ECE 4961- Group 2 - MarioKart Bike Design Phase 1

Reed Hester, Chase Griffin, Leah Faulkner
ECE Students, Tennessee Technological University
Cookeville, TN, United States of America
lrhester42@tntech.edu
cagriffin42@tntech.edu
lmfaulkner42@tntech.edu

***Abstract* --- This report dives deeper into the high-level design of the MarioKart bike. A block diagram of the system with descriptions for each block is given. Analysis, testing, and broader implications are also explored.**

Keywords --- Capstone Design, gaming, emulation, exercise, sensors, electromagnetics, engineering

I. INTRODUCTION

The Mario Kart exercise bike project is the result of an idea to bring physical activity together with the excitement of the racing video game experience. The objective with this idea is to design and develop a practical way to exercise using the traditional resistive elements of an exercise bike while simultaneously controlling a player in a game of Mario Kart in a simple-to-use, responsive, and enjoyable way. By using the bike as a direct controller, the player will be able to play as fluidly as they could with a standard controller with the added benefit of physical activity. Ideally, the bike will serve as a tool to assist in promoting a healthy lifestyle in an entertaining way.

To achieve this goal, the use of several external control systems combined with the inputs from a standard game controller will be employed. Microcontrollers with code interfaces set up to specifically target the operations for each subsystem of the controls will be utilized as well. These powered subsystems will consist of an adjustable resistance method, a pedalling speed sensing system, an encoded steering system, and an actual standard game controller. The resistance method will vary between two experimental versions of the bike. One method will consist of a magnetic resistance system involving a linear actuator that can incrementally increase or decrease the physical distance of a magnet from the bike wheel, thus creating dynamic resistance based on said distance. The other method of resistance is to use a DC motor connected to a MOSFET that allows the motor to be shorted on command, which creates resistive force when the motor is being spun. The pedalling sensor system will involve a tachometer sensor that measures the speed of the revolving pedals and uses that information to control the acceleration of the player's vehicle. The steering mechanism will govern a rotary encoder which will be used to interpret directional controls. Finally, in order to simplify menu navigation and other game interactions, a standard game controller will be included as well with the traditional button layouts and usage. Each of the sub-microcontrollers in these systems will be routed into a main microcontroller or RaspberryPi device that interprets the received data into usable control information that is forwarded to the device running the game. The game version itself will initially be the Nintendo64 copy running on an emulator program contained on a PC. These design choices have been made out of a combination of the apparent ease of use and design. The use of an emulator software will only be for testing purposes in order to verify the operation of all control systems and will not be used for distribution or copying of the game software or license. An official copy of the game will be employed in the final product. In order to be practical and efficient, the systems on the bike must be simple to understand for the user, easy to access for a varied user base, safe to operate while still being an effective exercise tool, and responsive enough to allow reasonably quick game control.

The measures used to determine the success and achievement of the specifications will vary for each subsystem. Some of them, such as the control accuracy and ease of use, can be qualitatively assessed through regular use. Analytically, MATLAB, LTSpice, and Simulink simulations will be used for the DC motor circuit, the magnet setup, and the pedalling system. Also, control fidelity and response time can be measured using input lag detection software.

A. Relevance in the Larger Process

This design document will ultimately be used as a guide for the development and building of a functional control system as is defined by the specifications and systems being described. Through careful documentation of each step, the aim is to eliminate as much ambiguity as possible in order to streamline the development process. Additionally, using the structure defined by each section, the project progress will follow a set timeline such that each step in the process can be clearly and uniquely identified.

B. Document Outline

In the following sections of this design document, the systems previously explained will be described in further detail including higher fidelity specifications and constraints, justifications for the methods chosen, the testing and experimentation options available for success measurement, any relevant ethical or professional concerns, a block diagram outlining the project as a whole, and lastly a chart depicting the necessary skills and resources to design and construct each portion of the project.

