The Details of Phoenix Claw

Stephen M. Hou, Chun-Chieh Lin, Tokumasa Yamashita
Team 34

6.270 Autonomous Robot Design Competition
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

February 7, 2002

1 Strategy

We broke our robot's course into three parts: (1) a get-the-first-ball-in part, (2) a score-more-points part, and (3) a attack-opponent part. Since we wanted to be both offensive and defensive, our robot needed to be very fast. Initial testing assured us that we would have a fast robot; thus implementing our strategy would be possible.

Like other teams, we needed a guaranteed way of scoring the ball closest to the alley in order to qualify. After building an orientation-determining procedure, we perfected our rotations so that the robot would be able to score that ball in all the time.

We constructed paths for the robot to follow after the first ball is in based on the location of the blue block. In each of these paths, the robot scores three or four balls. Thes robot traverses the matrix that contain four balls and deposits them into the alley. Since our robot used a fence to collect the balls, we needed to avoid rotating while holding balls. In the cases where a single ball was held during a turn, the rotation was successful. However, one of the cases required turning with three balls, which proved to be very difficult task, and led to a defeat in the actual competition.

Because our robot was fast, we believed that we were able to score our balls before our opponent would score more than one or two balls. So, our robot was designed to line-follow along the black line closest to the alley and align with the other robot. When the opponent is within a particular range, our robot would chase and hit it, preventing it from scoring more points. This section was not implemented until the last day of work and was not tested. The competition indicated that this did not work. Fortunately, the failure of this section did not cause us to lose any matches.

During the planning process, we had considered using a "tether-bot," or a miniature robot connected to the main robot by a custom-made tether and that would block the tubes. Other teams had successfully tested a tether strategy, so we also attempted one. Unlike other teams, however, our tether-bot would have no motors, so it would have to have been "dropped off" by the main robot. In the construction stage, we developed a tether-bot that could roll down the alley, but was not reliable. In the end, our failure to develop tether technology forced us to scrap the tether-bot completely from our final strategy.

2 Construction

Our basic idea in constructing our robot was to make it fast and powerful. To achieve that, we used all six of our motors to power a four wheeled differential drive in a 25 to 1 gear ratio. This setup makes turning a pain to calibrate. However, once we got through that the routines were not very hard to program since it also makes the robot go pretty straight.

In order for our robot to go fast, it must also avoid hitting the blocks and the wall. Therefore, we decided to make our robot relatively small. The biggest obtacle we faced as we tried to make a small robot was that we also wanted a fence structure to collect and hold balls with, and that would increase the size of our robot significantly. The solution we came up with was a special fence structure with its pivot point, powered by two servos, relatively high. That enabled us to set the servos to an angle that holds the entire fence structure higher than the blocks and the walls so that it does not obstruct the movements of our robot.

After some careful adjustments in the shape of the fence, we succeded in using the fence to collect the balls as the robot moves forward. Holding the balls was relatively easy as we just had to lower the fence around the balls. The fence was also used to hold the distance sensors, so we could adjust the height of the distance sensors to sense both the robots and the blocks.

With the distance sensors on each side of the fence and the ones on the front and the back, our robot is prepared to track down the opponent after it finishes its routines. We designed the back of our robot to be very sturdy, and programed it to smash into the opponent with its back. However, we did not have time to test that and there was a bug in the program so it didn't work.

3 Reflection

As many previous 6.270 contestants that I talked to testified, speed and robustness are the most important features to consider when designing Phoenix Claw. Since speed and power were our primary focus when we designed our robot, we used all 6 motors to power its wheels. This turned out to be a great advantage for us, as our robot was, in our eyes, the fastest and most powerful robot this year. Phoenix Claw could move fast and reliably score points.

When calibrating Phoenix Claw's control mechanisms, we initially coded the forward, rear and turning motions based on timing. We realized, however, that the accuracy of this method is dependent on battery power, and thus it was very unreliable. We quickly abandoned this plan and utilized the shaft encoding instead; this turned out to be fairly accurate.

As the term progressed, we noticed other teams using tethered mini-robots that block the holes, and we felt the need to add such capability to our robot as well, in order to remain on the competitive edge. Since we had already used up all six motor ports on the handyboard, we designed a "wireless tetherbot," which would simply roll down the gully if we drop it in. The tether, however, didn't work as well as we had expected, and we therefore gave up on the idea.

Although we are satisfied with our strategy and the design of Pheonix Claw, our biggest regret during this IAP was our lack of time management skills. We especially spent too much time working on the tetherbot on the night before the first round, when we should have been working on the software to get the core functions working. If we were to do everything over again, we would prioritize our tasks and work on insignificant extra features (such as the tetherbot in our case) towards the end if we have time to spare.