Learning Through Racing: Rochester Institute of Technology's Imagine RIT NXP Car Cup (May 2017)

John J. Judge, Suhail Prasathong, and Daniyal Iqbal, *Students, Rochester Institute of Technology*

***Abstract*—The adrenaline of racing paired with constructive competition has proven to be an excellent method to propel engineering education in the 21st century. With robotics racing competitions happening throughout the world, the "OG Maco" team at the Rochester Institute of Technology embarked on a challenging journey that ultimately led to success. The process involved comprehensive learning, trial and error, and eventual understanding of some of the most vital concepts in the field of computer engineering.**

***Index Terms*— Intelligent Transportation Systems-Autonomous Automobiles, Intelligent Vehicles, Autonomous Vehicles, Unmanned Autonomous Vehicles**

I. Introduction

THE inception of great engineering has always been devised from a necessity and more importantly passion that drives the necessity. The combination of various avenues of engineering and science allows for comprehensive ideas, products and studies. In this paper, the authors explore the application of an array of scientific and computer engineering principles to build an autonomous robot that traverses around a track within certain restrictions with the end goal of accomplishing the most optimal time around the track. At a higher level, this application requires a good understanding of basic physical concepts such as torque, motors, drag and drift. In addition to these concepts, a deeper understanding of microcontrollers and peripheral sensors was also applied.

The true beauty of the NXP Cup was the competitive aspect of it. Various teams across multiple universities came together at the Rochester Institute of Technology's Imagine RIT event to participate in the competition. Of the many teams that competed, only 8 teams made it around the track of which only 7 teams fell under the 21 seconds' mark. This metric describes the tight nature of the competition that unfolded.

The application of the autonomous robot was genuinely diverse in the methods of possible implementations. Various

Fig. 1. NXP Cup cars lined up ready to compete at Rochester Institute of Technology's Imagine RIT Competition in the Clark Gym. Team OG Maco's autonomous racer is ready to race (fourth from the left of the screen).

teams chose different paths to accomplish the same end goal. In conversation, the authors learned that some teams decided to use linear control theory, while others chose a quadratic approach. Additionally, different teams also chose an array of various pulse width modulation duty cycle rates and threshold values to vary their speeds. While there were stark differences in programmatic and algorithmic approaches as described, there were also differences in logistical approaches. There were teams who decided not to wipe their tires off suggesting that the dust helped with smoother turns while other teams taped their tires off after every round. The best part about these differences is that while they directly affect the result, they also engage engineers in a high intensity learning process that results in the most important win of professional growth.

While building the autonomous robot, the team encountered various challenges. To acquire the optimal duty cycle, the car swung off the track repeatedly. The "averager" method helped smooth the data coming in, leading to the team's first breakthrough solution. Next, to optimize time around the track, the team implemented a control model that altered the pulse width modulation values to acquire speed reduction around the turns. Another major issue faced was inconsistent

John Joseph Judge is a student at the Rochester Institute of Technology, Rochester, NY 14623. He is now a third year student pursuing a bachelor's of science in Computer Engineering (email: jjj4008@rit.edu).

Suhail Prasathong is a student at the Rochester Institute of Technology, Rochester, NY, 14623. He is now a fourth year student pursuing a bachelor's of science in Computer Engineering (email: sxp9646@rit.edu).

Daniyal Iqbal is a student at the Rochester Institute of Technology, Rochester, NY, 14623. He is now a second year student pursuing a bachelor's of science in Computer Engineering (email: dxi9678@rit.edu).

readings from the camera because the camera angle kept changing. To resolve this, a protractor was used to mark the most optimal angle determined through trial and error. Other logistical precautions such as taping tires and checking wiring before every run were also part of the team's practice.

This paper goes on to discuss vital practices and principles in building an autonomous robot. Concepts such as pulse width modulation, proportional-integral-derivative control and general control theory are covered in depth. Additionally, truly valuable best practices and the importance precautionary measures are also highlighted to provide a comprehensive understanding of the autonomous robotics racing space. Lastly, an encouraging forecast in the field of computing and engineering is described where the race is not simply about autonomous robots but is rather about leveraging the skills acquired through these competitions to build better engineers and a better society.

