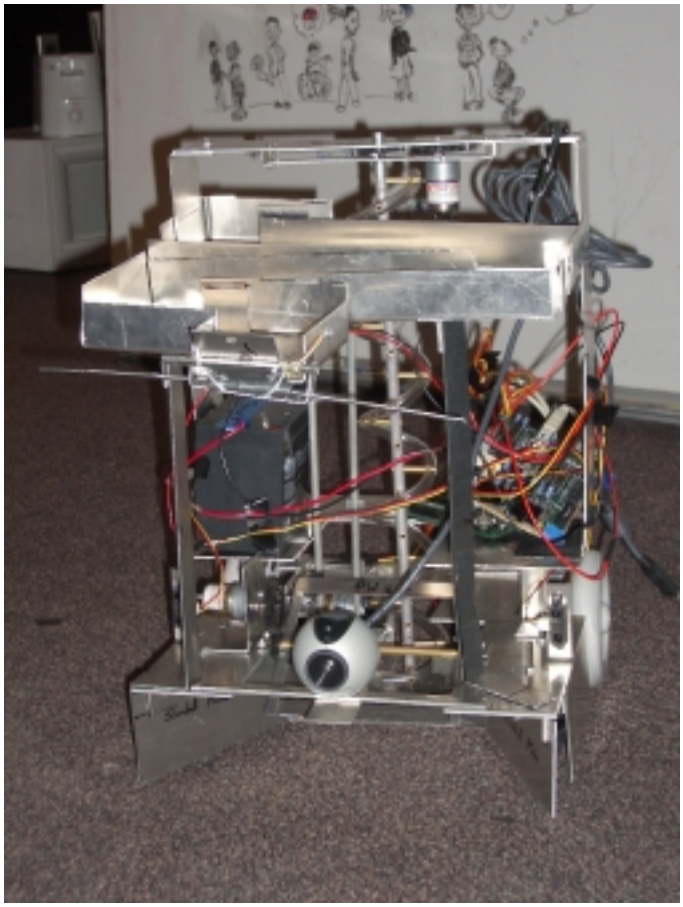
Team Five Paper

From Self Flagellation to Michael Craig: A Robot's Journey

The Bot



We named our robot Michael Craig to replace our lost teammate. Unfortunately, Michael did not perform up to our lofty expectations from the start of the course. Perhaps a stronger effort in the first two weeks of the course would have allowed us to finish the mechanical systems earlier and to make more progress on the software. Then again, perhaps not. In the end, Michael wandered in what appeared to be an aimless manner, but managed to score three points. It was lucky that we did score, because we chose to tempt fate and left the completion of checkpoint two until contest day. But we're gambling men, and such a situation was only to be expected.

Overall Strategy - Keepin' it Simple

There were a couple of things that we knew, from the start, that we didn't want to do in the contest. We didn't want to do tasks that would spend time and processing power

generating extra data. This meant metric mapping was out. The thought was that most of the details of the data we generated would simply be thrown away. Instead, we thought that the more abstract topological mapping was a better choice.

Second, we didn't want to design our robot in such a way as to require accurate movement and positioning. This meant that an arm was out. We wanted our robot to thrive in the error-filled environment that was sure to ensue. A large mouth that would allow us to drive in the general direction of balls appealed to us. A precision instrument to grab and lift balls did not (though the two arms that we did see were really cool).

Mechanical Systems

Frame

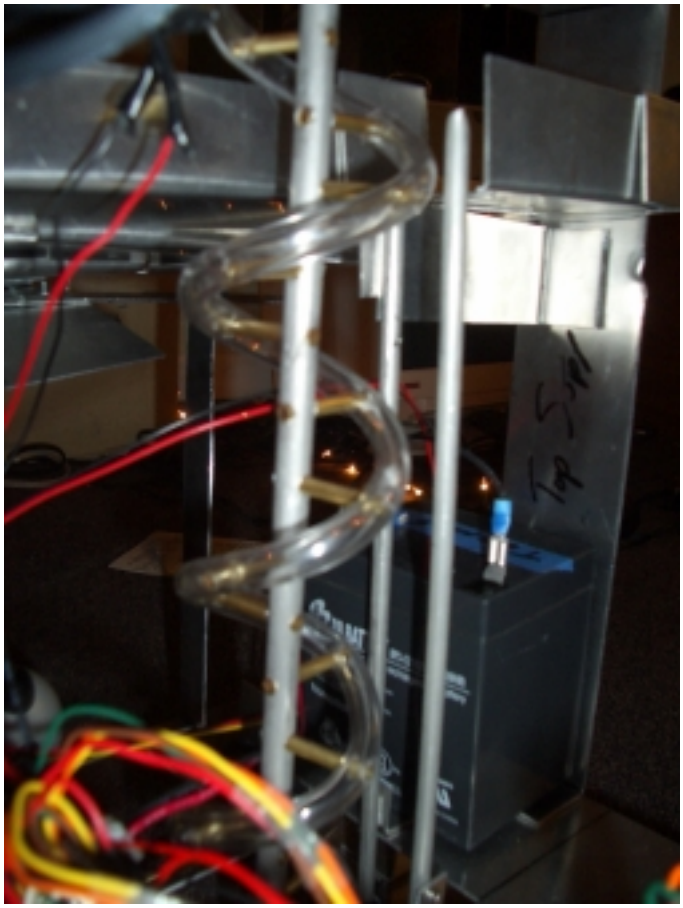
We decided to build the robot almost entirely out of sheet metal. The main reason for this was that it would look cool. Almost all of the sheet metal that we used was purchased from CMS, as MASLab did not have enough for it to be used for structural purposes. John acquired high-bond double-sided tape from McMaster to hold the different pieces together. It worked well in long strips, but was not so great for small bonds.