II. BLOCK DIAGRAM AND SPECIFICATIONS

On the next page is a diagram of the comprehensive layout of the entire system. Each block is categorized by a column letter and a number. The diagram follows a similar protocol to the universal modelling language (UML). Each block will be paired with a comprehensive description of its purpose in the whole system and how it interfaces with the other blocks it is connected to. The categorization method we implemented will ensure that these descriptions are clearly recognizable and understandable to anyone who may read this. Though the content of the descriptions may be technical, a non-technical reader can still interpret how the system is connected as a whole.

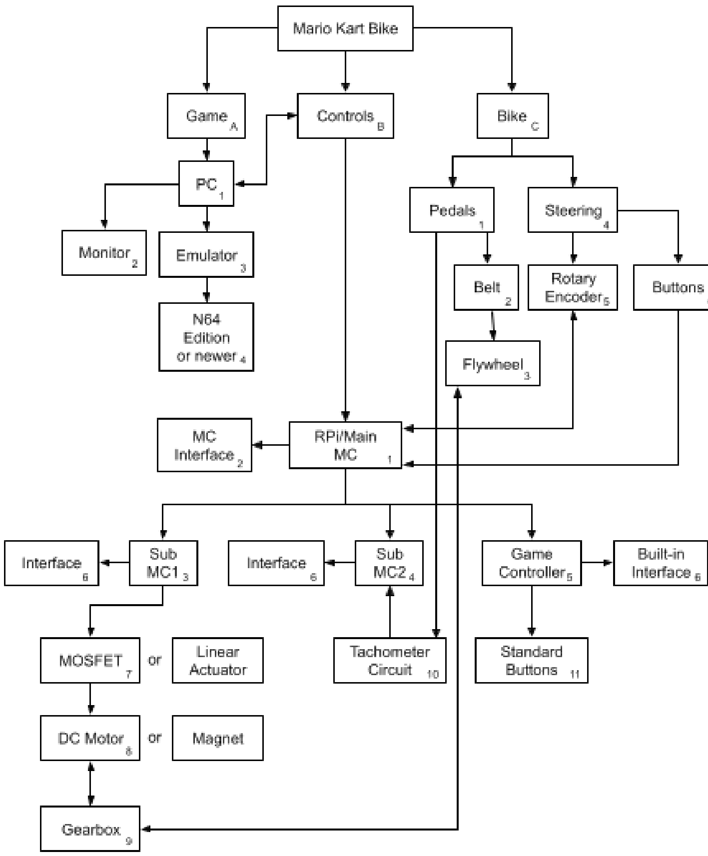

Figure 1. Block diagram of the MarioKart bike system.

A. Game

This column represents all components related to running and displaying the game program.

A1) All control data will be routed into a PC that contains and runs the emulator software and game copy. It will be connected to the main MC/RPi through a serial data line on a USB port to receive and interpret control data and relay it to the game. This will be powered by a standard 120V wall outlet or a battery.

A2) A display to allow the user to view the game process will be connected to the PC running the game. Powered by a standard 120V wall outlet.

A3) The emulator software being used is the ProjectN64 Nintendo64 emulator which will run on the designated PC. It allows for direct and custom control mapping from the connected microcontroller.

A4) The game copy used will be at least the Nintendo64 version of MarioKart.

B. Controls

This column represents the structure of the control system laying behind the scenes. All pieces of the bike will connect back to this main hub before communicating with the emulator.

B1) The main microcontroller or RaspberryPi device. This will ideally be connected by USB to all of the sub-microcontrollers and will interpret the data derived from each into usable control data that is sent to the connected PC. It will be powered through USB connection by the PC.

B2) The code interface of the main MC/RPi will be used to convert each of the received data signals into control inputs that can be read by the PC.

B3) Sub-MC1 will be the microcontroller that controls the MOSFET attached to the DC motor. It needs to be able to output 2.5 to 5 V via a digital to analog converter. It will be powered via USB from the main MC.

B4) Sub-MC2 will be the microcontroller that interprets the data received from the tachometer. Using a pulse width modulation system, control data output for the acceleration will be pulsed based on the speed of received signals from the tachometer to allow for gradual increases in speed until full speed had been achieved for the player in the game. The MC will be connected by USB to the main MC/RPi. It will be powered by the USB port.

B5) The standard game controller will operate as normal regarding inputs. It will be connected to the main MC/RPi by USB and will send its control input signals to be interpreted. It will be powered by the USB port.