II. Background

The NXP autonomous robot is an application of various key technical concepts and ideas. The discussion of these concepts includes a general strategy to achieve high speeds on the track, using control concepts like PID and interfacing with microcontroller utilities such as the flex timer.

A. Comparisons to Other Works

While the team takes pride in the rather successful strategy implemented on the autonomous robot, the comparison and learning process is vital to acquiring a full understanding of any concept. Hands-On Learning Through Racing describes how an NXP team in China used a Kalman filter to determine the angular position of the car [1]. By comparing signals from the accelerometer, gyroscope and Kalman filter output, the team could acquire optimal speeds and turning based on a full understanding of the car's location at any given time. The OG Maco team found this to be a considerably advanced approach to position acquisition.

B. Basic Strategy for Speed

Given that the primary objective was to get around the track quickest, the general strategy was straightforward. The team decided to go fast down the straights and slow down around turns. To reach high speeds down the straights, the team opted to increase the duty cycle, which resulted in more speed. When the camera detected turns, the control algorithm would kick in to scale the duty cycle and handle the turns without veering off.

C. The Proportional-Integral-Derivative Controller

The PID controller is a continuous error calculating feedback mechanism. The error calculated is between the desired point and measured variable. The correction applied is based on the formula described in equation 1.

$$u(t) = \overbrace{K_p e(t)}^{\text{Proportional}} + \overbrace{K_i \int_0^t e(\tau) d\tau}^{\text{Integral}} + \overbrace{K_d \frac{d}{dt} e(t)}^{\text{Derivative}} \quad (1)$$

The equation factors proportion, which represents present values of the error, the integral, which factors past values of the error, and the derivative, which models likely future errors based on the current change rate [2]. The application of the PID controller on the autonomous robot allowed for crisper readings from the camera which led to a more optimal threshold value selection process. Thus, the benefits of applying PID became evident in the success of the robot.

D. Flex Timer Module Application

The Flex Timer Module (FTM) is a timer designed for motor control and power management. The timer generates PWM signals and its peripheral modules such as polarity, fault control and triggering [3]. In this application, FTM modules were used to generate PWM outputs through interrupts. Using the FTM and its clock, the team selected appropriate duty cycles to implement and made optimal timing decisions.

III. Proposed Method

A. Camera Output

To get the image of the track, a wide angle camera was used. This was done to ensure that both lines of the track can be seen even when the car is at the edge. The camera captured 128 pixels of data and these pixels were converted to discrete values and stored into an array which could then be manipulated to extract valuable data. The pixel values varied significantly from 500 to 150,000 depending on the lighting conditions. The pixels with a greater value denoted a white pixels while pixels with values closer to 0 denoted a black pixel. The camera was placed at the center of the car and was pointed straight at the track and parallel to the lines. This was crucial to making sure that the camera readings are good. To visualize the camera reading, an oscilloscope was used to plot the wave from the camera. Figure 2 shows the data acquired from the camera.

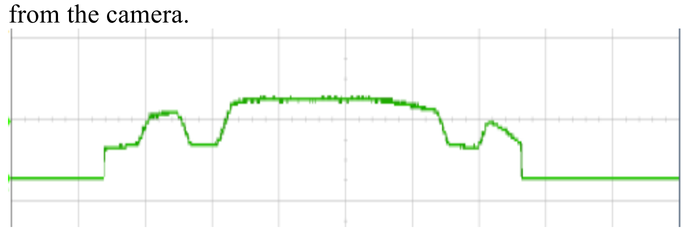


Fig 2. Image of an oscilloscope capture showing the image gotten from the wide angle camera.