Paddle Wheel

Our robot divided the task of delivering a ball from the playing field to a field goal between three main mechanical systems. The first was a paddle wheel which was slightly recessed in the frame of the robot. We recessed the paddle wheel in order to keep our camera as close to the ground as possible. To clear the top of the wheel, the camera would have had to be about 5-6 inches high. By recessing the wheel we were able to get it to only about 3 inches off the ground. We decided to use a paddle wheel rather than simply driving over the balls, because we decided that it would make it much easier to feed balls into a lifting mechanism if they were on an incline. In order to accomplish this we needed to push them up a small ramp.

We constructed the paddle wheel from small strips of sheet metal rotating around a section of brass tubing. We drove it with a high torque motor, which in retrospect was overkill. One problem that we foresaw was that at certain angles the wheel would push balls straight into the ground. We spent a lot of time worrying about this, but it turned out not to be a problem at all. Since we were always driving forward over balls, they never once got stuck during testing or the contest.

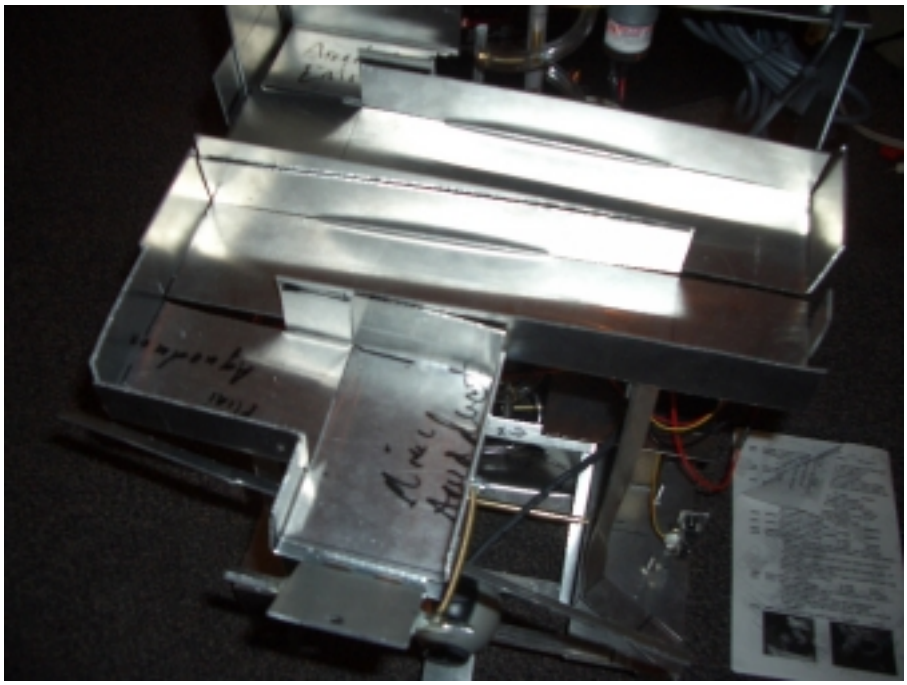
