Rocket Power Technical Report

Sam Parker, Bobby Pender, Abby McCall, James Carpenter, Nick Maruyama & Megan Hodge

1 Abstract

Our objective this year was to create a robot for the FTC competition HotShot!, which involved shooting Whiffle balls into a variety of different goals. We created a conveyor belt with a ramp to pick the balls up and get them into a holding box. We built the box with a hole in one side with mesh netting attached so that the balls could travel down to reach the shooter. At the shooter the balls would be propelled forward by passing through two wheels spinning in opposite directions. On the day of the competition our robot ended up not working at all as we had expected. Our shooter had a loose chain and a lot of friction interfering with the DC motor. We decided that we should have gone with a simpler design because too many problems came up with ours. We also decided that we should have tried various different approaches rather than just revising the same idea over and over again as we did. Our strategy was constantly refined during the competition as we adapted to mechanical problems and varying opponents, and was ultimately a strong point for our team.

2 Introduction

Over the course of three and a half months, our team designed and constructed a robot to compete in the 2010 First Tech Challenge. The challenge this year, called HotShot!, consisted of a 30-second autonomous mode and a two-minute tele-operated mode. At the start of the game each robot, two per team, was preloaded with up to eight balls, and during the autonomous mode a team could score points in four ways: by hitting the ball chute lever or by scoring in the one-point goal area, the raised five-point goal, or the ten-point off-field goal. The autonomous program proved essential, as points scored in autonomous

---

This paper was written for Marc Allard's Robotics class in the spring of 2010.

mode were awarded double point value. The basic idea for our robot included the abilities to pick up the balls, store them, and consistently shoot them into the upper 5-point goal and the 10-point off-field goal. Our final design was only able to score in the one-point goal, but due to our consistency and our understanding of the importance of strategy, scouting, and defense, we had great success at the FTC Challenge.

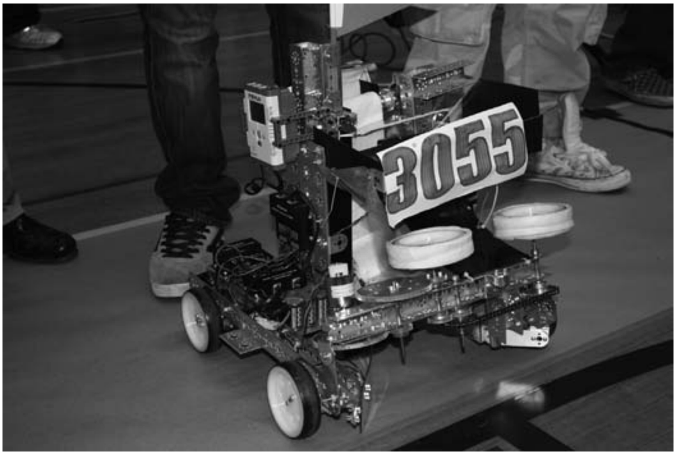

Figure 1: *The robot at the competition*

3 Robot Overview

3.1 Specifications

Size: 17 in x 17 in x 16 in
Weight: 15 pounds

Motors: 8 DC Motors
- 4 to power wheels in All-Wheel Drive
- 2 to power conveyor belt
- 1 to power shooter
- 1 to power spinning ball accelerator

Motor Controllers: 4 DC Motor Controllers
Wheels: 6 Full-Size Wheels
- 4 for locomotion
- 2 for shooter

3.2 Key Features

The first main part of our robot was the conveyor belt. It enabled us to lift the balls to the storage area and the path to the shooter. Figure 2 shows how we mounted the polycarbonate material in a way that allowed it to flex while still keeping pressure on the ball moving upwards. The polycarbonate is only attached to the frame of the robot at the bottom, so the top can move up to an inch to accommodate a ball coming up the conveyor belt. The DC motors used to power the conveyor belt are also visible to the left of the belt. The black plastic was added to ensure that balls did not fall out the sides of the conveyor belt while traveling up.