B. Processing Camera Output

In Figure 2, the straight line in the middle is made up of high pixel values which represent the white area of the track. The 2 pits on either end of this line are the low values which represent the black lines on the sides of the track. Additionally, the camera had an integration period which was the amount of time that one reading took. During this time, the capacitors in the camera are being charged up and the longer this time was, the more precise the readings were. In Figure 2, the integration time was set to 10 ms, which was why the camera reading was so precise. However, the downfall of having a high integration time was that the car made fewer decisions in one second. The goal was to make the integration as small as possible while still being able to get a valuable reading from the camera. This was to drive up the frequency of decisions, which would help the car stay on track and correct its position as fast as possible. However, as the integration time was decreased, noise began to have a greater effect on the data. To reduce the noise, a five-point averager was used which eliminated some noise on the signal. The code for the algorithm is presented in Figure 3. Additionally, Figure 4 shows the plot of the original camera data and Figure 5 shows the plot of the five point averaged camera data.

```
For i from 2 to 125
        smooth[i] = 0.20 * (noisyData[i - 2] + noisyData[i - 1] +
                noisyData[i] + noisyData[i - 1] + noisyData[i + 2]
```

Fig. 3. Pseudo code for 5 point averaging algorithm.

C. Determining Position

Additionally, from the image in Figure 2, it can be seen that there is a jump in the pixel value when the pixel read goes from black to white. Likewise, there is a drop in pixel value when the pixel read goes from white to black. To detect these cases, a derivative function was used which was efficiently done by subtracting two adjacent points. Now, when the track goes from black to white, the derivative will be large and this will allow for the detection of the left side of the track. Likewise, when the track goes from white to black, the derivative will be a large negative value and this will allow for the detection of the right side of the track. Once the noise was reduced, the derivative function was applied in order to detect the jump from low to high pixel values and high to low. The next step was to detect this large positive (peaks) and negative (troughs) value.

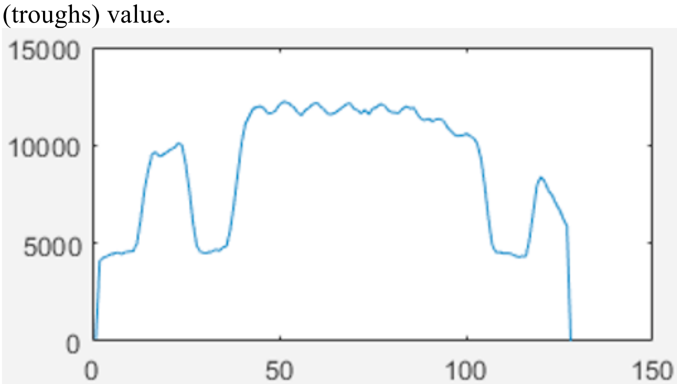


Fig. 4, Plot of the original data gotten from the camera and graphed by MATLAB

The common strategy that was taken to detect the peaks and troughs was to manually set a threshold based off of the lighting conditions of the track. This solution required tedious calibration of the threshold when lighting conditions are changed. In order to overcome this, an algorithm was made that does not rely on a preset threshold. First, the derivative array would be parsed to determine the greatest and the smallest value in the array. However, the index of the peak and trough were not used to determine to determine the left or right edge of the track, respectively. This was because the max value may not be exactly at the edge of where white and black meet. To get a more precise number on both sides, the first point from the left that is greater than 80% of the peak value is the left track point. Likewise, the first point that is less than 80% of the trough is the right track point. Other groups checked if the derivative was greater than a certain threshold to determine the track corners. However, if there was noise, the value may not meet the the threshold or meet the threshold when there is not a change from black to white. Additionally, if there was a shadow cast on the track or if the lighting gets brighter, then the processing of the data would result in an incorrect decision. A single incorrect decision can sometimes cause the car to go off of the track which is why a constant threshold was not used. The pseudo-code for the algorithm is shown in Figure 4.

