ASEE 2014 Zone I Conference, April 3-5, 2014, University of Bridgeport, Bridgeport, CT, USA.

Design of an Autonomous, Line Following Pace Car for Athletic Training

David Cain, Brian Layng, Kyle McNulty, Ryan O'Connor
Department of Engineering & Physics
Elizabethtown College
Elizabethtown, Pennsylvania, USA
McNultyK@etown.edu

Dr. Tomas Estrada
Department of Engineering & Physics
Elizabethtown College
Elizabethtown, Pennsylvania, USA
EstradaT@etown.edu

***Abstract* -- The total scope of the project aims to effectively and efficiently assist a runner with a visual representation of pace on an outdoor track. This pace car is intended to be a training aid for any runner. This project uses a repurposed RC car being controlled by an Arduino Uno microcontroller. It uses infrared phototransistor sensors as input to understand where the lane line on a track is. This project is unique in that it bridges the gap between athletics and academics. The members' personal interest in athletics helps to motivate the progress of the project. The eclectic team consists of engineers from different disciplines including mechanical, computer, and industrial.**

Keywords- robotics; multidisciplinary team; path following

I. Introduction

When individuals work out, often they will seek out training partners for the reason of having motivation. Many runners claim that workouts become easier when they have someone to run next to. Unfortunately, people cannot always find someone to run with and it is even harder to find a running partner that matches their exact skill level. For that reason, our project focuses on designing a small, 4-wheeled training robot that simulates the motivation of running with someone by giving the runner a visual representation of pace.

There is currently no pace car in the market that works specifically on an indoor and outdoor track. The closest mechanism to our pace car is something called the Joggobot, The Joggobot does not act as a pace car to aid in improving times but instead acts as a jogging buddy so the user is not running alone. Key differences between the two include; the pace car's method of mobility is a 4-wheel while a quad-copter method of propulsion used by the Joggobot. Another difference is the method of steering, instead of line following like our pace car, the Joggobot uses a camera and sensor oriented to follow a strip on a t-shirt worn by the runner. While the products are similar, they do not accomplish the same task of pacing [1].

The motivation for this project was sprouted by a group member's involvement with the Elizabethtown College Track and Field team. Having had first-hand experience with running a workout alone and with a partner, he has knowledge of how influential having someone to run with can be. As a group, we saw this project as a way to combine academia and athletics while challenging our knowledge of engineering skills.

II. Project Overview

In the spring of 2013, we enrolled in a Junior Project class offered by the College's Engineer Department. The class was mainly focused on research, development, and planning. Junior Project allowed us to refine our basic thoughts for possible projects. We had one semester to cultivate our initial ideas and determine the feasibility of our project. In the class we assessed the marketability of our project and started our initial design specifications of what we wanted to our pace car to achieve prior to Senior Project which is a class offered in the Fall of 2013 and continues on to the Spring of 2014.

Starting Junior Project we developed a problem statement regarding our intended project for senior year:

> *Design and construct a user friendly pacing vehicle to aid in the training of a typical athlete. The pacing vehicle will use line following technology in order to maintain a certain speed while navigating a set path.*

After defining the problem, we thought the best way to discover if the project would be a practical senior project would be to create a prototype analog line reading robot.

Completion of the Junior Project allowed us to enhance our goals of what we wanted our pace car to do for senior project and what was necessary to achieve those goals.

The main idea of our project is to use a prefabricated remote control car in junction with a programmable logic controller (PLC). The PLC will take input from infrared sensors in order to dictate the location of the track's lane line. Our initial idea was to keep our speed controller and steering controller as separate entities, however, we found the Arduino Uno was capable of executing both task in a single unit.

Our group is comprised of undergrads with a variety of majors and skills

Brian Layng: Mechanical Engineering

Ryan O'Connor: Computer Engineering

David Cain: Computer Engineering

Kyle McNulty: Industrial Engineering

III. DESIGN PROCESS: EARLY STAGES

During Junior Project, our initial design idea of the project was to manufacture a small car that will follow a line at high speeds. It was to be comprised of basic analog components. Simply said, this car would take analog input from infrared sensors, compare these values in a micro controller, send the results to a motor driver, and the motor driver will determine the direction that the axel will turn. We used a low cost remote control car as a platform for our research.