Screw



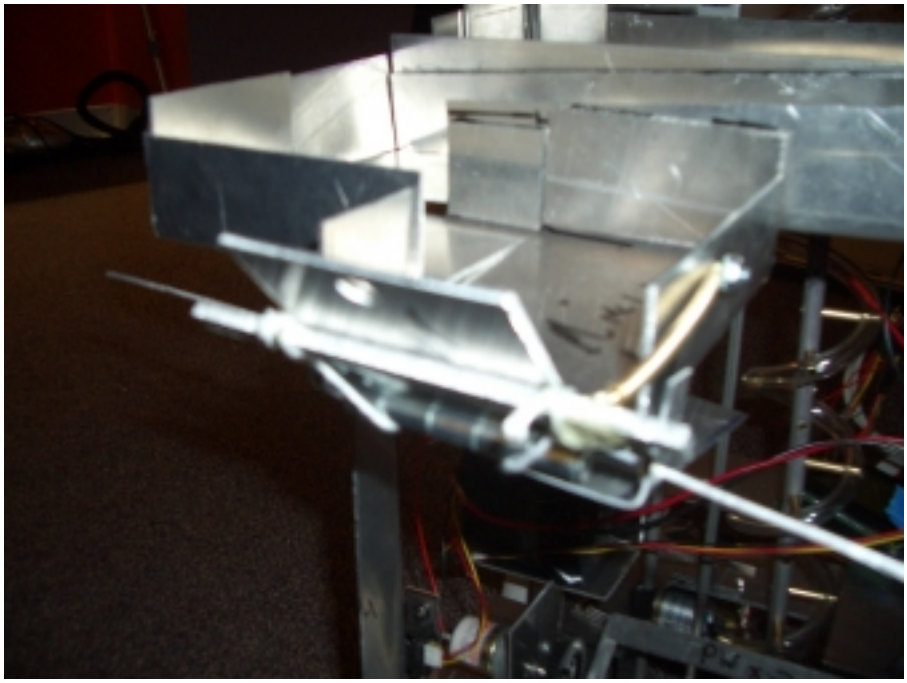
The second mechanical system was an archimedean screw. We used a single screw which trapped balls against two bars. The shaft was aluminum with 1 inch brass supports. The

helix was constructed using surgical tubing from Home Depot. We drilled holes in the tubing and superglued it to the brass supports. The screw was also driven by a high-torque motor. It was able to elevate four balls at once.

Aqueduct and Trapdoor

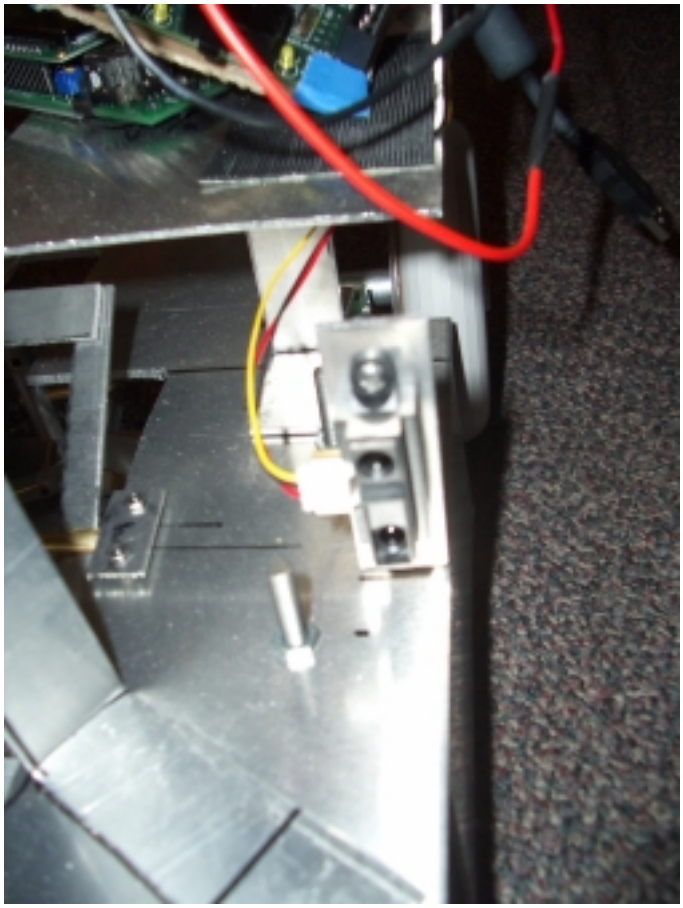


Once again, aesthetics and bad-assness played a major role in our design decisions. Rather than make a simple tilted container for the balls, we decided that it would look much better to have a long, winding aqueduct on top of the robot. The initial idea was to use PVC piping, but John decided that bending yet more sheet metal to his will would be even better.



The trapdoor was the result of running out of sensor points. Ideally we would have used a servo to open and close the door. Instead, we had a mechanical system that was improvised on the last day before impounding. It was a one-shot affair, with a spring holding the door closed and a bar acting as a switch. When the bar hit a field goal upright on either side, it pulled the trap door open.

