**ECE 4950 Project 3**

**Team Megafist**

**Group 13**

**Valerie Moran, Brandon Cuttino, Ben Liggett, Sarah Anderson, and Rocky Aho**

**November 18, 2020**

**Executive Summary:**

To begin this project, we started by developing the engineering requirements for the motor control subsystem. We did this by converting each customer requirement into an engineering requirement and then determining a test on the prototype to verify that the engineering requirement was met. Afterwards, we analyzed what hardware-in-the-loop means and how we could apply it to our project. We then began building and documenting the hardware design for the motor subsystem. Once we had the hardware built, we developed the software to run the design. Throughout this process, we evaluated and documented hardware-in-the-loop responses of the motor. After developing the code, we then developed a graphical user interface to run the system. Afterwards, we began analyzing the life cycle of our design, developing an employee training program, and analyzing the safety of our design. Throughout this project, we learned more about motor control, designing a prototype, creating a graphical user interface, assessing the life cycle of our design, developing an employee training program, and assessing the safety of our design.

**Engineering Requirements for the Motor Control Subsystem:**

**Table 1: Engineering Requirements for the Motor Control Subsystem**

|  |  |  |
| --- | --- | --- |
| **Customer Requirements** | **Engineering Requirements** | **Prototype Test** |
| Clock Hand | Attachable sturdy arm for the motor | Clock arm can distinctly point to each of the eight different well locations. |
| A subsystem to Output Position Identification | A DC motor used to turn a clock arm towards a single well location. System capable of pointing to all eight different locations. | The motor rotates the clock hand to the correct location. Motor can point to all eight well locations. |
| The system selects the closest well location when there are instances of multiples of the same color. | Place two markers of the same color on the board and have the system correctly choose the position of the closest marker. |
| Develop a system capable of identifying all locations of markers on the board. Able to identify only red markers. Able to identify only the red markers, then only green markers, and then only yellow markers. | All three scenarios are programmed and function correctly. |
| Closed-loop Control System | The closed-loop system performs both location identification and output identification. DC motor utilizes location identification from the camera subsystem to determine how far to turn the clock arm. | The system requires no user input after the initial color selection. Gains determined and documented in a closed loop system to create a smooth motion for position identification. |
| The motor returns to a set rest position after the completion of position identification. | The motor rotates back to the initial set starting position after 3 seconds of pointing to the marker’s position. |
| Simple and Sufficient Graphical User Interface | Interface for the user to start each of the three different tests. Initialization button to take initial background image. | GUI contains a minimum of three buttons to start each of the three tests. Also has a button to start the motor and take a background image |

**Overview of Hardware-in-the-Loop:**

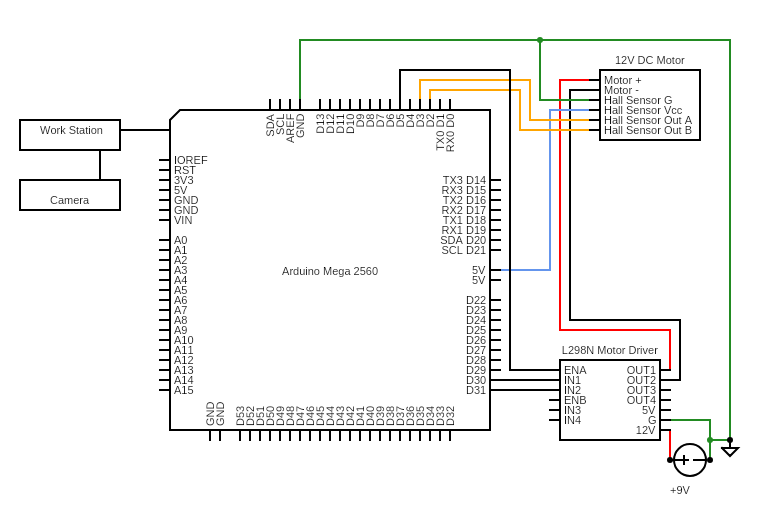
Hardware-in-the-loop is a type of simulation that allows manufacturers to apply real-world scenarios to a product and test its functionality. HIL allows the producers the ability to test individual subsystems without needing the entire system built, which can save time and money by not needing the entire system to run a test. This system is very useful for large scale products with many sensors, cameras, and radars so that each individual sensor or camera can be tested with real-world scenarios to determine its functionality. For example, if a company was building a new car, they would lose substantial amounts of money and time if they needed to create the entire car just to test its new alert system, but by using a hardware-in-the-loop system, the manufacturers can test and evaluate each subsystem prior to the completion of the entire car.

A hardware-in-the-loop system differs from a full simulation since hardware components are used in the hardware-in-the-loop simulation. A full simulation is done entirely through software, where every component is produced mathematically and modeled through software. The testing from a full simulation will show the modeled expected results of the system. A hardware-in-the-loop system also uses mathematically modeled components but also includes hardware components. This functionality allows the system to test the hardware components to see how they are functioning compared to the simulated components. This system allows the hardware components to be tested under the same conditions as the simulated components. By utilizing the hardware-in-the-loop simulation, manufacturers can pinpoint errors in the hardware designs without needing the entire system built.

The hardware-in-the-loop system has many strengths, including saving time, saving money, and lowering risks. The hardware-in-the-loop system can save time and money by finding errors in the design prior to the completion of the surrounding components. It also allows the manufacturers an easy way to evaluate the components. Additionally, the hardware-in-the-loop system can lower risks and costs by performing damaging tasks virtually. For example, a car would not need to perform full-scale crash tests numerous times just to test each individual component involved with crash prevention. By using hardware-in-the-loop simulations many costly tests can be automated or minimized.

**Document Hardware:**

The starting hardware connections, as shown in **Figure 1**, are from the workstation to both the Arduino mega and the camera. The mega is wired to both a 12V DC motor and an L298N motor driver chip. PWM digital pin 5 is connected to the enable A pin on the motor driver. This connection is used to turn on and off the motor through PWM pulses. Digital pins 30 and 31 are connected to IN1 and IN2 of the motor driver. These connections are used to determine the direction the motor will spin. Next, a standalone 9V power supply is connected to the 12V input pin on the motor driver. Note that the motor and motor driver can function on inputs less than 12V. Next, the ground pin on the motor driver is connected to both the ground of the external power supply and the ground of the Arduino. This connection allows the motor and motor driver to function with two different power supplies. The last two connections for the motor driver are from output pins 1 and 2 to input pins Motor + and Motor - of the 12V DC motor. Digital pins 2 and 3 are interrupt pins, and they are connected to hall sensor output pins A and B. Next, a ground pin from the Arduino is connected to the hall sensor ground pin of the motor. The last connection is between a 5V pin from the Arduino and the hall sensor Vcc pin of the motor.