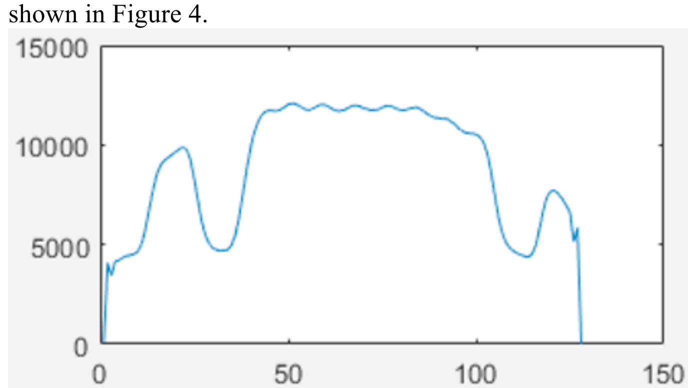


Fig. 5. Plot of the five point averaged data of the camera data in MATLAB

D. Centering the Car

From the left and right indices, the middle position was determined by taking the average of these two numbers. Since 128 points were measured, the middle position should be 64 if the car was in the middle of the track. So, if the calculated middle position was greater than 64, then the car is left of center and should turn right. Likewise, if the position is less than 64, the car is right and should turn left. The range of the car was determined to be 5.3 to 8.9. The scaling was found experimentally by putting the car at the far right corner and far left corner of the track to determine the middle position detected by the car and then mapping those values to 5.3 and 8.9. So when the car is at the far left corner, the scaling would set the direction to 8.9.

```
maxVal = 0;
For i from 2 to 125
        if (derivative[i] > maxVal)
                maxVal = derivative[i]
        if (derivative[i] < min Val)
                minVal = derivative[i]
For i from 5 to 123
        if (derivative[i] > 0.8 * maxVal)
                leftIndex = i
For i from 5 to 123
        if (derivative[i] < 0.8 * minVal)
                rightIndex = i
```

Fig. 6. Finding left and right points of the track

E. Variable Speed

Additionally, there was also variable speed to make the car faster when it is going straight and slower when it is going through turns. When the position was within 3 pixels of 64, the car was set to go straight at +20% faster than the minimum speed (constant variable set). When it was within 8 pixels of 64, it was set to go +10% faster than the minimum speed. When the position was outside of that range, the car was set to go at the minimum speed on turns. This allowed for achieving faster speeds while still being able to make the turns. When it was slower Additionally, to make the turns at faster speeds, a differential drive was also applied at sharp turns. When the wheel direction was at its maximum, then the wheels speed would be changed so the motor can help make the sharp turn. For example, if a sharp left turn needed to be made, the right motor would stay at the same speed while the left motor would be slowed down to 10% of maximum speed. This allowed for making sharp turns at higher speeds. Multiple other algorithms were tried such as quadratic scaling for turning, but these algorithms made the car less smooth and were not able to achieve higher speeds.

G. The Hardware of the Car

In addition to the KL64F microcontroller the standard NXP cup car kit was used. This kit included the Freescale Linescan Camera with accompanying wide-angle lens, the car chassis with attached motors, and FDRM-TFC motor-shield. Standard jumper wires were used to connect the ports on each one of these pieces. The motor shield came connected to the power enables, grounds, and control signals for the DC and servomotors. Essentially this motor-shield acted as an intermediate relay between the control signals (from the microcontroller and camera) and the motors. It also contained the H-Bridge used to control the directions and speeds of the DC motors.

The wiring connections shown in Table I details that the the DC motors used four control signals. The four signals were used to each of the DC motors clockwise and counterclockwise. This was done for turning purposes. When the car was trying to make a tight turn the inside wheel could be driven counterclockwise and the outside wheel clockwise. In order to achieve this the Flextimer channels needed to be added for the counterclockwise PWM signals. FTM0 was used for all four control signals with channels three and two to drive the left motor. Channels 1 and 0 were used to control the right motor. These channels were chosen because they had accessible pins on the K64F GPIO.

Table I. Wiring connections present on the car between the microcontroller, camera, and the motor-shield.

Component	*K64 Pin*	*Signal*	*Motor control board Connector-Pin*	*Description*
Power and Grounds	P5-9V_VIN-16	V+ Battery	J9-16	Power
	GND-14	GND	J9-14	Ground
	GND-12	GND	J9-12	Ground
	P3V3-8	3.3V	J9-8	Power
	P3V3-4	3.3V	J9-4	Power
DC Motor	PTC2	A-IN1	J1-5	Left motor control signal forward
	PTC3	A-IN2	J1-7	Left motor control signal backwards
	3.3V	Enable	J10-3	Motor enable
	PTC5	B-IN1	J10-12	Right motor control signal forward
	PTC1	B-IN2	J10-10	Right motor control signal backwards
Servo Motor	PTD0	Servo-pwm	J10-2	PWM signal for the servo
Camera	PTB23	SI	J2-19	Camera SI pulse
	PTB9	CLK	J2-20	Camera clock
	ADC0_DP1	AO	J2-4	Camera analog output

In addition to the electrical hardware the motor-shield, microcontroller, and camera needed to be attached to the chassis. This was accomplished using a few 3-D printed ABS plastic pieces. A rear mount that held both the microcontroller and motor-shield was attached to rear of the chassis. In order to attach the camera a mount, hinge, and base mount needed to be printed. The camera was then attached to the mount and that mount was attached to the hinge with a pin. This allowed the camera to be adjusted in testing to determine the best camera angle depending on the conditions of the track. The camera and hinge were then suspended above chassis using a wooden dowel. This dowel was attached to the chassis using the base mount and two screws and plastic nuts.