Some options for the line sensors were to create our own using photo resistor cells or to use a premade infrared reflective sensor. We ran tests with the photo resistors and decided that they would work, for this portion we decided to purchase cost effective infrared sensors that met all of the operational requirements. They were efficient in determining a black line against a bright surface.

Our next focus was on the microcontroller. After much research on the microcontrollers in the market, we agreed on using an Arduino Uno Rev 3 because it had the ability to do exactly what we wanted it to do, while being extremely user friendly and was an open source platform.

Due to the fact, the low cost remote control car's steering mechanism consisted of just a DC motor; we used a motor driver that paired well with the Arduino Uno to turn the Arduino signal into a physical action.

While the initial prototype completed its intended task, we understood our final model would require a higher velocity and a more advanced steering control. This brought us to the beginning of our senior year.

IV. IN-DEPTH DESIGN CONSIDERATIONS

Design considerations focused primarily on mechanical design and control design to confirm desired speed and usability were met.

Mechanical Design

- E10 Drift Remote Control Car
  - Carefully considering four high powered electric remote control cars, we developed a weighted Pugh diagram to determine the proper car to satisfy our design specifications. We chose to repurpose a remote controlled car in this project because it already had all the physical components that are necessary. We chose the E10 Drift because it was the right size and its driving and steering mechanisms are easy to control with the Arduino. We also upgraded the motor from the stock model of 27T motor to a more powerful 15T motor to reach desired speeds [2].

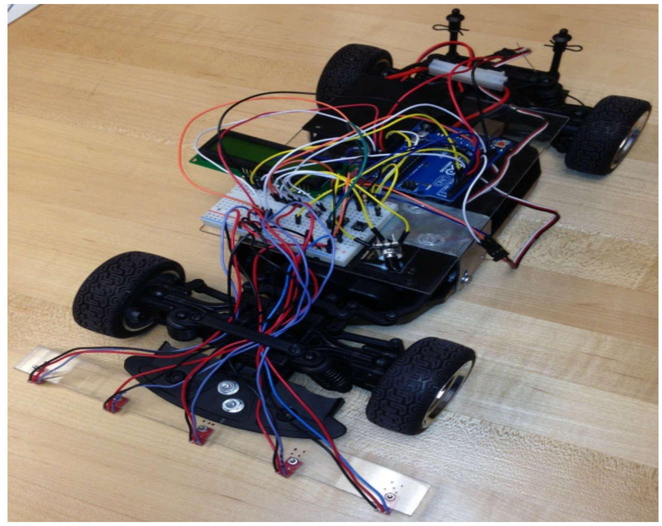

Fig. 1. Current aesthetics of the Pace Car project

- Plexiglass electronics staging platform
  - Attached above the chassis is a plexiglass platform. We needed a platform to aid in staging electrical components including the Arduino, the LCD display and breadboard. We chose plexiglass because is non-conductive and transparent so we could still see the components underneath. The plexiglass platform was initially designed as an L-shape as seen in fig. 2, but was later changed to a square platform for more space.

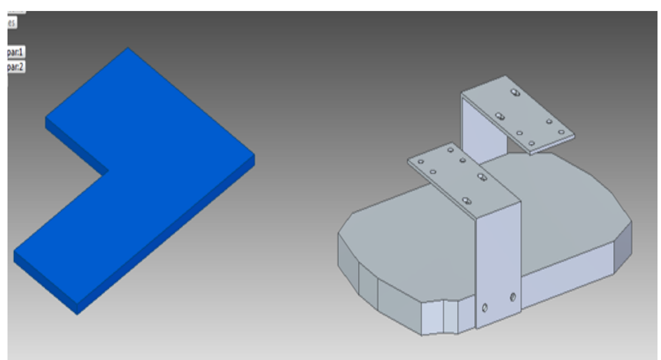

Figure 2: CAD drawings of chassis and platform

- Infrared Reflective Phototransistors
  - Mounted to the bumper of the E10, we attached a plexiglass sensor bar featuring infrared reflective phototransistors. The phototransistors are used as inputs for determining where the white line of the track is relative to the car. They work by emitting an infrared light off a surface and depending on the color and darkness of the surface, it will return a voltage. This voltage can be read by our microcontroller.