****

**Figure 1: Hardware Wiring Diagram**

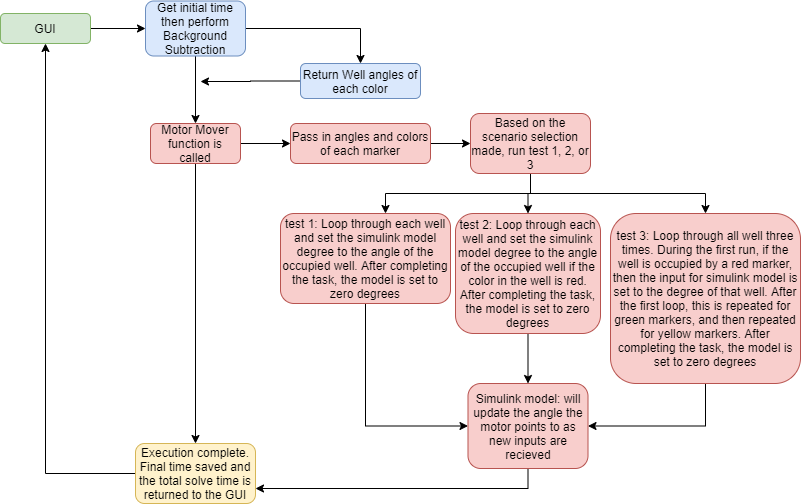
The setup for project 3 is shown in **Figure 2** below. The frame is made from half-inch PVC pipe. Three legs make up the frame, and they each measure at36 ¼ inches. At the bottom of each leg, a half-inch cap is screwed into the baseboard. The baseboard is a 15x 32 inchsheet of sanded pine plywood with its natural color of medium brown. The baseboard is large enough to encompass the current project and the future additions to the project that will take place in this development, with the possibility that it can be cut to a smaller dimension if needed. At the top of each leg, a 90 degree connector is attached. This connector is attached to another half-inch PVC section that measures 6 ¾ inches. At the center of the structure, a 3-way connector is attached to the three legs of the structure. Having the three legs tied together at the top makes the entire setup tie together in one solid structure. On top of the three-way center connector is a USB 2.0 camera. This camera was electrically taped down so that the camera faced straight down to the game board. The legs were designed so that the captured image didn't include any unnecessary area. With the camera at this height, the pixels per square inch were 11524.49. This value is an acceptable resolution, and the camera can be considered a good sensor. The gameboard was provided to each team and measured at 11 inches across. Four sections of half-inch PVC cut at 3 ¾ in are attached to the bottom side to elevate the height of the gameboard so the clock arm of the motor could point to each well distinctly. Eight circular disks, with a radius of 16 millimeters, were printed using an Ender 3 Pro. Each of these 0.125 inch thick chips was painted with ‘Pure White’ from Sherwin Williams along with the game board. Each chip was superglued into a well location. By gluing them, all of the washers will be closer to the surface of the game board and can be picked up easier by the electro-magnet. The washers were limited to 3 colors: green, yellow, and red. The washers were provided to each team and can not be modified. Next to the fixed structure, a lamp was glued to the plywood. This lamp is only for use if the lighting within the room is not sufficient. If the lighting is sufficient, then the lamp can remain off.



**Figure 2: Project 3 Design**

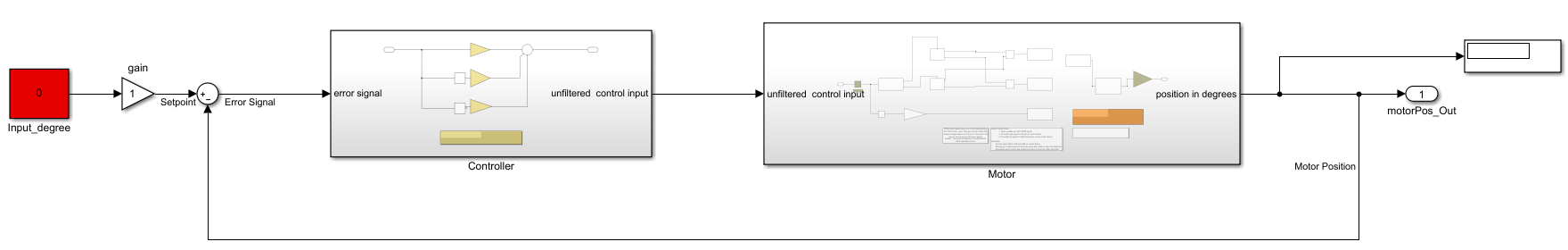
**Document Software:**

A flowchart outlining the different aspects of the code is shown below in **Figure 3a**. The code outlined below utilizes MATLAB 2020a and the Simulink add on.

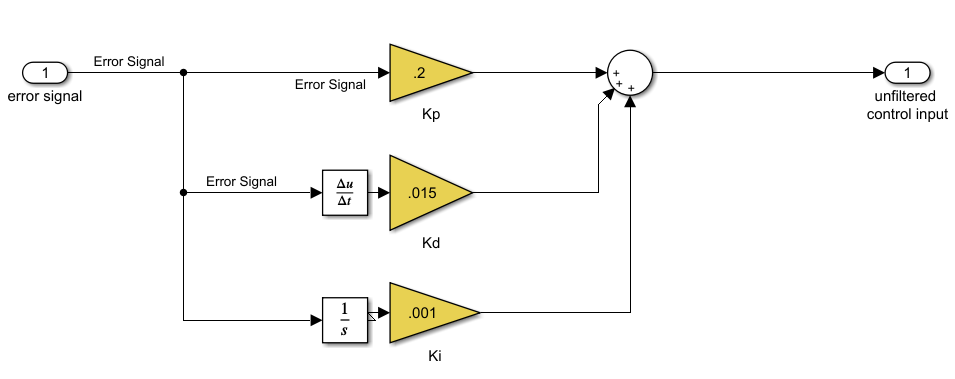
****

**Figure 3a: Software Flowchart for Project 3**

The Simulink model shown in **Figures 3b** and **3c** was supplied by the instructor. This model was used to connect our software to the motor. Within the model, the gain parameters 0.2, 0.015, and 0.001 were used for Kp, Kd, and Ki. The calculation of these values is discussed in the document motor portion of the report. Additionally, the counts per revolution for the model was adjusted to 700, such that the output degree position would be accurate. Also, each of the well locations was determined using their absolute position from the notch on the gameboard. Starting from the notch, the degree positions for wells 1-8 are 30, 75, 120, 165, 210, 255, 300, and 345. To move the motor to a well position, the corresponding angle value was inputted into the model using the set param command.



**Figure 3b: Closed Loop Motor Control**



**Figure 3c: PID Controller for the Motor**

The project three code outlined in **Figure 3a** begins with a graphical user interface. This interface is described in detail later. Prior to the selection of a scenario, the initialize system button is used. The function of this button is to take a background image of the game board. This will be utilized with the background subtraction code discussed in the Project 2 report.

