ABSTRACT

This project aimed to build a remote controlled car that is driven using a steering wheel and pedals from a console station, much like a real-world version of a driving simulator. Driving feedback is provided by a wireless camera and visual indicators, and through the use of sensors a controls algorithm is implemented for improving car performance through torque vectoring in turns. This platform was developed for an undergraduate computer engineering Controls class as well as demonstrations at RIT’s yearly Imagine RIT innovation and creativity festival.

INTRODUCTION AND BACKGROUND

The recent increase in availability of affordable electronic components has allowed creativity and engineering to merge through do-it-yourself (DIY) projects that stimulate imaginations around the world. The objective of this student initiated project is to utilize the design process facilitated by Multidisciplinary Senior Design at RIT to produce a captivating and educational project that demonstrates the sort of innovation that engineers can produce.

The genesis of this project idea came from a simple online video of a similar setup [1]. The setup in the video was not documented in any other capacity, and simply demonstrated the concept of an RC car being driven on a course with a camera providing feedback to a console. The console is an arcade-style booth setup with pedals and a steering wheel. The controls and video feed are sent wirelessly through an unknown system, but it appears that the system uses many custom made DIY components.

As the concept is interesting, fun, feasible, and open to much modification, this serves as a perfect project for MSD. Notable improvements that were targeted for the setup from watching the video included: 1.) making the course more mobile and practical, 2.) reducing the opportunities for the car to flip over and require manual resetting, 3.) making the console more mobile and practical, 4.) using more readily available components to make the project more reproducible, 5.) increasing the level of documentation significantly.

The project was approved for MSD after merging with the RIT Freescale Cup program done through the computer engineering department. The chassis used for the competition proved to be a very good platform for development of custom components, and the inclusion of Freescale sponsorship through electronics for the project made the project more feasible as a student initiated concept. This sponsorship involved addition documentation that is targeted at assisting undergraduate students in the Freescale competition to assemble and program the car.

Though Tim Southerton, the project originator, still serves as the primary project customer, an additional faculty customer had to be established for the MSD course deliverables. This is why Dr. Cockburn is listed as the customer for the project, as he elected to use the project for his undergraduate computer engineering Controls class. This additional component involved the addition of sensors and the implementation of a controls algorithm to control the rear-wheel speeds actively and improve cornering performance. With this comes the need for documentation necessary to use the car as a class project.
This combination of events led to the current project, which was demonstrated at Imagine RIT and will be used in the Controls class mentioned. It is our hope that this project idea will form the basis of future project ideas in which more work can be done to improve the application of the controls algorithm to the car and other components to make more impressive demonstrations.

**DESIGN PROCESS**

The complete setup for the project is required to move the car, protect the car components, provide an immersive driving experience, demonstrate control systems, and interest people in engineering. For moving the car, we established reasonable performance parameters (3 m/s max speed, 0.5 m turning radius, etc.) to be expected based on testing with other RC cars that team members individually own. Distances for wireless communications (200 ft) were also established based on available documentation. For the driving experience, physical parameters for the console setup were established by measuring real car components (5 cm throttle travel, 3:1 steering ratio), and camera performance parameters were established off of known camera options with budgetary limitations (VGA resolution, 15 fps). For the control system and other aspects of the setup, Boolean functionalities were targeted to allow for additional usage of the car (measuring wheel speed, equations of motion, forward and reverse, etc.) Major evaluating parameters included the $500 budget, 1 hour run time on a charge, and various participant feedback parameters evaluating the quality of the product at Imagine RIT.