IV. RESULTS

A. Fine Tuning PID and Differential Drive

There were multiple ways the PID was fine-tuned to achieve the fastest timing results. The initial linear scaling for direction was making the turns too late, so the scaling was increased so it would achieve the maximum direction at 75 instead of 79 for right turns and 55 instead of 51 for left turns.

This was done by experimenting with the scaling values to determine which scaling can make the turns at the fastest speed. Additionally, the differential drive was changed significantly especially when the motor duty cycle was increased to 70%. Instead of having differential drive in only sharp turns, a small amount of differential drive was used for moderate turns as well. At moderate turns (within 8 pixels), one of the wheels would be lowered to 40% of maximum speed. This turned out to work very well and was able to make the turns at faster speeds.

B. Results of the Race and Future Improvements

On the day of the race significant fine tuning of the line finding and turning algorithms was needed. This was due in large part of the relatively poor lighting conditions on the day of the race compared to the brighter conditions of the testing tracks. When first tested on that day, the car could not even find the lines of the track. The integration time needed to be increased because the capacitors of the camera needed more time to charge when capturing the lines of the track. In addition to this, the turning algorithm needed to be adjusted in order to be better suited for the turns of the final track.

Autonomous Model Car Competition 18.394

TeamName	Institution	Time #1	Time #2	Time #3	Final Time
The Bald Eagle Machine	RIT	17.394*			18.394
Savage Valhalla	RIT	17.683*			18.683
The Huns	RIT	17.923*			18.923
Team Ben	RIT	19.362*			20.362
Antlion	RIT	19.550*			20.550
Og Maco	RIT	X	19.764*		20.764
Team Amethyst	RHIT	26.755*			27.755
Attar-Scheler-Orifov	RIT	X	X	28.676*	29.676
In	RIT	X	X	X	
Fine Line	RIT	X	X	X	
Team Tom	RIT	X	X	X	
Langlois-Perry	RIT	X	X	X	
To Be Determined	RIT	X	X	X	
Jolly Rancher	RIT	X	X	X	
Team Silver	RHIT	X	X	X	
Kid Flash	RIT	X	X	X	
The Pi Kapps	RIT	X	X	X	
Magalhaes-Tosaya	RIT	X	X	X	
Can We Think about It	RIT	X	X	X	

Fig. 7. The results of the competition. Team Og Maco finished in 6th place.

As seen in Figure 7, less than half of the teams completed at least one lap around the track. This car, with team name Og Maco, ended the competition being placed sixth. The lap time of 19.764 seconds (with the additional 1 second penalty for not breaking upon completion of the lap) was accomplished on the second attempt. The first attempt resulted in three of the four wheels leaving the track on a sharp turn resulting in a non-completed lap. On the next attempt the buttons on the K64F were used to decrease the speed at which the car would be making the turns as well as driving down the straightaways. The car was then able to complete the lap while still maintaining all wheels on the track at any given time. This is evident of successful line following and turning algorithms.

Although merely completing the lap was certainly an accomplishment there was certainly room for improvement. When the speed of the car was altered with err to caution. It remains to be seen whether the speed was decreased too much and a faster speed would have still kept the car on the track but shaved some time off the lap. In addition to this, the wheels of the car were not cleaned during the competition itself. During testing the wheels were cleared with a solution to remove the buildup of dust on the treads. Tape was then wrapped over the wheels and then removed, leaving a sticky residue on the wheels allowing the car to better grip the track on the turns. This was simply an oversight during the competition but surely it could have improved the car. Perhaps the car would have kept at least two wheels on the track during the first, and more aggressive, attempt.