After a selection is made from the GUI, the program stores the start time. This value will be used to calculate the solve time of the code. Regardless of the scenario chosen, background subtraction is used. During the execution, the color and angle of each marker on the board are determined. A string array will store the colors of each marker. An empty string will be stored in the array at the corresponding well location if a well is empty. The angle of each marker is determined by taking the middle point of the game board and subtracting by the location of the marker’s centroid. The angle received from taking the inverse tangent of these values is then looped through and matched with one of the eight well locations. After finding the angle of each marker, the color and location of each marker are returned. These values are also sent to update the GUI table. On this table, it shows which well contains which colored marker.

Next, the program calls the motor mover function and passes in the angle and color of each marker. The function looks to see which scenario was pressed. If scenario 1 was selected, then the program will use a FOR loop and will run for each of the eight different wells. For each loop, the code will compare the color values stored in a string array with an empty string. If the value stored in the string array is not an empty string, then the code will update the input angle of the Simulink model to the angle of the corresponding well location. If the well was not empty during the loop, then the code will enter an infinite while loop. The loop can only be broken when the user selects the next well button on the GUI. This selection allows the user ample time to see that the motor correctly points to each of the filled well locations. If the color found in the well was not red, then the code will loop to look at the next well. After the user selects the next well, the code will continue to the next loop. After looping through each well, the input angle of the Simulink model is set to zero, which is the rest position.

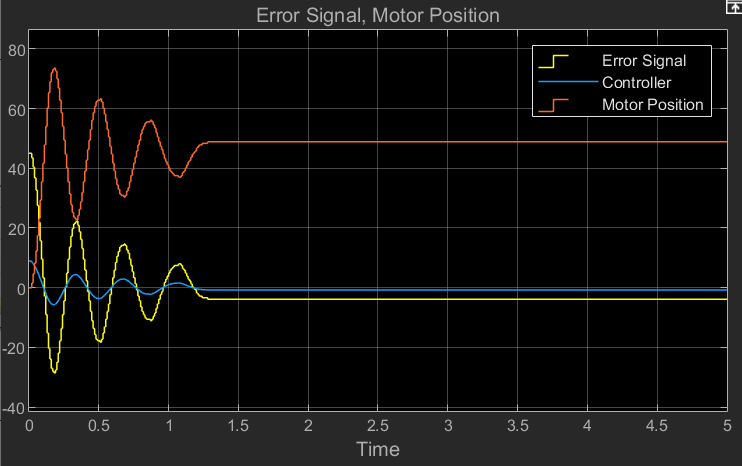
Similarly, if scenario 2 was selected, the program will use a FOR loop and loop through each of the eight different well locations. For each loop, the code will compare the color values stored in a string array with the string “red”. If the value stored in the string array is red, then the input angle for the Simulink model is set to the angle of the red marker. If the color found during the loop was red, then the code will enter an infinite loop. The loop can only be broken when the user selects the next well button on the GUI. This selection allows the user ample time to see that the motor correctly points to each of the filled well locations. If the color found in the well was not red, then the code will loop to look at the next well. After running through all eight well locations, the input angle for the Simulink model is set to zero, which corresponds to the rest position.

Lastly, if scenario 3 was selected, the program will use a FOR loop and loop 24 times. For the first eight loops, the code will compare the color values stored in a string array with the string “red”. If the value stored in the string array is red, then the input angle for the Simulink model is set to the angle of the red marker. If the color found during the loop was red, then the code will enter an infinite loop. The loop can only be broken when the user selects the next well button on the GUI. This selection allows the user ample time to see that the motor correctly points to each of the filled well locations. If the color found in the well was not red, then the code will loop to look at the next well. After looping eight times, the code then loops eight more times except it now compares the values stored in the string array to the string “green”. Similarly, if the two strings are the same then the code will set the input angle for the Simulink model to the angle of the green marker. The procedure for the next well button is the same. After looping a total of 16 times, the code is repeated, and the string array is compared to the string “yellow”. When the string stored in the array is yellow, the Simulink input angle will be set to the angle that corresponds to the yellow marker. After running through all eight well locations three times, the input angle for the Simulink model is set to zero, which corresponds to the rest position.

The motor mover function does not return any values. Once the function call to the motor mover function is complete, the code then determines the current time. This value is subtracted from the initial time and the runtime duration is sent to the GUI. After updating the run time, the code execution completes, and the user can select further scenarios from the GUI.

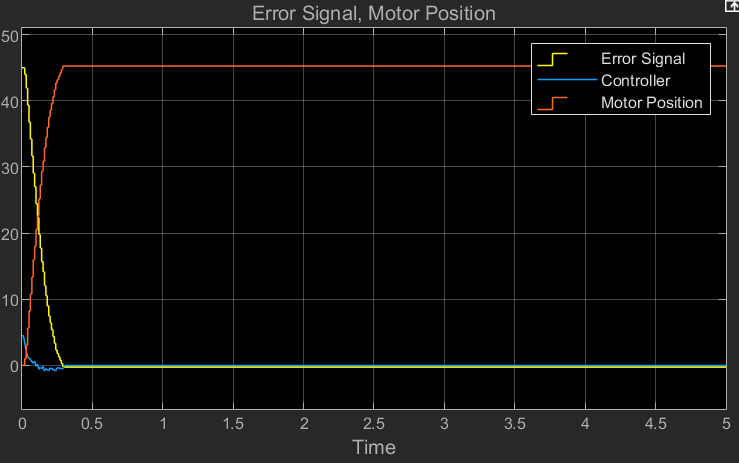
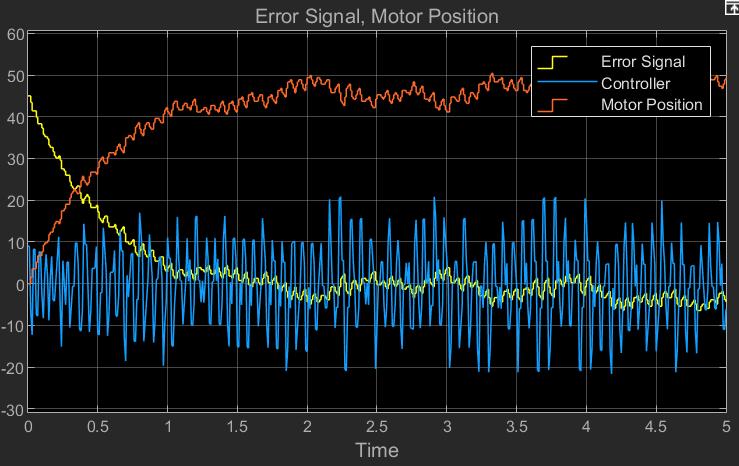
**Document Motor Experiments:**

As shown in **Figure 4**, increasing the value of Kp, the proportional gain, from 0.1 to 0.2 can increase the number of oscillations and the peak overshoot. Additionally, when comparing the final outputs from the left half of the graph to the right, we can see that the final value for the graph on the right is closer to the intended value of 45 degrees than the one on the left. This issue is caused by the steady-state error not going to zero. By only using Kp, the system has a large peak overshoot and maintains a substantial steady-state error. By adjusting this value, we can find an optimal settling time and a low steady-state error. Additionally, if the value of Kp was greater than 1, then the system would be unstable, and the motor would have jerky movements. A value of 0.2 was chosen for Kp due to the low SSE and minimal oscillations.