- Arduino Uno Rev 3 Microcontroller (See Description in Section III)

Control Design

A basic control block diagram can be seen in fig. 3. The sensor bar sends an analogue value dependent on the brightness of surface the sensors are viewing. These values are processed by the Arduino to determine the location of the line relative to the car. The microcontroller calculates error based on which sensor is reading the line. This error is fed into a PID (Proportional Integral Derivative) control loop where it determines what angle the wheels should turn towards. We received help with coding from tutorials found on the Arduino website [4].

The throttle control is mainly taken care of by the preexisting electronic speed controller (ESC). However it does still need to be sent information given by a user through the Arduino. To make the Arduino talk to the ESC, extensive testing was needed. Essentially, the Arduino treats the ESC like a servo motor and sends it a pulse width modulation value between 0 and 180. We found that by initializing the ESC with the value 77, we could effectively control the speed of the motor by sending it values between 77 (0%) and 180 (100%). Initializing it with a value lower than this would result in sporadic throttle behavior. Initializing it with a value larger than 77 would cause the lowest value to become the new 0% [7].

V. Performance and Testing

Due to inclement weather we have set-up a small indoor track used for testing our code. The test track is black while the outside track is red. We believe that the indoor track is a sufficient test analogue. We rigorously testing but are currently facing problems with overshoot and undershoot. This will be a problematic challenge to overcome due to varying speeds. We plan on intensively testing and tuning the code. Initial tests verify line following on a red track is a plausible outcome.

In short, the steering control works by taking input from the phototransistors. These inputs are received by the microcontroller in the form of values between 0 and 1024. The lower the value received, the lighter the surface below it is. The logic determines what sensor is reading the lowest and lightest value. From there, the microcontroller determines the error, which is how many sensors away from center is reading the lowest value. This error is fed into a PID control algorithm. This algorithm attempts to smooth out the turning by reducing over and under damping. The outcome of the PID algorithm is sent to the steering servo as a turning angle [5][6].

VI. Reflections and Discussion

With many hours of planning, decision making and testing a few key take always we have learned from the project thus far is to set goals for each week. Having pre-established goals for each week allows us to focus our efforts in the correct areas, if we all know what we want to achieve a comradery is established to accomplish our common goal. Another thing we learned from working together is that an agenda is important to stay on task, if we all know our assignments prior to each meeting we waste little time getting to work. Through this project we expanded our knowledge on circuitry and computer programming.

The Pace Car Project provided us with a practical platform to apply and expand upon our knowledge of electronics, mechanics, and robotics. The project allowed to study how electrical components work in tandem.

Working on a multidisciplinary team allowed us to have many different perspectives and ideas of how to proceed forward in the stages of designing, building and testing the car. The diversity of team was beneficial due to the different academic backgrounds in the area of knowledge. Being a part