B6) Each microcontroller will have a code interface that allows it to convert the signals obtained from external sensors and parts into the data necessary for the main MC/RPi to generate control input. The game controller will already have a built-in interface that achieves this result for that subsystem.

B7) The MOSFET will likely be a power MOSFET, built to handle high voltage and high current. It will need a threshold voltage lower than 5 V in order to be controlled by a microcontroller. Since high current will be running across the terminals of the DC motor, a power MOSFET is necessary. Alternatively, a linear actuator could be used. This will also need to be controllable by low voltage. It should have a long range of motion in order to provide more variability to the resistance.

B8) The DC motor needs to be high RPM and able to withstand the power generated by a human on a bicycle. Most people can generate 50-100 W on a bike, with around 30 ft-lbs of torque at 50-60 RPM. The motor needs to be a permanent magnet variant in order for our application to work. The motor itself won't be powered, just controlled by the MOSFET. Most motors can't handle the high torque produced by a bicycle, so a gearbox will be necessary. Alternatively, a permanent magnet could be used. Magnetic resistance stationary bikes come pre-equipped with the magnet and flywheel needed.

Traditionally, the position of the magnet is set by the user via a locking mechanism. This would be replaced by the linear actuator, controlled by Sub-MC1.

B9) Since the DC motor requires a low-torque high-RPM input, a gearbox will be necessary to convert the motion from the bike. The gearbox will step down the torque and step up the RPM. A ratio of 1:30 will be sufficient. It needs to be able to handle 30 ft-lbs and output a high RPM at around 100 W. The gearbox will be directly connected to the flywheel on the bike.

B10) The tachometer will be connected to Sub-MC2. Its job will be to monitor the RPM of the pedals, then send a signal to the control system. This signal will be converted to the in-game acceleration of the vehicle. It will be powered by the MC itself.

B11) The standard layout of a video game controller will be necessary to allow the user to navigate the console menus. The bike's only functions will be to drive, steer, and use items. Therefore it cannot control the console interface alone.

C. *Bike*

This column represents the physical build of the bike. These are the pieces that the user will interact with directly. These will likely be preinstalled on a stationary bike that will be purchased.

C1) The pedals attached to the stationary bike will be suitable for our application. They are attached to a belt, driving the flywheel. There will also be a tachometer stationed near the pedals in order to monitor their RPM. For safety, the pedals should have foot straps/holsters so that the user doesn't slip.

C2) The belt connecting the pedals to the flywheel comes preinstalled. It is usually a rubber belt with teeth that groove into the gear teeth on both the pedals and flywheel. Since it's already built to handle the torque produced by a person, it should still serve our application.

C3) The flywheel will be connected directly to the gearbox and driven by the belt and the pedals. It is usually large and weighted to allow a smoother pedaling experience. Some modification will be needed since the flywheel won't have a hole that can connect to the shaft coming out of the gearbox.

C4) The handlebars attached to the stationary bike will have to be modified since they are designed to not rotate. The column the bars are attached to is usually hollow, so putting a pipe inside of it and attaching the bars to the pipe would allow them to rotate freely. To keep them from spinning 360 degrees, buffers will be put in place via two springs. These springs will both keep the bars from rotating too far and bring them back to the center position. The handlebars will also need to be modified to hold buttons for playing the game.

C5) A rotary encoder will be necessary to change the physical motion of steering into game data. The encoder would be mounted below the pipe running through the steering column and attached directly to the pipe. The springs on the pipe will also act as buffers for the encoder to prevent 360 degree rotation. It will be powered directly by the main MC.

C6) These buttons will serve as the drift, item, and back throw functions in-game. These are needed to complete the immersive experience. They could be mounted via 3D printed parts or other means. Repurposing pull-levers from bicycle handlebars would be a clean way to mount the buttons. The buttons would be mounted on the inner portion of the levers, so that pulling the levers towards the handlebars will press the button.