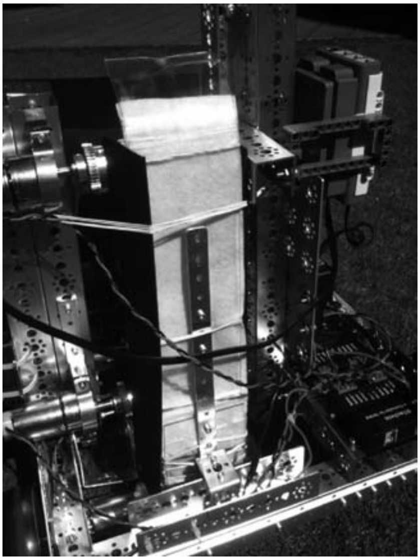

Figure 2: *Conveyor belt*

The next main element of our robot was the storage box used to hold the balls on the robot. It was mounted near the top of the robot so that once the balls were released they could roll to the shooter using gravity, gaining momentum and eliminating the need for another motor. Balls exit the top of the conveyor belt and fall directly into the box, which can hold up to five balls. In Figure 5, the hole seen in the bottom of the box on the left side allows balls to fall into the path towards the shooter. This easy design worked very well for us, even though we could not regulate when the balls would fall through.

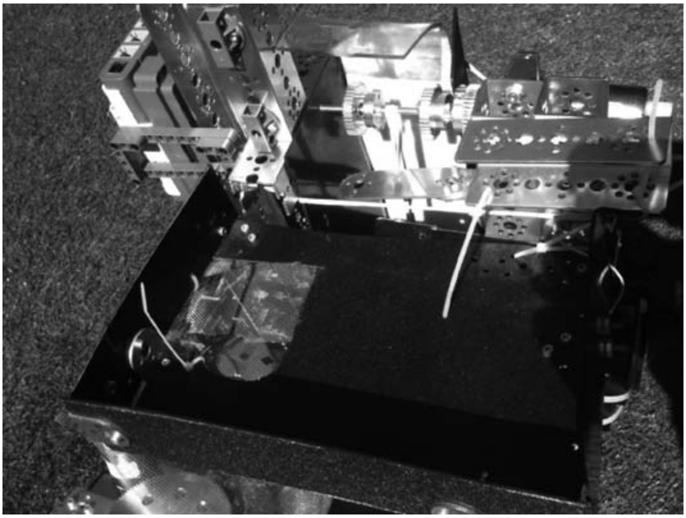

Figure 3: *Storage box*

The last critical part of the robot was the shooter. The shooter operated on the same principle as a baseball pitching machine: two wheels spin very fast in opposite directions so that when a ball rolls between the wheels, it is shot out at a high speed. For this design to work, we needed to make sure that the wheels had just enough space between them to fit a Whiffle ball, but not so much that the ball could avoid one of the wheels and not be properly shot out. We also needed to gear the wheels 9-to-1 against the motor to make sure that they spun fast enough. It turned out that the only way to meet both of these criteria with the gears and wheels we had was to use a chain and sprockets to connect one wheel to the other. The chain, which can be seen in the bottom right of Figure 4, was the most troublesome part of the robot. In order to work it needed to fit perfectly onto the sprockets, but our chain was a little too loose and kept falling off. Because of this and the fact that our gearing system had too much friction, the shooter often got jammed, damaging the motor. We ultimately decided to get rid of the shooter by removing the left wheel, as seen here. This way the balls could just roll out of the front of the robot into the one-point goal.