Sensors



We didn't use many sensors, because we had used most of our points on the two high torque motors for our capture and elevation systems. We had two short range IR sensors on the sides of our robot. These pointed forward and were mainly used for obstacle avoidance. We also used optical encoders to help with straight driving.

Software Design

Robot Control

We used optical encoders to dynamically adjust the speed of each wheel when moving forward and backward. We decided that a PID controller was not necessary since we ultimately could drive straight without it. For turning, we simply used the gyro. This was fairly accurate for angles between 30 and 90 degrees. It was not accurate for small angles except on very low speeds. This was because we could only access a gyro reading 20 times a second. For medium to high speeds, this made the error on angles under 20 degrees prohibitive.

Robot control turned out to be one of the biggest surprises of MASLab. We realized it would not be simple, but it turned out to take up an enormous amount of our time. Simply driving straight and turning a requested angle were not simple tasks. The motors never behaved how we thought they would. We inititially did not want to use encoders, but soon

realized that if we did not, we would not be able to rely on the robot going where we wanted it to. A pleasant surprise was that they worked really well, and we never needed to fiddle with them.

We decided fairly late in the course that we would use current sense to detect bumps. This introduced a whole new set of problems. We had a set of constants calibrated for each speed, which indicated when the current was a result of normal driving and when it meant that the robot had hit something. Every time we made a major change to the structure (which was all the time during the last week) we had to fix these constants. In short, driving was a major headache and seriously impeded our progress in other areas.

Image Processing

The image processor on the robot ran at a peak of 12 frames per second. The final speed was lower than this as we started scrambling during the last week and never got around to optimizing. Barcode recognition proved to be the only non-trivial task in this area (in fact, it was very far from trivial). We ended up using the low-resolution image to define the area of a barcode, and then taking a high-resolution picture and analyzing only that area. A problem we ran into was that if we were moving, the high-res picture would be slightly different than the low-res. The only way to fix this was to stop when looking at a barcode. We didn't want to do this, but we had to, as correct barcode recognition was absolutely essential to our mapping system.

Another feature of our image processing was that we were able to estimate the distance and angle to an object based on its size and position in the picture. The estimates were accurate enough to get balls and drive near barcodes.

One thing our image processing never managed to do was to determine what the orienation of a goal was. This didn't turn out to be a problem as we never got close to scoring a field goal.

Mapping

We decided against metric mapping. In fact, we never had any desire to do anything close. Our biggest thought on metric mapping was that we would have to pay an enormous amount of time and processing power to generate and upkeep a set of specific data. The problem was that in the end, this data didn't seem all that useful. Instead, we sought to create a topological map of the playing field. We would note the different barcodes and connect them using breadth-first search. We had plans to note the ball density around each barcode, but this went on the backburner.

We ended up with accurate mapping that correctly determined the barcodes present and a

good portion of the links. However, due to lack of testing time, we never used our maps. We were concerned about our ability to reliably travel through more than one link. Because we didn't test, we also weren't sure it would save time over random wandering.

Scoring

Our initial plan for scoring was to use our map to ensure that we travelled to all of the rooms on the field. We would weight the different barcodes on the map based on their:

- ball density
- proximity to robot
- proximity to a goal

The third value would be weighted higher as the round progressed. We would use this to determine which barcode we should travel to next (and the current room would be included in this computation). At a certain point, we would drop everything, go to a goal, and try to score.

This all went out the window when we realized we were out of time. We ended up wandering randomly. With another 1-2 days, we could have figured out some basic uses for our topological map. With another 3-4, we could have generated this complex system to compute the value of each of our options at any given point.

Suggestions for Future Teams

Driving Good is Hard

Do not underestimate the problems you will have in this area. Get encoders, even if you are not doing metric mapping and don't care where on the field your robot is at any given time. You will care about driving straight at some point, we guarantee it. That is all we used our encoders for, and they were invaluable.

Be prepared to spend tons of time on this.

