Northeastern University

Final Tech Report

Gear Up

Sumo 2 Group 1

12/7/2023

GE1501 Cornerstone Engineering 1

Mark Frigiani, Caio DaSilva, Peter Hunt, Justin Ho

To: Professor Whalen

Subject: Cornerstone Sumo 2 Final Tech Report.

From: Gear Up (Mark Frigiani, Caio DeSilva, Peter Hunt, Justin Ho)

Date: 12/8/2023

The problem that our group attempted to solve was surrounding an autonomous sumo robot. The objective of our group was to design a fully autonomous robot that could detect other robots inside of a white ring on a black mat, and either push them out of the ring or render their robotics nonfunctional. Building upon Sumo 1, our group added side ramps to our robot, intending to disrupt opponents' movements. Additionally, a fork was added to the top of the Sumo 1 robot to disrupt opponents' wiring. Furthermore, coding was changed to implement a millis timer, allowing for more fluent movement of the robot.

Come competition day, we had implemented all of these new designs onto Gear Up. Some design features that were considered but not added to the robot include additional distance sensors, a third and fourth wheel, and a frontward-facing wedge. Although all of these would have been useful, a lack of time and money caused them not to be added.

Throughout the challenges of this assignment, members of our team have learned to work collaboratively to brainstorm while in a group environment. Additionally, we have all picked up the skill of using decision analysis tools, such as a decision matrix, to decide the best course of action for our project. A final important takeaway is that when completing design projects under the pressures of a real-world environment, time and money do play a huge factor. If our group had been given an extra two weeks, we would certainly have been able to add extra features, such as those listed above, that would give us a leg up in the competition.

All in all, the sumo design project was a fun and intellectually challenging assignment that pushed all members of Gear Up to innovate and problem-solve with the constraints of a realistic work environment.

Table of Contents

1. Figures, Tables, Definitions, and Abstract

1.0 Figures

1.1 Tables

1.2 Definitions

Arduino IDE – Coding language used for this assignment.
Autocad – Design software used to create models for 3D printed objects as well as all laser-cut items.
Autonomous – Fully operational without outside influence or control. Ie. no remote controls, human assistance, etc.
Breadboard – Extra panel for electrical connections to be made before jumping wires to the redboard.
Decision Matrix – Table used to analyze qualitatively and quantitatively which design should be used.

Distance Sensor – Sends a signal that bounces off of objects. The time it takes for the signal to return to the distance sensor is used to calculate the distance the nearest object is in the direct line of sight of the sensor.
Duncker Diagram – A diagram used to break down complex problems and decide which concrete, specific solutions should be used.
Redboard – Control panel of the robot. The mainframe for all of the electronics.
Servo – Motor used to spin objects. Can either rotate 360* (360* Continuous Servo) or 180* (micro-servo).

1.3 Abstract

The group designed a sumo robot that performed many basic functions and competed in a competition against other sumo robot designs. The design team utilized the Engineering Design Process to assess the problem, generate solutions, make decisions, and build and test a working robot. The process began with assessing the immediate needs and requirements for the sumo robot. The team formulated a problem statement which was used as guidance for what needed to be completed and what parameters and expectations were to be met. Upon the conclusion of identifying the problem, the team began the idea-generation process using brainstorming tools like the C-Sketch method to create possible ideas (*Figure 2*). The brainstorming methods proposed several possible additions to the robot. To select the ideas that would contribute to the robot most effectively, the team designed and implemented a Weighted Decision Matrix. The designed matrix (*Table 1*) prioritizes effectiveness when the idea is added to the robot, and evaluated based on criteria from the problem statement generated. The results of the decision matrix yielded the three solutions that would be used in unison on the sumo robot: a fork, side wedges, and a modified code utilizing the millis timer.

Upon generating the solutions to be implemented, the team created visual representations utilizing Computer Aided Design with the program AutoCAD (*Figure 8 and 9*). The resulting schematics were used for 3D fabrication utilizing a 3D printer for the side wedges, and a laser cutter for the fork. The newly fabricated components were fastened to the robot chassis creating a prototype of the final robot design. At this stage, the team wrote the program that would allow the robot to run autonomously. The program was generated in the Arduino IDE, a work site where the program could be uploaded directly to the robot's RedBoard (*Figures 5, 6, and 7*). The code implemented coding principles to power the robot's servos based on readings from the sensors. To test the success of the robot at this stage, several test strategies were used. A large variety of methods tested all of the components individually, as well as how they function when running at the same time as other components.