****

**Figure 4: Motor output when adjusting Kp from 0.1 to 0.2 with Ki and Kd set to 0.**

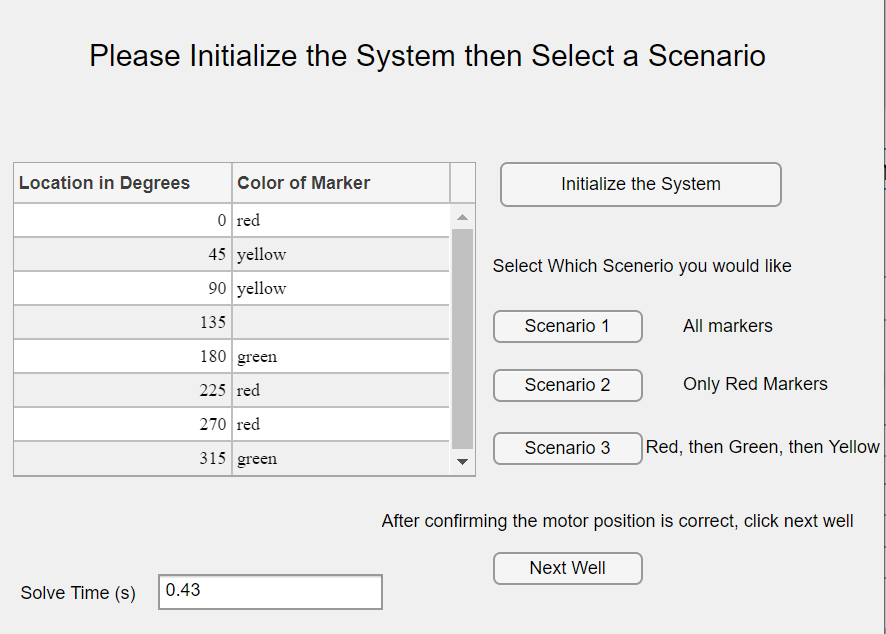
As shown in **Figure 5** below, we can see that adding both integral and derivative gains will substantially impact the system. By adjusting the derivative gain, the system will reduce its peak overshoot and have a faster response time. The response time, or slope of the Motor Position signal, increased substantially from **Figure 4** to the right graph in **Figure 5**.When adjusted, the derivative gain reduces the number of oscillations, but when this value is too large, there is too much error, as shown in the left side graph from **Figure 5**. For this reason, the value of Kd should be larger than Ki but less than Kp. Additionally, when the integral gain, Ki, is added and adjusted, the system will reduce the long term drift and force the steady-state error to go to zero. We can see in the right graph of **Figure 5** that the steady-state error reaches zero. This is necessary to remove any oscillations or jerky movements made by the motor, and this also allows the final location of the motor arm to stabilize at the intended position. Values of 0.015 and 0.001 were chosen for Kd and Ki. These values resulted in the right side graph of **Figure 5**.

****

**Figure 5: Motor Position and Error Signal when setting Kp = 0.2 and Adjusting Ki and Kd**

**Document and Evaluate User Interface:**

The GUI developed for project three is shown in **Figure 6** below. Included in the figure are text fields that help the user understand each feature of the graphical user interface.

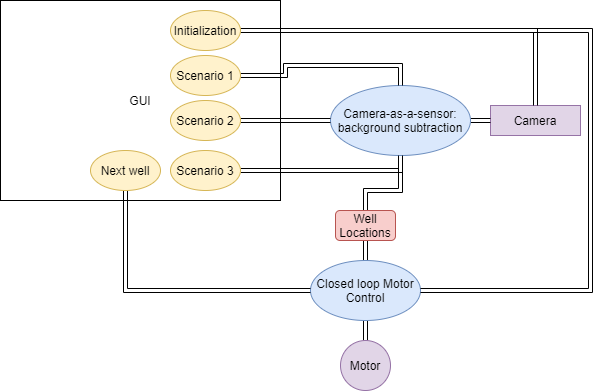
****

**Figure 6: Pop out window for the GUI**

The GUI has five buttons, one table, and an output text field. Starting from the top of the layout, the GUI starts by telling the user to initialize the system then select a scenario. The initialize button will use the camera to take a background image of the empty game board, which will be used with the camera-as-a-sensor later on, and it will start the motor simulation. This feature allows each test to have a background image that is sufficient for the lighting conditions of the room. It is also important because it starts the motor simulation, which is used for the clock arm. To the left of the initialize button is a two-column and eight-row table. Column one is constant and states the degrees for each of the eight different well locations. These angles are labeled from the top and moving counterclockwise. The second column will store the color of the marker stored in each well. The value stored will be red, green, yellow, or empty. Column two will also be updated after each time a scenario button is clicked since the table is dependent on background subtraction. This table is displayed to show the functionality of the camera-as-a-sensor system. It also shows the user that the sensor system is correctly determining the locations of each marker.