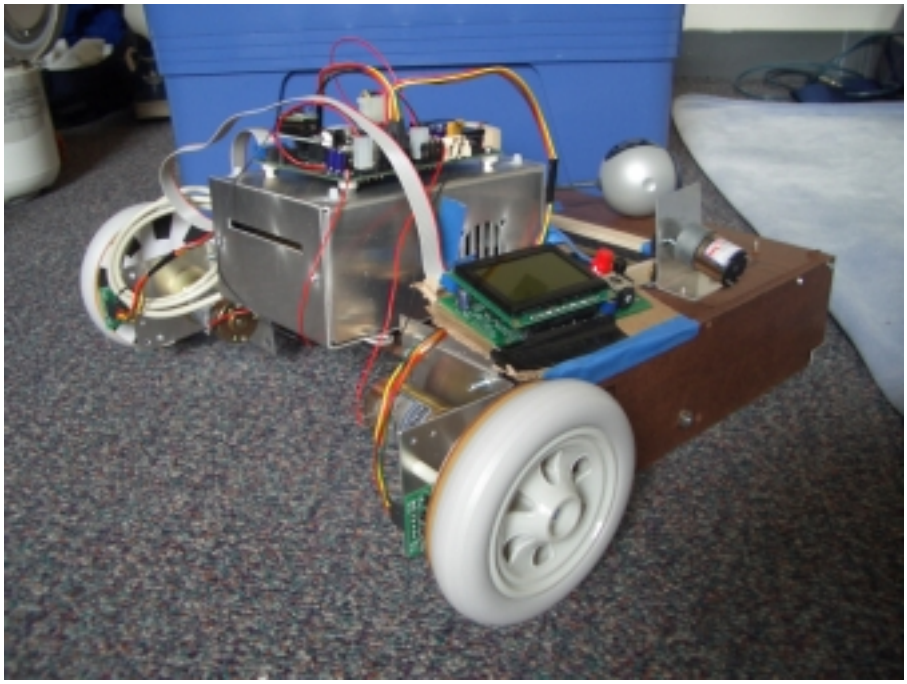
Get Your Mechanical Stuff Done

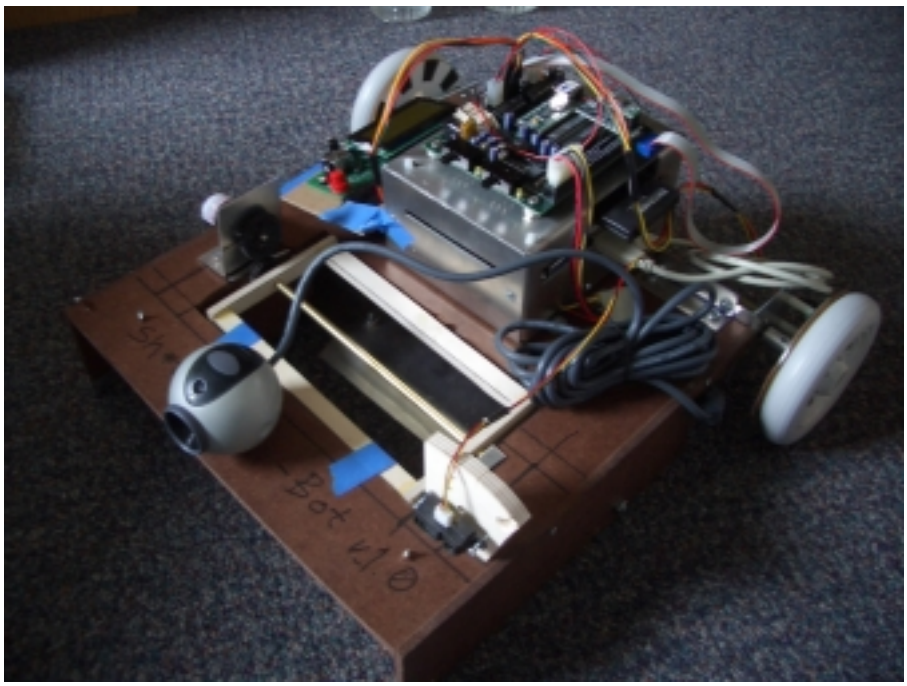
One of our biggest problems was that we couldn't do testing. Our pegbot had issues that we resolved with the final design, such as wheels sticking way out of the frame and getting stuck on stuff. As a result, we couldn't do useful testing of our software for much of the time we had. We finished our mechanical systems about an hour before impounding. We tried to test as much as we could before then, but it wasn't nearly as effective as it could have been.

Having a near final robot by the end of the second week would have been real nice.

Small and Simple

Our robot, though it performed all of its intended functions well, was too cumbersome and too slow. If given another chance, we would have ignored field goals and gone for a short, light, fast robot. To succeed in this contest, one doesn't need to do it all. Just make sure you do what you do well. Here's an example of a robot that didn't do anything well. Our prototype, lovingly dubbed "Sh*tBot v_1.0". Notice the distinct lack of front wheels, and the excess amount of masking tape, often serving structural function.





Take Advantage of the Staff

Yes, this is exactly what it sounds like. And in addition, the staff is amazingly helpful and friendly. Don't hesitate to ask them for help when you get stuck.