III. MEASURES OF SUCCESS

For this project, several tests can potentially be conducted to evaluate the success of the bike, controls, and user experience. The goal for the maximum control input latency is set to a standard 40 ms, which will ensure a fluid game experience. The first method to determine the achievement of this will be to have each team member as well as other external participants qualitatively test the input lag in a gaming session, and then determine whether it either was negligible or noticeable enough to cause an issue while playing. In the case of a success, a majority of the participants should think that it is negligible. The second and more empirically sound method would be to use an input lag detection software that is free and open source. This method would be the more accurate of the two and will serve as a more detailed marker for success in general.

In addition, another important aspect of the success of this project is the satisfaction levels of the users in terms of quality of gaming experience. This metric will be determined based on a survey conducted after the user plays through a full session on the bike. The first survey needs to assess whether users found satisfaction in using the device. It will be designed to quantify on a 0-5 scale the satisfaction of the gameplay and experience: 0 (not enjoyable) 1 (somewhat unpleasant), 2 (neutral), 3 (somewhat pleasant), 4 (enjoyable), 5 (very enjoyable). The majority of surveys on the satisfaction scale should fall in the 3-5 range to qualify the bike as a success.

The most important measure of success in terms of the exercise experience on the bike is to ensure that the resistance levels fall within a certain practical range, and that the resistance system functions properly in general. Youth and adults can test 50cc, 100cc, and 150cc modes in the in-game race settings, which will correspond to progressively harder resistance settings on the bike. The survey will inquire as to which level was picked, as well as the intensity rating of the resistance experienced during the race. This will be quantified on a 0-5 difficulty scale: 0 (not at all), 1 (somewhat), 2 (neutral), 3 (somewhat difficult), 4 (challenging), 5 (Too difficult). The 50cc mode should fall in the 1-2 range, the 100cc mode should fall within the 3-4 range, and the 150cc mode should fall within the 4-5 range in the ideal outcome and implementation. These survey methods are examples to demonstrate what we could ask the users after using the system to obtain qualitative data on the system experience. Further research will be conducted upon construction of the resistance system to find a preexisting and reliable survey method that prevents bias and follows effective standard protocols in order to generate useful and applicable survey results.

IV. ANALYSIS AND EXPERIMENTATION

For Design Phase 1, it was very imperative that we make sure the DC motor system will function the way we expect it to as it was the most critical system in our design. To test the efficacy of the MOSFET, LTSpice simulation was used. The figure below the circuit we built for the analysis.

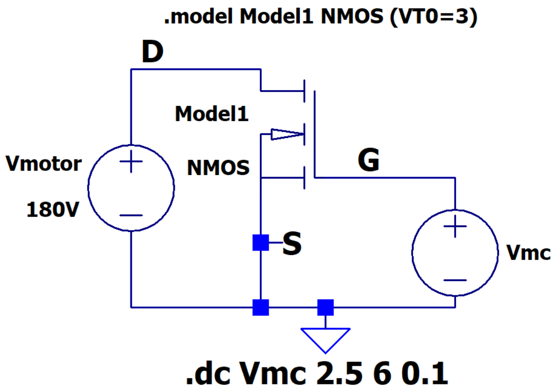

Figure 2. LTSpice Circuit for DC Motor and MOSFET

The voltage source "Vmotor" represents the DC motor. If the motor is spinning at a constant RPM, it will produce a constant voltage. The motor we will likely choose has a max of 180 Volts, so we chose to use that max value to represent the whole range of voltages seen by the MOSFET. Next, we used a standard NMOS component and a model command to represent the MOSFET we chose. The model command sets the threshold voltage to 3 Volts. Finally, the "Vmc" voltage source represents the signal being sent from the microcontroller to the gate of the MOSFET. The simulation command ".dc" was used to run a DC voltage step. The Vmc source started at 2.5 Volts and was stepped to 6 in increments of 0.1 to show how the resistance of the MOSFET varies as the gate voltage changes. Below is a graph of the change in resistance as the voltage increases.