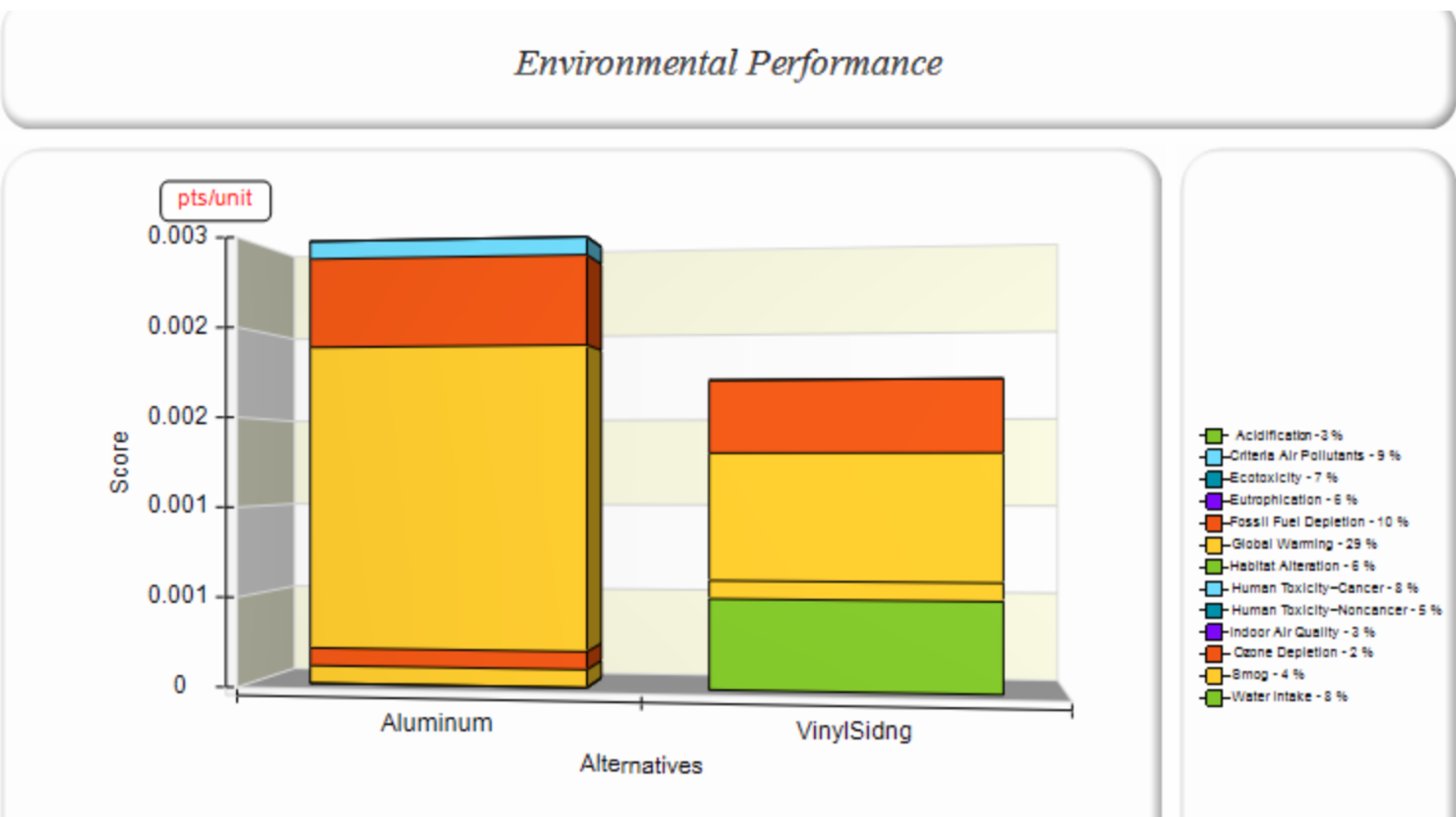
The three scenario buttons are used to run the three different tests required for project 3. By having them as buttons, they are easy to use. Once clicked, the scenario buttons will run through the background subtraction, update column two of the table with the corresponding colors of the markers, and then begin the test. If the user selects scenario 1, then after determining the locations of all the markers, the motor will then start pointing to all of the existing markers in a counterclockwise direction. For Scenario 2, the motor will only point to each of the red markers in a counterclockwise direction. For scenario 3, the motor will point to the red markers, then the green markers, and then the yellow markers. All three of these buttons will adjust the input parameter for the Simulink model based on the locations of the markers found earlier. The last button on the GUI is used during one of the scenarios. The motor will only move to the next well location if the next well button is pressed. This lets the user confirm the position prior to the arm moving to the next well location. The last element of the GUI is an output text box. This box will be used to display the solve time of the test. This does so by taking the initial start time and subtracting the final stop time. Additionally, the table and solve time will be utilized further in the final project. The table will have a third column that allows the user to select which colored marker they would like in each well location, and the solve time will be utilized to score the speed performance of the final model.

Shown below, in **Figure 7**,is the connection diagram for the GUI to the Camera-As-A-Sensor and the motor. We can see that each of the five buttons connect to either the motor, the camera, or both the camera and motor. The initialization button and scenarios 1, 2, and 3 all connect to the camera and the motor. The next well button only connects to the motor.

****

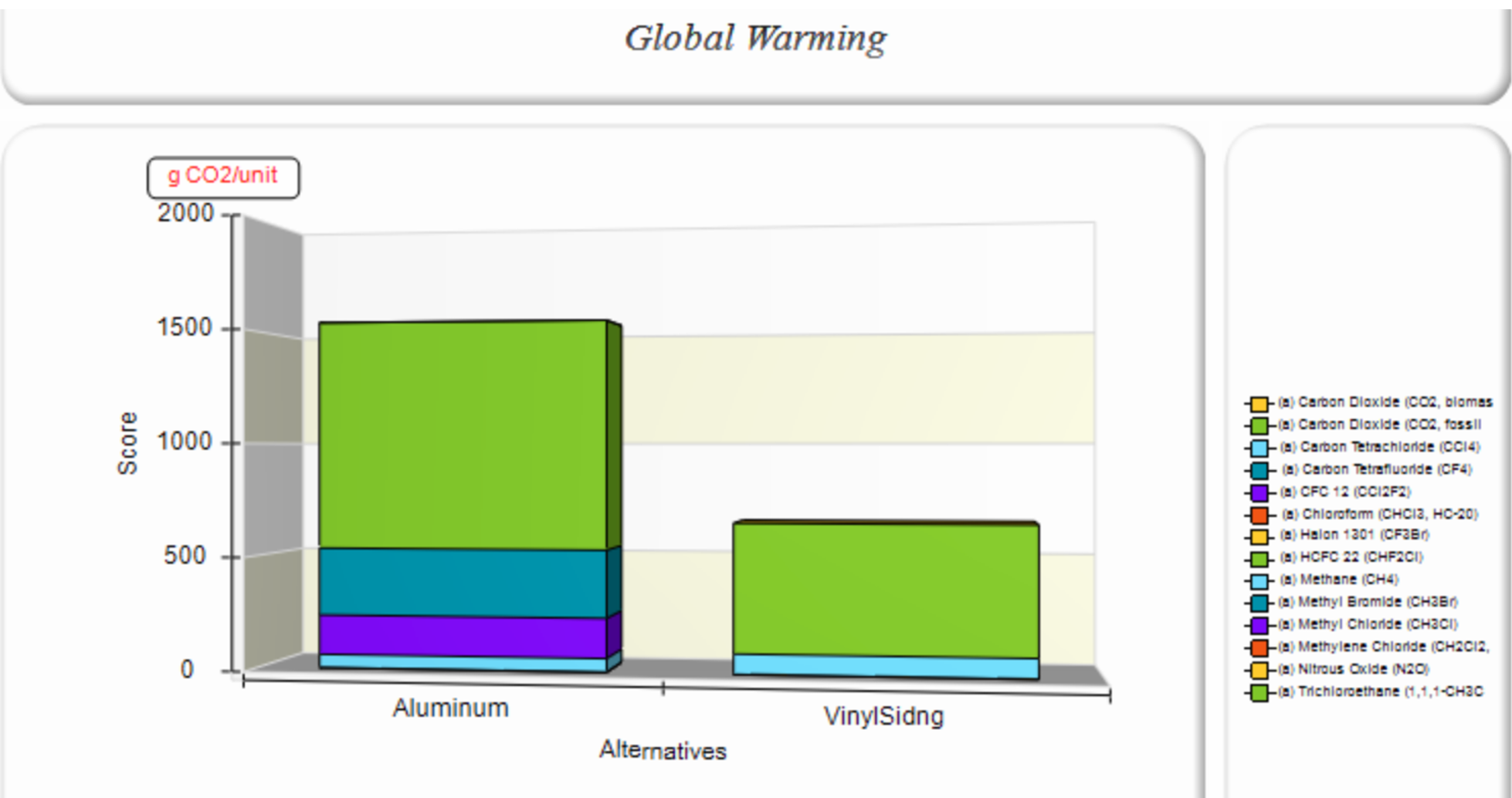
**Figure 7: GUI Connections to Camera-As-A-Sensor and Motor**