Many of the design choices for the major components were fixed by the budget and sponsorship. The car chassis chosen is the Freescale Cup chassis, the microcontrollers used are Freescale Freedom boards (KL25Z), and the batteries used are two 3000 mAh NiMH batteries, as these all have the functionality necessary and were provided at no cost. To accomplish the controls application custom encoders were designed with optical gates to provide feedback. The car also had limited ground clearance so larger tires were added to the design, and to protect the components two bumpers and a series of Lexan plates were designed to do so at minimal cost with considerable room for modification in the future. For wireless communication a donated set of XBee 802.15.4 XB24 2mW trace antenna modules were used for the data transmission, which operate in the 2.4 GHz range and cover the target range with minimal interference. An affordable first person view (FPV) camera kit was selected to reduce lag seen in other video systems. For the console, a steering wheel and pedals were selected as a Logitech MOMO was donated to the project, and a custom car seat and driving station with an LCD monitor was selected to provide a mobile, immersive experience due to a car seat donation from FMS and a TV donation from a group member.

**RC CAR PLATFORM**

Due to the amount of components necessary to accomplish the objective that had to be added to the car chassis, a number of modifications had to be made. The original car suspension system, which was poorly designed, was removed from the car and the orientation of the steering servo was reversed to accommodate additional components. Detailed measurements were used to generate a CAD model of the Lexan plate structure which was used to house and protect the components, attached to the chassis using standoffs. A bumper was also designed to reduce front impact collisions which could damage the electronic systems. This provided a strong base for the development of the electronic components while allowing for easy modification. These components can be seen in Fig. 1.

Due to the ground clearance issue with the stock chassis, new wheels were specified and purchased, which necessitated the design of new transaxles and rear axle for the chassis to accommodate turning. This design tied in with the encoder design, which included 3D printed parts for direct connection to the existing chassis structure. Encoder clearances were taken into account to that ground contact was not made with the new wheels, and reflective optical switches were able to be mounted directly to the back of the chassis to collect speed data. The final components can be seen in Fig. 2.
The camera system chosen for the car was a 5.8GHz FPV kit with a 420 line CCD camera, 600 mW transmitter, a standard receiver, two 3S 35C 11.1V LiPO batteries, a voltage indicator to prevent battery damage, and a set of circularly polarized antennas. This system was chosen as it has the minimum interference with Wi-Fi commonly available while still avoiding the lag issues associated with Wi-Fi cameras. A heat sink was added to the transmitter as it is advised for any conditions in which the transmitter receives minimal airflow (in comparison to being mounted on a radio controlled plane). During our testing the heat sink reached a steady state temperature which is warm to the touch but not hot, which is a worst case airflow scenario and is desirable. The camera is mounted to a servo that turns with the steering to help driver visibility, and this setup was incorporated into the Lexan plate designs to be protected while not reducing viewing area. Transmitter mounting was similarly included to place the antenna as high as possible to get the best signal transmission. Battery placement was a minimal priority.

**DRIVING STATION WITH CONTROLLER**

Though not an area of major importance to the functionality of the project, the console design was taken on as a chance to utilize donated components to produce a mobile platform with a professional appearance for this project. A rear car seat from a 2002 Chevy Venture was donated from RIT FMS, a 19” LCD TV from Brian Grosso, a computer tower, mouse, keyboard, and wireless adapter from the RIT Mechanical Engineering Department, an office chair swivel mechanism from Tim Southerton, casters and table surfaces from the RIT Machine Shop, a desk that was being recycled, and angle iron, nuts, bolts, and washers from the MSD area. These various pieces were combined with minimal supplies from Home Depot to produce the entire console for nearly no cost to the project. This platform is the basis for all testing and programming, and was used for the Imagine RIT demonstrations. Design points included ride height and caster size, adjustability of the seat and table for different height users, ease of getting into and out of the seat, adequate mounting area for all electronic components, adequate load bearing structure, and a convenient handle for transportation.
CONSOLE ELECTRICAL INFORMATION

The main component of the console is a donated Logitech MOMO game controller with a steering wheel and pedals, as seen with our CAD model in Fig. 3. Since the steering wheel features mechanical stops for the motion that cannot be removed due to wiring limitations, the steering ratio was limited to 3.4:1 while still adjustable to higher ratios with less wheel turning angle. The device was modified to allow for easy access to the electronics by way of a hinge mechanism, and a KL25Z board was installed inside the enclosure for programming. More information about this microcontroller can be found at the MBED handbook page [2].