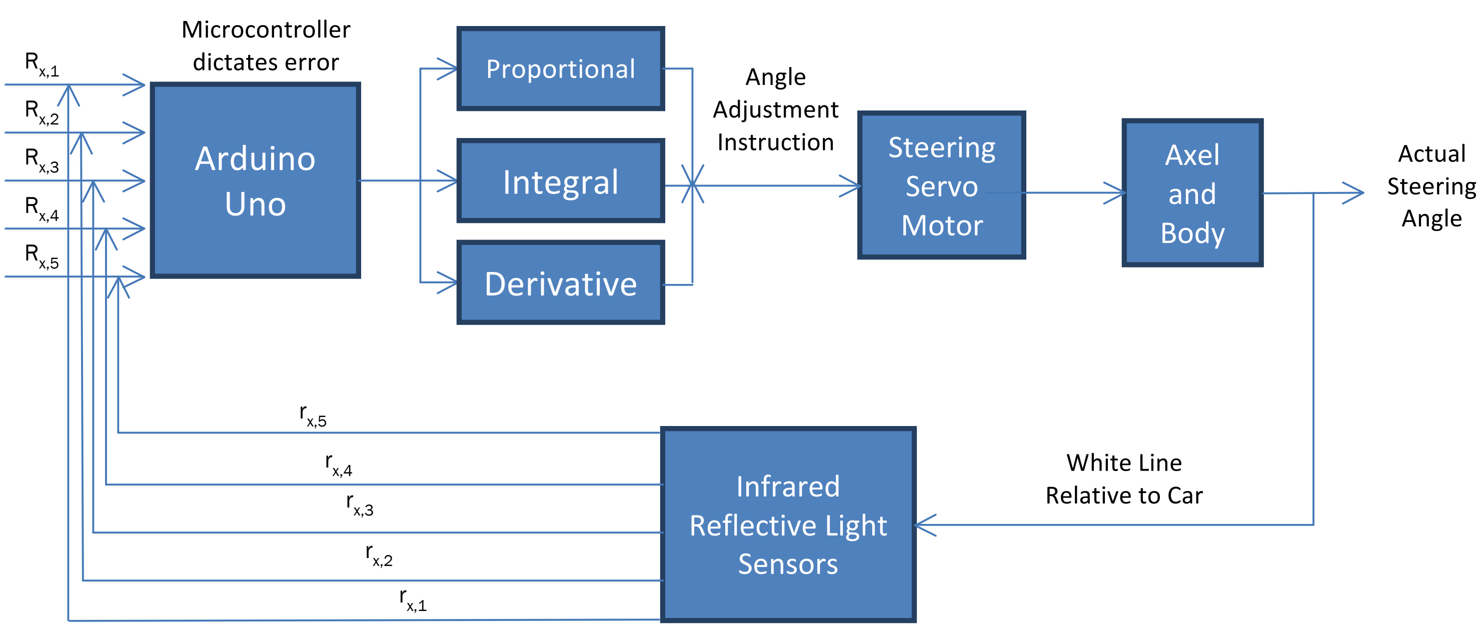

Fig. 3. Block Diagram of Control System [3]

of multidisciplinary team meant exceptional communication and accountability were key for our project. Another key aspect that made for an ease of working together was not only our passion for athleticism but also the ability to take pride in our respected areas of study.

VII. Conclusions and Future Work

While our project is not yet working as precise as we would like it to, there are still notable hurdles we have overcome. We have all learned a great deal through research and hands-on application including programming a microcontroller, applications of phototransistor sensors, control systems, and gained practical experience that we believe will be of use after graduating.

We will continue to improve on our car for the remainder of the semester. Our passion for this project has developed to a point of genuine enjoyment. We believe that we are leaving behind an exceptional learning platform for future students to expand and improve on while learning themselves. Possible work for future students to pick up includes refining code to reduce any errors and enhance response to the cars position relative to the line, improving battery life, creating software to be coupled with the pace car for user friendly programming, and creating a wireless communicator between the user and the pace car. Weather proofing is also something that needs to be looked into; the car as it stands has difficulty operating in cold conditions, as well as, rain/wet conditions. An emergency stop function to stop the car in case there is an error or incase the user cannot keep up with the car is another improvement that could be added to the project.

REFERENCES

[1] E. Gräther & F. Mueller, “Joggobot,” 2012. <http://exertiongameslab.org/projects/joggobot >

[2] Hobby Warehouse, “Difference Between Brushed and Brushless RC: Choosing Your Motor Type”. 2012. <http://www.hobbywarehouse.com.au/articles/difference-between-brushed-brushless-motors.html>

[3] G. Krolin-Taylor, “Micro- and Nanosystems,” in Control Systems, vol. 33, 2013, p. 51.

[4] M. Banzi D. Cuartielles, T. Igoe, G. Martino, D. Mellis, “Arduino Support and Tutorials,” <http://arduino.cc/>

[5] R.C. Dorf & R.H. Bishop, “Modern Control System,”Prentice Hall,ed.11, 2007.

[6] SWR, “PID Control: A Brief Introduction and Guide, Using Arduino,” 2011. <http://www.maelabs.ucsd.edu/mae156alib/control/PID-Control-Ardunio.pdf>

[7] Arduino forum. Date accessed: 5 January 2014.<http://forum.arduino.cc/index.php?topic=50306.msg369845#msg369845>