The final, and perhaps most beneficial, improvement that could be made is the ability to brake following the completion of the timed lap. Although a algorithm to brake was initially created testing proved it was not reliable. With the speed the car was driving at, and integration time of the capacitors, the camera would not always detect the black bands signifying the start and finish line. On other occasions the algorithm also proved to be troublesome when noise on the cameras AC signal was misidentified as the start and finish line, resulting in the car braking prematurely. Had this algorithm been working the one-second penalty could have been avoided thus moving the car from 6th place to 4th place, *ceteris paribus*.

V. Conclusion

The successful completion of this project and competition added to team Og Maco's engineering ability. This project contributed to valuable engineering skills such as problem solving and the implementation of embedded systems. Perhaps the most important of which was working in a team setting and achieving a common goal. This project served as an opportunity to see how sensors can be interfaced with electronics and software as well as a practical application of general engineering concepts.

References

[1] Qing Zhou, Yanpin Ren, Yongheng Jiang, Changshui Zhang, "Hands-On Learning Through Racing: Signal processing and engineering education through the China National Collegiate Intelligent Model Car Competition," *IEEE Signal Processing Magazine*, vol. 34, pp. 31-39, Jan. 2017.

[2] Bennett, Stuart, "A History of Control Engineering," *IET.*, pp. 48, 1955, *ISBN:* 978-0-86341-299-8.

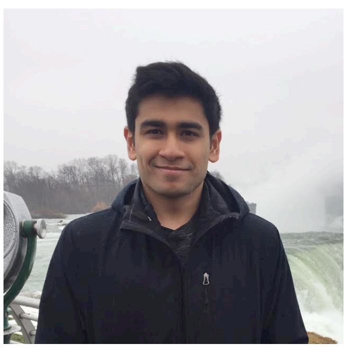
[3] Pavel Rech. (2011, November). Flex Timer for Field Orientation Control. *Freescale Semiconductor Group.* [Online]. *AN4410.* Available: http://www.nxp.com/assets/documents/data/en/application-notes/AN4410.pdf

John J. Judge was born in Pearl River, NY, in 1996. He graduated from Pearl River High School with honors in June of 2014. He is currently in his third year of study at the Kate Gleason College of Engineering at Rochester Institute of Technology in Rochester, NY. He plans on graduating in May of 2019 with a Bachelors of Science in Computer Engineering. In this time he is also pursuing a minor in Economics. He is working for Architecture Technology Corporation in Ithaca, NY from May 2017 to January of 2018. He will be working on both low-level I/O applications as well as software design. His qualifications include hardware design as well as low-level

design protocol. In his personal time he devotes himself to projects that include digital system design as well as instruction set architectures that can run on the projects. These projects include an 8-bit MIPS personal computer and microcontroller programming.

Suhail Prasathong was born in Bangkok, Thailand in 1995. He is pursuing a B.S. degree in computer engineering from the Rochester Institute of Technology, Rochester, from 2013-2018. Throughout his experience at RIT, he has worked towards achieving a comprehensive understanding of algorithms, data structures, lower and higher level programming, computer architectures, digital signals and so forth. Prasathong's main field of interest in Machine Learning where he intends to pursue a senior design capstone project that uses machine learning models to recognize objects and translate them to different languages. His expertise in higher and lower level programming provides him a good platform to pursue projects in the machine learning space.

In his time at RIT, Mr. Prasathong has acquired 3 co-op opportunities at Kodak Alaris, PayPal Inc. and Deutsche Bank's machine learning team respectively. Through the co-op program at Rochester Institute of Technology, Mr. Pasathong has had the opportunity to apply the skills acquired through academics. Taking full advantage of the program, Mr. Prasathong was able to leverage his academic learning complemented by industry experience to his advantage.



Daniyal Iqbal was born in Karachi, Pakistan in 1997. He is pursuing a bachelor's in Computer Engineering and Computer Science at Rochester Institute of Technology. He is currently involved in the research of using wireless communications in data centers to improve power efficiency. He is passionate about machine learning and computer vision, and he hopes to devote his time to exploring these fields further.. He hopes to one day combine his interests in entrepreneurship and engineering to launch his own tech start up.