The schematic in Fig. 7 represents the custom connections made in the MOMO console. These connections are made to enable the sensors in the MOMO console to communicate safely with the FRDM Board. The MOMO console signals have been conditioned with filtering and a level shifter. The only signal that uses the level shifter is the centering signal for the steering wheel. The other signals are filtered for noise and to reduce the current and voltage that the KL25Z observes. The steering wheel motor contains two encoders that are out of phase from each other, creating a quadrature encoder. The quadrature encoder works by converting the two encoder signals into a binary counter. This gives the user the advantage of knowing both the speed and the direction of the turning. This is known by finding the change in the binary counter created by the quadrature encoder. The centering signal is used to find the physical center, or origin, of the steering wheel movement. The speed and direction of speed is known from the quadrature encoders, but the location must be established each time the device is turned on. The centering signal is an active low signal. The two pedals are simple potentiometers, that is, varying resistance resistors. These have limits in voltage and are converted into a digital system using an ADC. This voltage level determines the position of the pedal, which is all that is needed to control the speed of the car. These signals are then transmitted over XBee to the car and data is logged over USB to a computer.

CAR ELECTRICAL INFORMATION

The electronics for the car consist of a KL25Z board with a plug-and-play TFC motor shield that comes with the Freescale Cup car kit. The setup had to be modified to include connections for the encoder circuitry and XBee signals while still allowing complete functionality of the servo connections, motor controllers, and potentiometers. This was done using a custom perfboard shield that connects directly to the top of the motor shield. Unused pins were repurposed, and wires and connectors were soldered to the motor shield to connect the custom shield without removing functionality. This setup was included in the design of the Lexan plates so that USB connections remained available for debugging.

The Freedom Board motor shield is a shield that is used to control the motors and allow for other sensors such as speed sensors, line cameras, and servos to be controlled. These sensors are those allowed in the Freescale Cup, and therefore to access the other pins on the Freedom Board an adapter had to be made which would sit on top of the Freedom Board motor shield and allow for external components to be connected to the microcontroller.

The external components which were connected are the XBee for wireless communication, an additional servo for the wireless camera, and encoders on the rear wheels. The motor shield already allows for motor control through the H-Bridges present on the shield. This functionality was not modified and was left untouched with the new adapter board. The adapter board needed to be placed on top of the shield and therefore required certain pins to be moved and/or removed. The servo pins which the shield already has connections had to be moved and jumped through the adapter board since the adapter board was designed to fit on top of the motor shield using the servo male pins as a placement harness.
Figure 4 shows the top view of the board where the input signal bus would be connected (A) and where the XBee would be placed (B). In Figure 5, which is the bottom view, the adapter board harness for the motor shield can be seen (C). Figure 6 shows the adapter board on top of the motor shield. Certain filtering capacitors were also required to smooth the signal coming from the encoders as the H-Bridges were generating noise. These filtering capacitors were attached across the battery terminals and motors.

![Console Electronics Schematic](image)

**Figure 7. Console Electronics Schematic**

**PROGRAMMING**

As both the console and car use KL25Z microcontrollers, we used mbed as our primary compiler. This online software is openly available and has a wealth of documentation and premade libraries submitted by an online community of users that allow for easy implementation of many basic functions. The UML seen in Fig. 8 was created to show an overview of the coding done in this setup. This shows the interplay between the console and car microcontrollers, along with the libraries used for each system.

In its most basic sense, the user inputs steering wheel and pedal commands into the console microcontroller, which converts the signals and sends them over serial using the XBees to the car. The car decodes the data and calculates the desire target speeds using this input and feedback from the encoders through a PID controller. The encoder speeds are also sent back through the XBees to the console while the speed and turning values are written to the servo and motors. The console then sends the raw encoder and user input data over USB serial to a computer, where it is decoded and graphed using an Excel VBA script and ActiveX add-on.