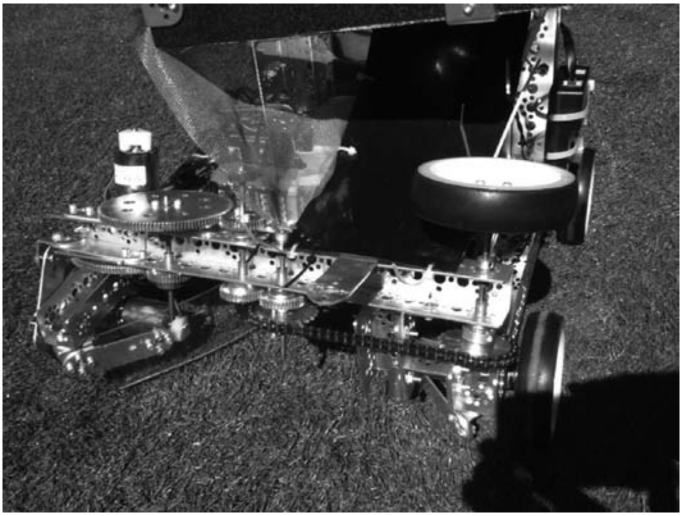

Figure 4: *Shooter*

4 Design Strategy

Our design focused around our shooter, which we expected to be able to shoot into the off-field goal. In order to make this possible, our first problem was being able to pick up balls and raise them so that we could store them high on the robot before shooting them out near the bottom. Our solution was to create a conveyor belt out of non-stick material that pinched the balls and raised them to the top of the robot. On one side of the passageway was a belt of non-stick material powered by a DC motor at each axle. On the other side was a polycarbonate board covered with a layer of nonstick that was mounted to the main frame only at the bottom. The passageway was not quite large enough to fit a ball all the way up, thus forcing the polycarbonate to bend and keeping pressure on the ball. In this way we were able to get enough tension on the ball to raise it with the conveyor belt.

We had trouble getting the balls to get caught on the conveyor belt, so we added an extra DC motor with more non-stick material attached to shoot the balls towards an incline into the conveyor belt. This worked by applying a lot of small forces on the ball in the form of spinning pieces of fabric, with the end result of greatly accelerating the ball towards the conveyor belt.

The storage space for the balls went through many different theoretical designs. Our first idea was just to build a box of plastic that could hold about eight balls, with a hole in the bottom where balls could drop through to the shooter. Then we thought this was not effective enough at holding balls, so we drew up a system of rails that could transport the balls downwards in either a spiral or a snake pattern to the shooter. That, we decided, was far too complicated, so we also considered a straight rail from the top of the conveyor belt to the shooter, which would be fast but couldn't hold many balls. Eventually we came back to the box idea and finally built and mounted the box onto our robot.

Lastly, we experimented with a number of ways to feed the balls into the shooter. Our main problem was that the hole in the box was not in line with the center of the shooter where the balls would have to end up. One of the proposals was a bent plastic ramp between the hole in the bottom of the storage box and the shooter. We weren't able to secure our plastic to the robot in a way that would make this possible, so we next thought of trying to bend metal rails to guide the balls to the shooter. However, the rails proved unbendable, so we tried to use a loose net to funnel the balls straight to the shooter. The net was too loose, however, and ended up just catching the balls in a slack spot and not getting them to the shooter at all. We finally decided on a net to let the balls run sideways and down from the hole in the box to a plastic ramp, which led into the shooter.

The shooter was always a source of trouble to our team because of the friction in so many spinning axles. Also, the chain and sprockets did not mesh as tightly as we wanted, so the chain was liable to slip off the sprockets or jam in the teeth. After borrowing some lubricant from another team, we were finally able to get our shooter working smoothly. However, the shooter was almost parallel to the floor, so it wasn't able to shoot into the off-field or high goals. For our first match, we planned on using the shooter to shoot balls into the low goal, but the balls ended up going much too far or rolling out of the low goal. Since we didn't need the balls to be shot into a goal on the ground, we just took a wheel off the shooter and took its coding out of our programs. This way, the balls could simply roll out of the front of the robot straight into the low goal. Although anti-climactic, this strategy ended up being our best way of scoring points

5 Software Strategy

Our autonomous program was designed to move the robot directly towards the center goal and unload all our balls into the one-point goal. We decided on this strategy because the one thing our robot could do consistently was move and let balls roll out of the front end. We thought it was more valuable to get five balls into the low goal than to try to hit an elbow and release the ball chute, because the balls would be worth an additional five points at the end of the match. Our program coded for each of the wheels independently using wait commands, as that was the easiest way to make adjustments on the go. Our robot would immediately pivot about ten degrees to the left, then move forward for three seconds to stop right in front of the low goal. This movement jostled free the balls we had stored in the storage box, so they would fall through the hole and roll out the front of the robot.