**Life Cycle Analysis:**

****

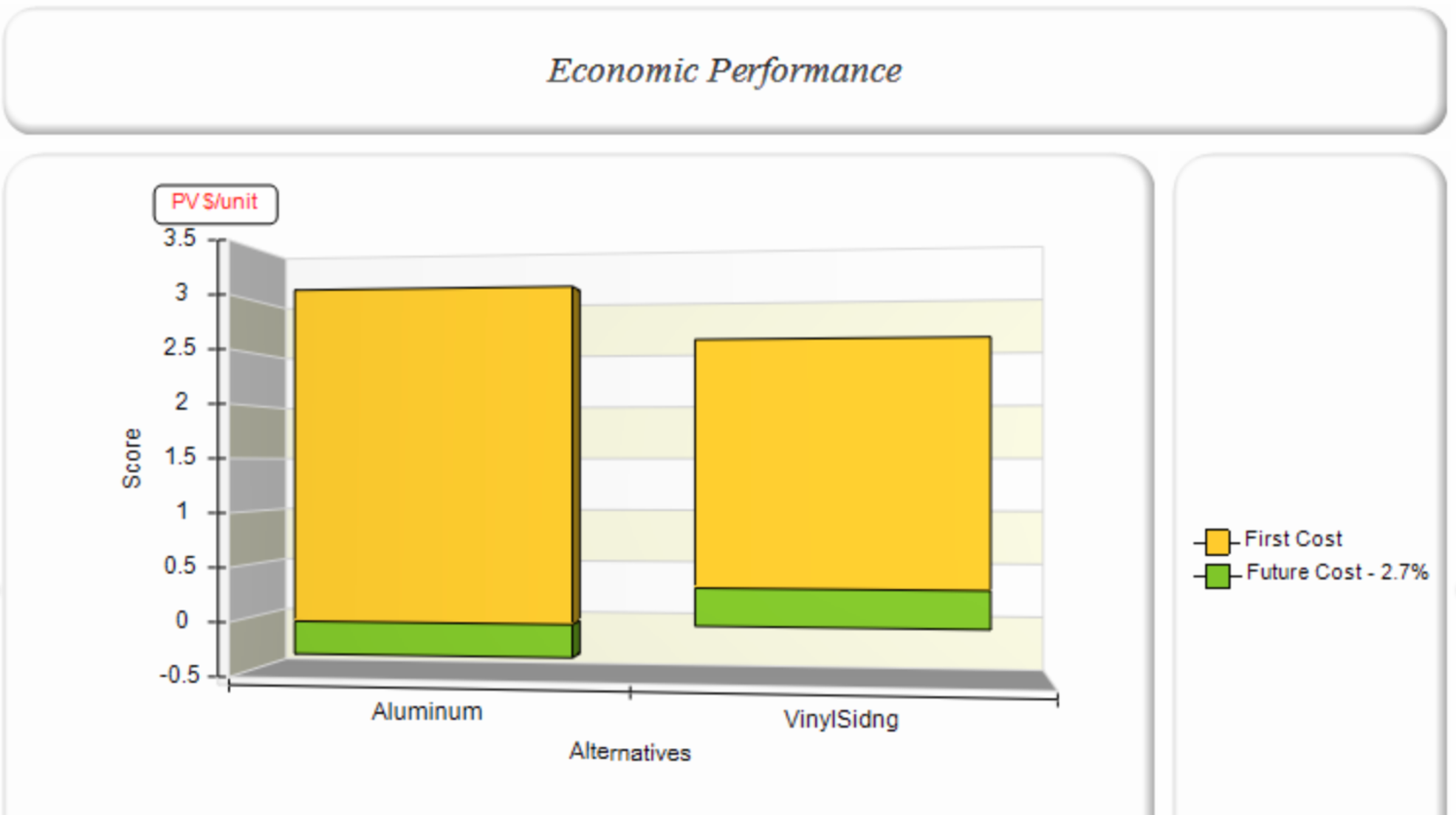
**Figure 8: Environmental Performance**

In terms of environmental performance, as seen above in **Figure 8**, we feel that vinyl siding is better because overall it contains less harmful toxins that affect the users or environment. Also, the vinyl siding doesn’t contain any materials that could potentially cause cancer. In this case, vinyl siding is the less harmful choice.