**CONTROLS ALGORITHM**

The controls application for Dr. Cockburn’s computer engineering controls class is a simple torque vectoring algorithm. This algorithm simply attempts to produce drive wheel speeds that minimize wheel slipping, in our case this is done to optimize turning performance. During a turn, the inner wheels will need to spin slower to improve efficiency, reduce slip, reduce tire wear, and reduce stress on driving components.
A mathematical model is required to determine the car’s turn radius in order to correctly supply reference velocities for a torque vectoring algorithm. The physical model of the car turning is a simple, no slip, rigid body, low-speed turn model. The no slip assumption implies that the car’s wheels are not slipping radially or tangentially. The rigid body assumption implies that all parts of the car move through a given turn with the same angular velocity. Lastly the low-speed turn assumption means that lateral forces are negligible; most importantly this assumption decouples turn radius from speed and simplifies the analysis. The turn radius of the car in the model depends only on the average Ackerman steering angle of the front wheels. The final equations for the inner and outer wheel reference velocities reduce to Eq. (1, 2).

\[
\text{Inner speed} \quad \frac{v_1}{2} = \text{Throttle speed} \left( 1 - \frac{w}{2b} \tan(\delta) \right) \\
\text{Outer speed} \quad \frac{v_2}{2} = \text{Throttle speed} \left( 1 + \frac{w}{2b} \tan(\delta) \right)
\]

Where \text{Throttle speed} is the speed to which the current throttle input corresponds [m/s], \(w\) is the track width [m], \(b\) is the length of the wheel base, and \(\delta\) is the average Ackerman steering angle.

In order to study the system, a continuous-time model was created using Simulink. The system uses the known vehicle parameters and accepts inputs of \(\delta\) in degrees and \text{Throttle speed} in m/s, using Eq. (1, 2) to generate the reference signal for the inner and outer velocity control loops which operate in m/s. The inputs of the left and right control loops are then assigned to inner and outer wheel reference speeds using logic based on the current value of \(\delta\) to switch the two when \(\delta\) crosses zero. The model defaults to an extremely slight right turn when given a value of \(\delta\) that is very close or equal to zero.

The motors in the system are characterized using a first order time constant approximation. Dead zone and saturation effects are accounted for in the model as well. The compensator is a PID block, which applies a PID compensator of the form of Eq. 3.

\[
CO = e(t) \left( P + I \int e(t) + D \frac{\Delta}{1 + N} \right)
\]

Where \(CO\) is the controller output, \(e(t)\) is the error signal at time, \(t\), \(P\) is the proportional coefficient, \(I\) is the integral coefficient, \(D\) is the derivate coefficient, and \(N\) is a filtering coefficient.

Several parameters for the vehicle had to be determined in order to analyze the system. Wheelbase, track width, tire radius, and total weight can be measured directly. The time constant, \(\tau\), was obtained fitting a first order exponential curve to the system given a step input from low to maximum speed with the values recorded from the encoders. The range of the steering angle, \(\delta\), was designed into the car when the new transaxles were designed. The range of tangential velocity, \(V\), simply provides a reasonable range of expected values. These parameters can be seen in Tab. 1. Note that \(V_{\text{sat}}\) and \(T_{\text{sat}}\) values have dead zone take into account.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wheel Base</td>
<td>b</td>
<td>0.1981</td>
<td>m</td>
</tr>
<tr>
<td>Track Width</td>
<td>w</td>
<td>0.1397</td>
<td>m</td>
</tr>
<tr>
<td>Tire Radius</td>
<td>r</td>
<td>0.0356</td>
<td>m</td>
</tr>
<tr>
<td>Time Constant</td>
<td>τ</td>
<td>1.4</td>
<td>s</td>
</tr>
<tr>
<td>Total Weight</td>
<td>Fg</td>
<td>1670</td>
<td>g</td>
</tr>
<tr>
<td>Steering Angle</td>
<td>δ</td>
<td>-35 &lt; δ &lt; +35</td>
<td>deg</td>
</tr>
<tr>
<td>Tangential Velocity</td>
<td>V</td>
<td>0 &lt; V &lt; 6 (ideal)</td>
<td>m/s</td>
</tr>
<tr>
<td>Dead Zone</td>
<td>DZ</td>
<td>+/- 2.78 (0.5 PWM)</td>
<td>m/s</td>
</tr>
<tr>
<td>Velocity Saturation - DZ</td>
<td>V_{sat}</td>
<td>+/- 1.52 (0.75 PWM)</td>
<td>m/s</td>
</tr>
<tr>
<td>Torque Saturation - DZ</td>
<td>T_{sat}</td>
<td>+/- 0.7 (0.75 PWM)</td>
<td>m/s²</td>
</tr>
</tbody>
</table>