The main objective of our tele-op code was simply to code for each function of the robot with a separate area of the controller. First, we set all the wheels to respond to one joystick to keep things simple for our driver. A few of the functions of the robot we decided should always be running during game play; these included the spinning ball pusher and the conveyor belt, so those were set to start running at 100 percent from the beginning of the program and never stop. Other functions, like the shooter, were more advantageous when turned on and off, so we had separate buttons on the controller. One set the shooter to 100 percent power, one set it to 50 percent power, and one turned it off. When we eventually decided to dismantle the shooter, we deleted all of the code to run it. Our tele-op program worked to perfection in all matches.

6 Robot Characterization

In order to create the graph in Figure 5, we had our robot run for four seconds at a certain power level on all wheels. We measured the distance that the robot traveled in those fours seconds, and then used that value to calculate the robot's average speed in feet per second. The averaged values shown in the graph below are an average of the three tests run at each power level.

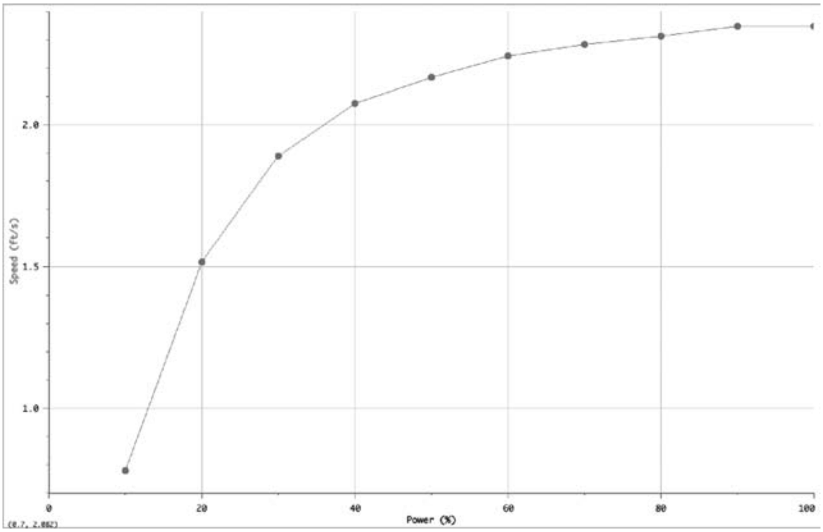

Figure 5: *Speed vs. Power*

7 FTC Result

Our robot came into NorCals without any testing, so we were learning on the fly. In our four qualifying matches, we won two and lost two, but one of our losses was originally a win before the head referee overturned a disqualification for the other team. We finished in 19th place out of 34 teams. We had a number of problems during the day that affected our design. In our first match, a plastic piece of the robot got caught in the foam floor of the arena so that our robot could only move backwards. Our driver then accidental backed it into a wall, hitting the power switch and turning off our robot mid-game. We corrected this after the match by raising the robot to give it more clearance and moving the power switch to a more protected position. We also ran into problems with our shooter in the early going and decided to take it off and change our strategy a little. Removing the shooter was a last-minute decision but it turned out to be a good one. Rather than shoot the balls, we dropped them into the goal during the autonomous stage. We also had some problems with the ramp to the conveyor belt because balls got jammed inside, making it impossible to pick up and shoot any loose ones in the field. We had to rely on our four preloaded balls and our teammate's balls for points, which in the end was not a reliable enough strategy to win.