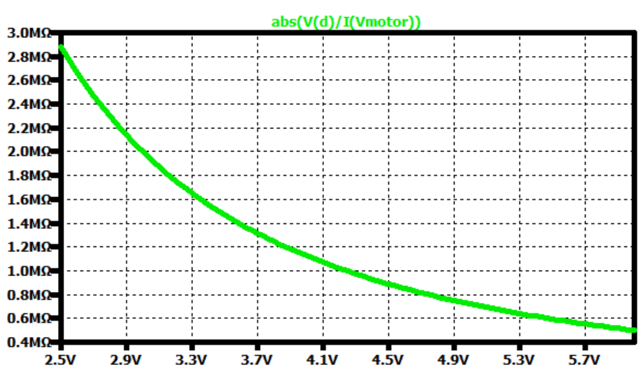

Figure 3. MOSFET Resistance over Gate Voltage

As shown in the graph, the resistance decreases as the gate voltage increases. The change in resistance begins to decrease towards the higher voltages because MOSFET is approaching the saturation region. Further, since there is a proportional relationship between the current flowing through the motor and the torque generated, we can conclude that the torque will increase proportionally as the resistance decreases.

For the rest of the systems in the bike, the power connections will be modeled in LTSpice. The power drawn from each microcontroller will need to be shown, as well as wall connections like for the monitor and the PC. The completed DC motor resistance circuit will also be simulated.

V. ETHICS AND BROADER IMPLICATIONS

This project aims to avoid any copyright issues by following guidelines provided in Nintendo documentation. Nintendo explicitly states that game ROMs downloaded from the internet are copyright infringements. However, the company has claimed that the use of a backup copy made from an owned copy of the game does not pertain to ROMs downloaded from external sources via the internet. They do not allow selling or distribution of the game from a copied license in this way. [1] The emulated game copy will be used for testing purposes and not for commercial use or exchange. Though there are no legal precedents or trials relating to the individual use of emulators or ROMs, after testing is completed, an officially licensed copy of the game will be used for the final state of the project in order to avoid any legal issues.

The Mario Kart Bike will be designed with consideration of environmental impact. The main housing material will be made of recycled filament for 3D-printers to house buttons. In addition to using environmentally friendly materials, the bike will implement reusing parts such as used bike brake handles to mimic realistic breaking for the player.

The exercise bike will still comply with any regulations or safety standards regarding exercise equipment as we will not be modifying the frame or operational parts in any way that could compromise the integrity or effectiveness. Also, analysis and testing of safe design and operation of the resistive methods will be conducted to prevent physical harm to the user. These safety regulations have determined many design choices. For example, the resistive capacity will be limited to follow existing resistance safety standards on commercial exercise bikes, and the rate of change of resistance will be tuned within the MC interfaces to prevent sudden stops or adjustments that may lead to injury. Also, instead of using a conventional bike that is converted into stationary function, a traditional static exercise bike will be employed and modified to simulate the steering function of a regular bike. This is to allow for greater stability and easier mounting of the motor/magnet systems, again in the interest of preventing injury to the user.

The team conducted an architectural risk analysis on the bike. The consensus is that there are few attack surfaces based on the design. This is evident because there are no wifi or bluetooth connections, although bluetooth is a possibility if it is added in later versions. However, there are serial data connections between the microcontrollers of each subsystem. To mitigate risk, the serial data will be encrypted via code. The team identified the cable linking the controller to the switch as a trust boundary. Unfortunately, the switch data cannot be encrypted. This boundary will be protected with a barrier to house the cord so that the switch data is protected.

There are additional risks to the system's performance that pertain to user safety. To protect the bike system from being manipulated by the user, components with safety features were selected. For example, the team chose a DC motor with a junction box to house wiring. This decision was made to prevent the user from harm via electric shock.

VI. TIMESCALE AND GANTT CHART

To lay out the remaining tasks for this semester, the team used a Gantt Chart. This type of format shows what the task is, who is working on it, and an estimated timeframe for the completion of the task. The chart is broken down to a day-by-day scale, extending to December 15th - the end of finals week for the Fall 2021 semester. Each team member created their own tasks and gave a liberal estimation of the time required to complete each one. The final page of this document shows the full-size chart.

REFERENCES

[1] "Legal information," *Nintendo*. [Online]. Available: https://www.nintendo.com.au/legal/information. [Accessed: 01-Nov-2021].