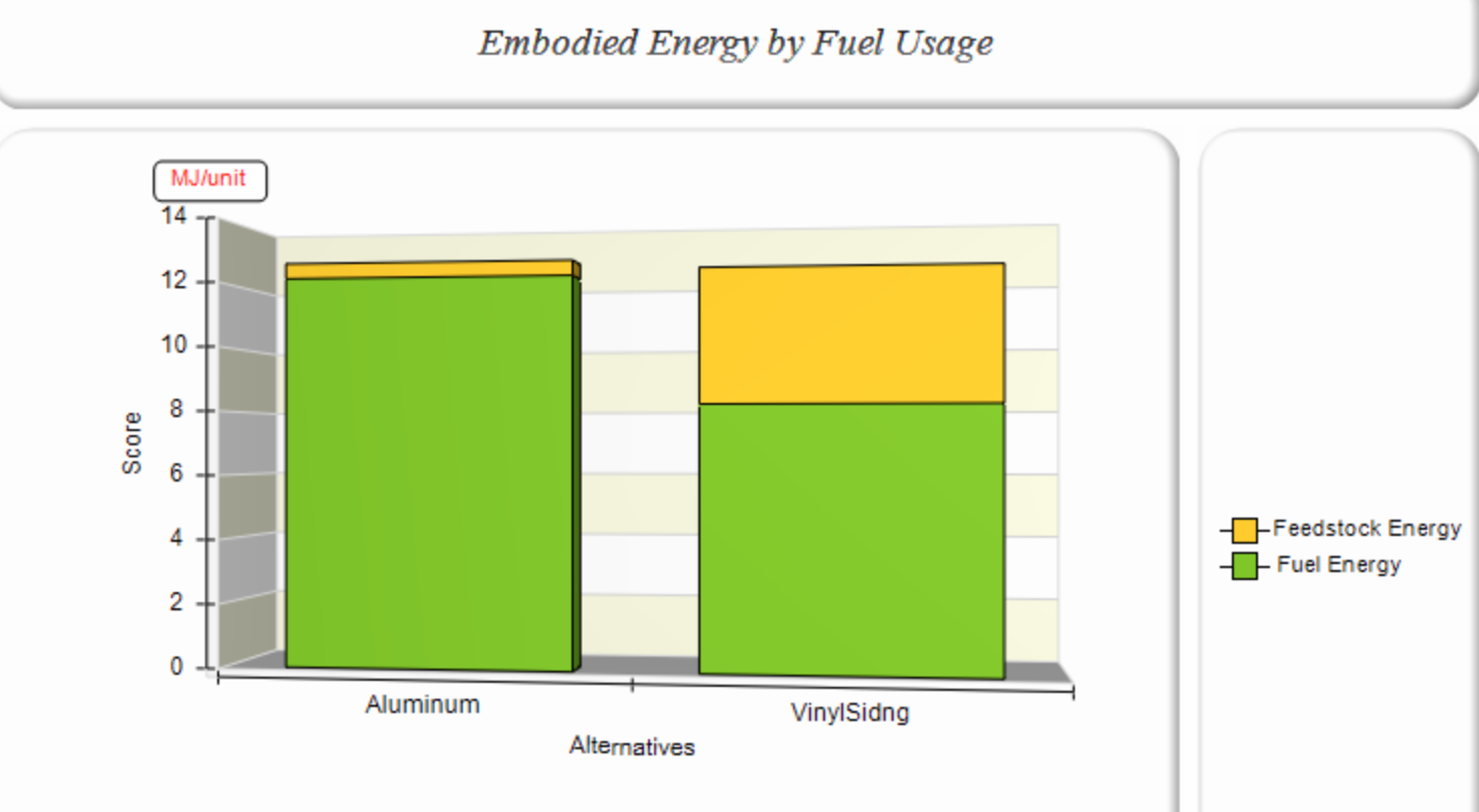
**Figure 9: Global Warming**

In terms of global warming as seen above in **Figure 9**, vinyl siding is the correct choice here as well. Vinyl siding produces less harmful pollutants to the air and thus is better for global warming. Vinyl siding produces about 1000 g of CO2/unit less than that of aluminum siding.



**Figure 10: Economic Performance**

When comparing aluminum siding and vinyl siding, the environmental impact, economic factors, and global warming are consideration factors. In terms of environmental impact, as seen in **Figure 8**, the vinyl siding has a smaller impact with 0.0008 pls/unit difference than that of the aluminum siding. We believe that the vinyl siding is better for the environment because it has an overall smaller impact and doesn’t include any materials that could potentially cause cancer. In terms of economic factors, as seen in **Figure 10**, the vinyl siding comes to be less expensive than aluminum siding by 0.19 PV $/unit. And lastly, In **Figure 9**, the global warming impact is less by about 1000 g CO2/unit for the vinyl siding than aluminum siding, thus making it a better choice. Overall, we feel that vinyl siding is the better choice for our robot case due to it costing less, having a better environmental impact, and less global warming impact.

.

**Figure 11: Embodied Energy by Fuel Usage**

As seen above in **Figure 11**, in terms of embodied energies by fuel usage, either siding takes about the same amount of energy to create. For your choice, it depends on what is available to you. If you have more access to feedstock, which is a raw material to make things with, then vinyl siding might be the way you could go, but if you have less access to feedstock, then aluminum might be the way to go. Additionally, all of these tests can be performed on the previous two projects and the final project to reduce the environmental impact.

**Employee Training Program:**

As a startup company, it is important to consider and budget for events that will help advance the lifelong learning of the engineers working there. We have outlined two different employee training programs in **Table 2a and 2b** below. One is designed for a newly graduated engineer who will solve technical problems and projects, while the other is designed for a newly graduated engineer who will manage technical projects.

**Table 2a: Employee Training Program - Solving**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Benefit** | **Cost** |
| Pay IEEE Membership | Exposure to updates in technology. | Membership = $400 |
| Attend HRI ‘21: ACM/IEEE International Conference on Human-Robot Interaction  (Mar 8-11; Virtual) | Gain exposure to and knowledge of the newest advancements in robotics, AI, cognitive science and other fields relating to HRI and address the related problems. | Registration/fees = $620  3 Days of Work = $634.62  Total = $1,254.62 |
| Attend 2021 Annual Conference on Information Sciences and Systems  (Mar 24-26; Virtual) | Hear speakers present their latest results and developments in a wide range of areas pertaining to information sciences and systems. | Registration/fees = $30  4 Days of Work = $634.62  Total = $664.62 |
| Attend 2021 IEEE Conference on Software Testing, Verification and Validation  (Apr 12-16; Virtual) | Learn about advancements in software testing and hear keynote speakers present their latest research findings and developments. | Registration/fees = $250  3 Days of Work = $846.15  Total = $1,096.15 |

**Table 2b: Employee Training Program - Managing**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Benefit** | **Cost** |
| Pay IEEE Membership | Exposure to updates in technology. | Membership = $400 |
| Attend 2021 Annual Reliability and Maintainability Symposium (RAMS)  (Jan 25-28; Virtual) | Learn about new and up-and-coming advancements in management approaches, including how to make them cost-efficient and reliable. | Registration/fees by 12/14/20 = $995  4 Days of Work = $846.15  Total = $1,841.15 |
| Attend International Conference on Management, Economics and Industrial Engineering  (Aug 12-13; Virtual) | Learn more about new strategies in management and economics, while still gaining technical IE knowledge. | Registration/fees = $250  2 Days of Work = $423.08  Total = $637.08 |
| Attend Technological Systems Innovation, Entrepreneurship, and Strategic Management Conference  (Nov 26-27; Virtual) | Gain knowledge of advancements in strategic management and other fields that can provide technical insight into projects. | Registration/fees = $250  2 Days of Work = $423.08  Total = $673.08 |

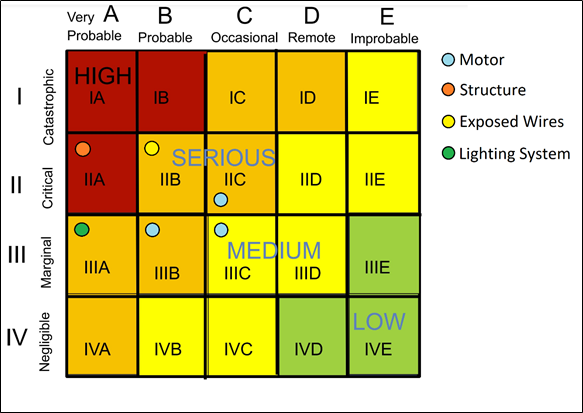
Since the United States is still being affected by the Coronavirus, many of the 2021 conferences have been moved to a virtual platform, saving the company on travel and lodging costs. The company should budget for between a week and two weeks worth of time per employee for the year so that they can attend conferences and other events that will further their lifelong learning. This will also encourage employees to want to remain at the company longer and prevent them from feeling as if they are stuck in a job with no room for learning and improvement.