Table 1. Relevant System Parameters

For simplicity and speed, the PID library built into mbed was chosen to run the control system on the car’s KL25Z. This requires no input from the computer at the console, and thus reduces system lag and simplifies communications. Throttle and steering input are communicated to the car through the pair of XBee modules, and Eq. (1, 2) are used to generate the reference velocities for the PID loops. The PID controller in mbed uses Eq. 4.

\[
CO = CO_{bias} + K_c \left( e(t) + \frac{1}{T_i} \int e(t) dt + \frac{dPV}{dt} \right)
\]  

(4)

Where \( CO \) is the controller output, \( e(t) \) is the error at time, \( t \), \( PV \) is the process variable (velocity from encoders), and \( K_c, T_i, \) and \( T_d \) are the PID proportional, integral, and derivative coefficients. The PID coefficients can be modified by changing \( K_c, T_i, \) and \( T_d \) in the code. The discretization method can be seen in Eq. (5, 6).

\[
Integral = e(t) + e(t - 1)
\]

(5)

\[
Derivative = \frac{2-\frac{1}{T_d}}{2(T_d)}
\]

(6)

The integral contains an “if” statement which only integrates if the input is not pegged at a limit. This is to prevent reset windup. Note that the derivative control is unfiltered. In addition, another variable, \( Rate \), is introduced to describe the discrete time-step used by the PID controller on the KL25Z.

**TESTING RESULTS**

To evaluate if the project deliverables, we tested the final product in three areas: basic kinematics, controls, and user experience. For the basic kinematics components we measured physical parameters of the car’s performance. The accuracy of our encoder measurements was right on 3 m/s as confirmed with our timing gate and sensor data, and the steering servo accuracy was measured at ± 2% with a protractor. Using sensor data the acceleration and braking times were measured at 3 s and 2.5 s, respectively, and the minimum turning radius was measured at 0.3 m. The final car weighed in at 1670 g with a battery runtime of 1 hour and a camera runtime of 3 hours. For the controls components, we physically measured the XBee and camera operating distances to be around 50 m and 70 m, respectively. Through modifications in the code and component selection we were able to achieve smooth driving performance in all scenarios and almost completely unnoticeable delay in both user control and video feed. For the user experience components, the steering ratio developed in the code has been made adjustable, and through testing we have concluded that the original metrics were not actually desirable for our application. The pedal travel used was right on the target range at 4.5 cm. The only area where we did not meet the desired metrics was the video quality of the camera, as the 5.8GHz RF transmission used experiences interference from some indoor areas. Based on our other choices this does appear to still be the best option based on current technology, and we have done everything possible to mitigate the issue. Additionally, the camera worked very well in the Clark Gym at Imagine, which was the target demo for the project. All of these values surpassed our metrics.

**CONCLUSIONS AND RECOMMENDATIONS**

Based on our results, there are many areas in which future work could be done to increase the quality of the product and extend its usefulness into new areas. Some of our suggestions are as follows.

Following Imagine RIT, it is clear that this project is very successful at drawing the attention of small children. Though our design attempted to make the console adjustable for people of all sizes, very small children seem to be outside of this range. For the future, it would be helpful to make the pedals fixed to the table base and adjustable in height with around a foot of travel upward. More adjustment in the height of the seat and being able to reduce the height of the top table surface may be desirable. Effectively the lack in adjustability made it difficult for children to
press the pedals, which would move on the table base and had to be held in place. Since the seat had to be low to reach the pedals, children were below the height of the steering wheel, which made them pull downwards to see the screen. This broke the clamping mechanism, which had to be held in place manually by a team member. This was replaced by a bolted connection. An additional option may be a small console specifically set up for small participants, now that the functionality is fully developed.

