Team Four Paper

Overall Strategy

Coilette is a bulldozer robot, with a shaped shovel in the front designed to gather balls and ram them into the mouseholes. Designed for turning in place and high speed forward movement, the robot was controlled by a series of behaviours to gather balls and deposit them in the goals.

Mechanical Design and Sensors

We chose to construct Coilette out of foam PVC due to its strength and ease of manufacturing our design. In particular, using a heat gun we were able to create arbitrary curves in the material, which we used to sculpt an outer circle, a raised front array of sensors, a strong angle bracket for the compute, and a complex curved surface for the shovel. According to usplastic.com, "this PVC sheet has excellent corrosion resistance and weather resistance. The working temp is 33 deg F to 160 deg F. and the forming temperatures of 245 deg F. It is good electrical and thermal insulator and has a self-extinguishing per UL Test 94. PVC applications are almost unlimited. It's the most widely used member of the vinyl family. It is excellent when used for corrosion-resistant tanks, ducts, fume hoods, and pipe. Ideal for self-supporting tanks, fabricated parts, tank linings, and spacers. It is not UV stabilized and has a tolerance of +or 10%. Not FDA approved materials." The only reservations we have about this material are the following:

- It produced itchy white fine dust wherever we machined it. This dust was particularly annoying when it got into our eyes.
- A TA at maslab, told us we will probably get cancer from it.

  Besides these problems, it was an awesome material.

Our design had no motorized parts except the initial pair of wheels, so we used our sensor points to make a large array of IR sensors in front to keep it moving without colliding much with the walls. We wanted to go fast and be cool like sonic. Sonic has no motors!

We chose to build a cylindrical robot because it enforces better configuration space modularity and ensures turning in place is safe. There were some implementation details that caused this to only be partially true, but for these cases it was able to escape with a simple left and right turning timeout. This shape did not guarantee our robot would not get stuck, but it certainly helperd us out a little!

The sculpted front shovel was designed to match our desired mobility. The back side of the shovel had a depression so that the lowest potential for a given ball is at the center of the shovel. That way Coilette can expel balls into the goal by driving straight and stopping rapidly. Turning moves the balls to a particular side of the shovel, so wings were added to the front sides to prevent balls from being lost when turning. It worked well for our purposes, but testing showed sustained turning wasn't able to control more than 2 or 3 balls.

The front had an array of IR sensors to facilitate forward movement. 6 IR sensors: 45, 20, 0, 0, -20, -45 degree angles respectively. The two straight sensors are at different heights to find goals without need for vision (one is above 6" and the other is below -- so they will return significantly different values when pointed at a goal). Due to time constraints, they were merely used for redundancy in our wandering algorithm, as we only used vision for goal detection.

Software Design

Our software was written as a single threaded driving loop that queries its input devices, IR sensors and processed images, and then acts on the first behaviour that subsumes the robot at that time. For this project our behaviours were simply layered Boredum, Score Goals, Chase Balls, Wander with the first behaviour that has an action controlling the robot. In heinsight, swapping the order of the goal scoring and ball gathering may have led to better performance, since we would attempt to have all known balls before going after the goal.

AI Behaviours

1. Boredum -- Boredum's purpose was a catch-all errors kind of thing. It was added to improve our performance with blind-spots and turned out to interact well with scoring goals and gathering balls near walls. Boredum used the difference between every few processed camera image to determine whether the robot was not moving. This was chosen after current sensing proved unreliable without excessive modeling of the robot's current state (desired PWM to motors, current voltage of the batter, averaging over time, etc). A quick and dirty implementation of image difference turned out to be more than enough because it had effectively no false negatives (it wouldn't subsume the behaviour layers in the middle of gathering a ball, for example), and the false positives were recoverable (if the image was changing enough, even though it was stuck against a wall, it generally meant that while the robot was dragging one side against a wall, it was still moving forward). However, as I noticed during the exhibition rounds, it turned out not to be generalizable to other enviornments. In particular, it would sometimes observe other robots movement and believe it had made progress. I believe with an improved image differential algorithm, it would be able to overcome this.
2. Score Goals -- If the camera sees the top bar of a goal (best fit by a 7:1 ratio yellow bounding

box) it accelerates towards the center of the goal -- and the first frame it drops out of vision off the top, it tops out its speed and all non-boredum behaviours sleep for half a second. This worked because of the design of the shovel in the front -- the balls would gather in the middle of the backside because of the forward momentum and some would roll out upon collision. Limitations of the motors speeds prevented the proportional controller from ever reaching a point where we needed to make it a differential controller, since the motors' top speed was our coefficient. It did appear to hit off-center on the goal fairly regularly, so perhaps it did need an integral controller. But what we had was simple and worked for our purposes.

3. Chase Balls -- If the camera sees a red ball (best fit by a 1:1 ratio red bounding box) (selecting largest if multiple observations occur) it drives proportionately to the x-coordinate of the center of the region. The actual output to the motors was very similar for ball chasing as for goal chasing, but since balls remained within its field of view until they were inside the shovel, it didn't have any problems not being centered like the goal scoring behaviour did.
4. Wander -- If any of its sensors sense a nearby wall it turns away from the wall, otherwise it moves towards the largest measured distance. While we didn't use PID controllers for any of our layers, for this behaviour it is important not to use one because the camera is our primary means of observing balls and goals it is important to change what it is looking at when nothing interesting shows up. This also results in a more thorough observation of the field than a wall-following or center-following algorithm.

Action Selection

One behaviour drives the robot at any time -- the first behaviour that returns a desired action in the order listed above; however, if either desired motor speed is non-negative and lower than its current motor speed, both motors speeds are weighted with the current motor speed. The purpose of this is to prevent sudden stopping as a result of AI decisions.

Camera Input

Image Processing

We performed a decent amount of precomputation on our image and reduce its data down to a managable size for the behaviours to interact with. First, it does horizontal edge detection in the HSV domain by looking for significant differences in the any of the three (hue was the most discriminating, as individual objects are all monochromatic, value was second most to distinguish between two balls that are overlapping in our vision). Then we have a large number of line segments, the color for each was averaged to reduce the amount of noise in the color system and enhance the differences between objects. Finally, a color lookup table converts each color to an enumerated list of colors to be used for data association (Red, Green, Black, Blue, White, Yellow).

Data Association

Image processing returns a list of line segments in colored buckets and a count of how many line segments have which x-coordinate as their center. Data association attempts to match up multiple line segments of the same color that are centered roughly at the same height. For the actual competition, we only extracted 3 types of data, balls (1:1 red), goal sides (1:3 yellow) and goal tops (7:1 yellow). But in the past it extracted barcodes as well by the same system.

Overall Performance

Coilette placed 3rd in the final competition, but more importantly she performed in a manner we felt was more interesting than other robots. She may not have gotten the same kind of reaction from the audience (due to the continuous nature of Coilette, no particular instance was particular amazing or hindering), she was the only robot to thoroughly explore the map, the only one to score a ball through the mouse hole, and managed to compete against robots that were much larger than she was. Because she is so small, and not hindered by mechanical mechanisms, she will be easy to mutate into an awesome pet robot.

Conclusions/Suggestions for future teams

One of the biggest downfalls of coilette was the small opening for gathering balls. While chosen specifically for this robot to improve ball control with no motors, it really couldn't gather balls as quickly as the ones that had a foot wide entrance. But it sure was cool to have an awesomely small robot! Also, it is unfortunate that we will get cancer, according to TA.