and charge it continuously. A "dynamic" turning technique was used so that the robot could always be moving towards the ball so as not to waste time. This behavior would check periodically to ensure that the robot was not stuck on an obstacle or facing the wrong direction.

- Go Back To Start
  The robot would turn to face the enemy goal and then reverse until it hit a wall in order to reposition itself to get a full view of the field as well as be in good position to safely attack the ball.

There were a few other supporting lines of code used to support the above methods (to move the robot into initial position and the like), but the controller really was that simple.

1.2.3 Penalty Kick Attack Controller

For penalty kicking, the robot simply swayed back and forth a little to add some variability to the trajectory of the kicks, and then the robot would drive the ball into the goal. This was very successful (100%!) and even allowed for some game time variability when necessary since the trajectory of the kick could be controlled by placing the robot closer or further from the ball at the start of a kick.

1.2.4 Penalty Kick Defense Controller

This controller simply caused the robot to turn 90 degrees to use the robot's side to block, and then the robot moved forward and backward random distances to guard against kicks. This behavior was chosen because relying on sensors to track the ball and move appropriately had results that were just too slow to be desirable.

2 Issues With the Robot

The issues with the robot were minimal. These would be limited to things that were largely out of the scope of the project: limited computational speed and power, refinement of the controls, improvement of sensor readings, improved motor control, and so on. But these issues were minor since Gadnuk won the tournament and only scored a few self goals, so clearly our issues were less than those faced by other robots. Gadnuk's "go-back-to-start" behavior likely eliminated many of the issues faced by other teams' robots.

3 Unsolved Problems

Since Gadnuk was the champion robot, there are not problems to solve!

...Just kidding! Actually there were only minor problems that needed to be addressed. First, control could have been better, it would have just taken more time and possibly more advanced control techniques to refine the robot's control. Second, the robot did get stuck during some turns, so increasing the wheel power may have been desirable just to break out of turns.