For the visual aspects of the project, new camera and antenna options may be useful to help with multipath interference issues in some areas, but there does not seem to be a clear solution to the problem at this point in time based on cost. The microcontroller requires lower voltages than the sensor provides, which requires extra circuitry to operate properly. A change in microcontroller would result in less hardware requirements, and, therefore, a simpler design. The Freescale dashboard is something that could also be integrated into the project but time constraints and lack of support limited its use. This would add a great visual element to the display that demonstrates other Freescale products. Similarly suggestions were voiced at Imagine RIT to include aesthetic touches (brighter colors, a fiberglass car body, etc.).

The car chassis required many modifications to achieve the functionality desired. A new chassis could be used for future developments that was more structurally sound for use with other testing. These chassis would be more expensive and include higher quality servos, suspension, tires, and possibly better encoders. Such a chassis would also lend itself to the possibility of a car balancing controls application. The resulting bigger motors would also help with the limitations in the current chassis with motor output so that larger PID coefficients can be used to get more dynamic system responses.

There will be a new Freescale MATLAB interface that will be useful for designing the controls of the car. This was being developed while the project was ongoing, but it should provide a direct interface for using the TFC kit motor shield with Simulink. Based on conversations with MATLAB representatives at the Freescale Cup, there are still some issues with noise being emitted by the h-bridges in all aspects of the microcontroller’s operation. Therefore we believe it might be useful to separate out the operations onto separate microcontrollers to isolate the system noise in the future. This would require the addition of another XBee and the configuration of a three node network, but more precise data sampling (on a dedicated board running only that process) could then be completed with less noise and interference.

A final area to investigate would be the issues encountered when enabling two-way communications on the console. This caused the XBees to intermittently lose connection, but the functionality was only used for data logging and was thus disable during demonstrations. Despite our attempts at resolving the issue the problem persists, so in future projects getting this issue fully resolved would be optimal.

Overall the project is a major success that can be built off of in future projects to build truly inspiring exhibits. Just like the original intent, the project is fun and shows what engineers are capable of creating.

REFERENCES

ACKNOWLEDGMENTS
Many people helped make this project possible across multiple departments at RIT. We would like to acknowledge them in Tab. 2 and thank them for their generosity.

<table>
<thead>
<tr>
<th>Organization</th>
<th>Person</th>
<th>Investment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freescale Semiconductor</td>
<td>Andy Mastronardi</td>
<td>Electronic Components, Software Assistance</td>
</tr>
<tr>
<td>ARM Processors</td>
<td>-</td>
<td>Electronic Components</td>
</tr>
<tr>
<td>MBED</td>
<td>-</td>
<td>Software Development Suite</td>
</tr>
<tr>
<td>RIT MSD</td>
<td>Mark Smith, Chris Fisher</td>
<td>Funding and Purchasing Support</td>
</tr>
<tr>
<td>RIT CE Department</td>
<td>Dr. Becker-Gomez</td>
<td>Adviser, Components</td>
</tr>
<tr>
<td>RIT CE Department</td>
<td>Dr. Cockburn</td>
<td>Controls Customer</td>
</tr>
<tr>
<td>RIT CE Department</td>
<td>Rick Tolleson</td>
<td>Components, Advice</td>
</tr>
<tr>
<td>RIT ME Department</td>
<td>Bill Finch</td>
<td>Computer, Wireless Card, Software Assistance</td>
</tr>
<tr>
<td>RIT ME Department</td>
<td>Machine Shop</td>
<td>Table Surfaces, Casters, Fabrication Assistance</td>
</tr>
<tr>
<td>RIT FMS</td>
<td>Chris Furnare</td>
<td>Console Car Seat</td>
</tr>
</tbody>
</table>

Table 2. Sponsors and Acknowledgement