8 Conclusion

From our experiences in class and at NorCals we learned a lot about what we should have done in building our robot. First, we made our robot too complex, causing it to be very unreliable and prone to failure. After NorCals we realized that the simplest robots were the most successful, so we should have designed the robot in the first place to only go for the one-point goal. By removing the need for a shooter and a space-hogging conveyor belt, we could have made the robot much more reliable and efficient. Although our tele-op software was consistent, our autonomous program was inconsistent and only ran for about 12 seconds. We now realize that autonomous was crucial to win matches because all points scored are worth double. If we could rewrite our software now we would make a consistent program that dropped all the preloaded balls into the 1-point goal. After scoring, the robot would back up to hit the ball chute lever for another 5 points. If we had gotten this down, we could have scored 21 points in autonomous mode, which alone would have been enough to win several of our matches. Strategy was even more important than we had thought, and we ended up spending most of our time at NorCals scouting other teams. By scouting other teams we were able to defend against their strengths both in tele-op and autonomous modes with great success. ●

9 Appendix

Bill of Materials • FTC Robot, Team 3055 Menlo School • 2/18/10

Level	Part No.	Description	Quantity	U/M	Unit Cost	Total
1	W739074	TETRIX Tubes	2	80mm	$1.58	$3.16
1	W739073	TETRIX Flat Building Plates	3	each	$5.98	$17.94
1	W739069	TETRIX Channel	4	416mm	$7.98	$31.92
1	W739068	TETRIX Channel	3	288mm	$11.98	$35.94
1	W739067	TETRIX Channel	5	160mm	$7.18	$35.90
1	W739066	TETRIX Channel	5	96mm	$5.58	$27.90
	W739065	TETRIX Channel	1	32mm	$4.78	$4.78
	W739070	TETRIX Flat Bars	2	288mm	$4.38	$8.76
	W739057	12V Rechargeable NiMh Battery Pack	1	each	$31.96	$31.96
	W739025	TETRIX Wheel	6	3” diam	$15.96	$95.76
	W739088	TETRIX Axles	8	100mm	$2.39	$19.12
	W735871	TETRIX Axles	2	250mm	$4.79	$9.58
	W739172	TETRIX Axle Hubs	14	each	$3.18	$44.52
	W739091	TETRIX Bronze Bushings	15	each	$6.38	$95.70
	W739092	TETRIX Axle Set Collars	6	each	$1.58	$9.48

(continued)

Level	Part No.	Description	Quantity	U/M	Unit Cost	Total
	W739028	TETRIX Gears	10	40-tooth	$9.98	$99.80
	W739085	TETRIX Gear	2	120-tooth	$23.96	$47.92
	W739171	TETRIX Sprocket Pack	2	32-tooth	$13.98	$27.96
	W739173	TETRIX Chain with Links	1	each	$11.96	$11.96
	W739089	TETRIX Motor Mount	7	each	$15.96	$111.72
	W739061	TETRIX Flat Brackets	6	each	$2.78	$16.68
	W739063	TETRIX Servo Joint Pivot Brackets	2	each	$4.78	$9.56
	W739062	TETRIX L Brackets	18	each	$2.38	$42.84
	W739090	TETRIX Gear Hub Spacer	4	each	$3.18	$12.72
	W739079	TETRIX Motor Shaft Hubs	7	each	$3.18	$22.26
	W739078	TETRIX Split Clamps	3	each	$3.18	$9.54
	W739097	Socket Head Cap Screws	60	1/2”	$0.09	$5.26
	W739098	Socket Head Cap Screws	29	5/16”	$0.10	$2.77
	W739094	TETRIX Kep Nuts	79	each	$0.02	$1.86

(continued)

Level	Part No.	Description	Quantity	U/M	Unit Cost	Total
	W758335	Cable Ties	13	each	$0.02	$0.30
	W739083	TETRIX DC Drive Motor	8	each	$23.96	$191.68
	W991444	HiTechnic DC Motor Controller	4	each	$63.96	$255.84
	W731903	TETRIX Motor Power Cable	7	each	$1.56	$10.92
	W739129	TETRIX On/ Off Switch	1	each	$5.56	$5.56
		NXT Brick Module	1	each	$300.00	$300.00

TOTAL: $1,659.57