**Safety Analysis:**

Once a formal design had been established and created, we then observed areas of the design that could be considered a hazard for individuals using/interacting with the device as well as pose a threat to the function of the system itself. To do this, we listed the possible hazards, the symptoms of those hazards, and the effects those hazards could have. We then ranked them based on the likelihood of their occurrence and the severity of the risk. The hazards that we observed and categorized are listed in the DFMEA table labeled as **Table 3** and the corresponding risk assessment matrix labeled **Figure 12** below.

**Table 3: Design Failure Mode and Effects Analysis Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Component** | **Possible Hazard** | **Symptom** | **Effect** | **Chance of Failure** | **Severity of Failure** | **Risk Index** |
| Motor system | Motor casing may be very hot | Air around motor may be warm | Represents dangerously hot surface | D | III | III-B |
|  | Motor speed set improperly | Motor arm moving at a high speed | May damage the system | C | II | II-C |
|  | Appendage could get caught in the motor Arm | Sticking an Appendage in between motor arm | Can cause injury to person | C | III | III-C |
| Structure | System structure may topple over | Undesired lateral motion | Heavy machine can cause injury to device or individuals | A | II | II-A |
| Exposed wires | Wires exposed out in the open can get snagged or broken | Wires may be frayed/disconnected from correct position | May represent a shocking hazard. Can damaged the system significantly | B | II | II-B |
| Lighting system Bulb | Light bulb likely to get hot | Air around bulb is warm | Represents dangerously hot surface | A | III | III-A |

****

**Figure 12: Risk Assessment Matrix**

Once we had established areas that possessed a hazard, we then set about rectifying and mitigating those hazards. To begin, we looked at the motor and its risk of getting hot. Though it is likely this would take a period of sustainable operation, it is possible and can pose a risk. To solve this, we 3d printed a motor housing, which prevents direct contact to the motor housing. The second motor hazard is that the motor could be set to a high speed. This posed the greatest risk to the machine and anyone surrounding it. In terms of the system, it could shake and break things due to its momentum. Additionally, the motor attachments could fly off and injure people. To solve this problem, we made sure to test the motor to make sure that the speed it ran at was safe. The final motor hazard is that it could be a limb hazard, where it’s possible to get a finger stuck in the machine. To rectify this, we added a PVC cage around the whole system that discourages the touching of the components.

When it comes to the structure, our system ended up being quite tall and could be a tipping hazard. To fix this, we simply gave it a very sturdy and heavy base. The next hazard discussed was the exposed wires. After compiling the design, we noticed our wires were exposed and messy. To fix this, we mounted everything inside of the base. Finally, to fix the hazard of people touching the hot bulb, we put a shade around the bulb. The shade allowed the bulb to emit enough light for the camera, but at the same time still prevent people from touching it. After making these adjustments, we can conclude that our system is safe.

**ECE 4950 Project 3 –Closed-Loop Motor Control, Life-Cycle Analysis, and Risk Assessment**

Use the guidelines below to complete your report and add at the end of your report.

Group Member Last Names: Aho, Anderson, Cuttino, Liggett, and Moran

|  |  |  |  |
| --- | --- | --- | --- |
| Score | Pts |  | Performance Indicators |
|  | 5 | **General Format - Professional Looking Document/Preparation (whole document)**   1. Fonts, margins (11pt, times new roman, single spaced. 1" margins on all sides). 2. Spelling and grammar are correct 3. Layout of pictures – all figures need numbers and captions and must be referenced in the text 4. Follows the page limitations below. 5. References. Use IEEE reference format. 6. This grading sheet is included as the final page. | g.1 |
|  | 5 | **Page 1: Title, Group Name, Group Members, and Date Executive Summary** (1 concise, well-written paragraph)  Provide an overview of this project. Briefly describe what you did and what you learned. | g.1 |
|  |  | **Subsystem Design** |  |
| 10 | **Page 2: Engineering Requirements for the Motor Control subsystem** (~1 page) | c.2 |
|  | Considering only the Stickerboard-Motor subsystem, make a three column table that lists the |  |
|  | Customer Requirements in the first column, the resulting Engineering Requirements in the |  |
|  | second column and the tests done to verify that the design chosen meets each requirement in |  |
|  | the third column. Note that one Customer Requirement could map to multiple Engineering |  |
|  | Requirements. |  |
| 5 | **Page 3: Overview of Hardware-in-the-Loop** (~1/2 page) | k.3 |
|  | Describe in your own words what Hardware-in-the-Loop means. What is the difference |  |
|  | between a full simulation and a Hardware-in-the-Loop simulation? What are the strengths of |  |
|  | HIL? |  |
| 10 | **Pages 3-4: Document Hardware** (1 page) | k.1 |
|  | Describe and show images of the equipment used, connection diagrams, calculation of |  |
|  | resolution – pixels per square inch/cm on game board etc. Is the camera an appropriate |  |
|  | sensor? |  |
| 10 | **Pages 5-7: Document Software** (3 pages) | k.2 |
|  | Using Flowcharts, state diagrams, data structures etc. describe how the software is |  |
|  | implemented. There is no need to include the source code. |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 10 | **Page 8: Document Simulated Motor Experiments** (~1 page)  Plot the effect of changing gains using the reference, actual position and error signals. What happens when the proportional, derivative and/or integral gains are changed? | b.2 |
| 10 | **Page 9: Document Motor Experiments** (~1 page)  Plot the effect of changing gains using the reference, actual position and error signals. What happens when the proportional, derivative and/or integral gains are changed? Describe the  relationship between these simulated experiments and the HIL experiments. | b.1  b.3 |
| 5 | **Pages 10-11: Document and Evaluate your User Interface** (~2 pages)  How does the user interface connect the Camera-As-A-Sensor and the Tohoku motor? What  information is provided to the user and why? Document using screenshots and similar images. | j.1 |
|  | 10 | **Pages 12-13 Life Cycle Analysis (2 pages)**  You are proposing a design that consumes resources. Follow the “Life Cycle Assessment  (LCA) Exercise” for the shipping box for your project to examine the life cycle for this one part of your design. Be sure to interpret the results of the computer program. Complete this section of the report by saying that a similar analysis could be done on the entire project to  reduce environmental impact. | c.3 |
|  | 10 | **Page 14: Employee Training Program** | i.1 |
|  | 15 | **Pages 15-16: Safety Analysis**  The project must be safe for use by the customer. Perform and Document a DFMEA for your project. Document your analysis using the DFMEA Table and Risk Assessment Matrix shown in the class slides. Show that you have implemented the results of the analysis to make your  design and workspace safe – that is document what changes you made to make your system and space safe as a result of the safety analysis. Can you conclude your system is safe? | c.2 |
|  | - | **Page 17: Grading Sheet** | g.1 |