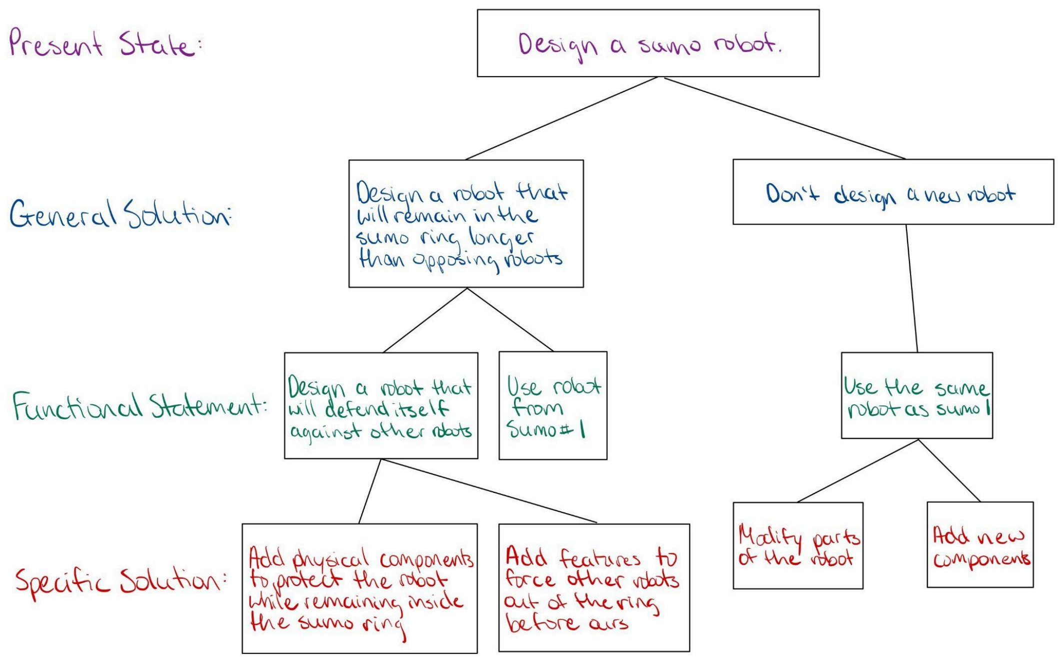
After completing several tests and ensuring all components were functioning as desired, the team prepared for the final competition and presentation. The final competition consisted of a "winner's bracket" and a "loser's bracket". In the first round of the competition, the robot was outperformed by the opponent and forced out of the ring, suffering a loss according to the rules of the competition (outlined in the appendix). After the first round, the robot was more

successful in the “loser’s bracket” where it was able to remain in the ring for longer than the opposing robot, success as defined by the generated problem statement. In the next round of the competition, the robot suffered another defeat and was eliminated from the competition. Upon conclusion, the team evaluated the performance of the solution and reflected on the overall process.

2. Implementation and Iteration

2.1.1 Duncker Diagram

Figure 1 – Duncker Diagram



2.1.2 Duncker Diagram Usage

The Duncker Diagram helped the team gain insight into the root problem that needs to be addressed. In the problem statement that we were given in preparation for Sumo 2, we were told that we need to prepare a sumo robot that fits under certain conditions and will defeat another robot in a sumo competition. Using the Duncker Diagram has helped us determine what needs to be done and put the scope of the problem into something realistic and achievable.

2.2 Rank-Order Objectives

1. A fork that moves up and down to tangle opponents’ wires.
2. Robot moves continuously without any delays.

3. Robot operates a fork when a robot is located within two feet of the front sensor.

2.3 Constraints

Designed robots must fit within a bow of width and length of 16 inches and 12 inches respectively. The robot can be any height desired. The robot must remain no heavier than 3.5 pounds.

2.4 Functions

1. Design a robot that fits all design constraints.
   a. This will be a success if the robot is smaller than a 16 x 12 inch rectangle that is less than 3.5 pounds.
2. Design a robot that will operate autonomously.
   a. This will be a success if the robot drives and moves on its own once a member connects the robot to a power source.
3. Design a robot that will remain within the boundaries of the sumo ring.
   a. This will be a success if during the competition the robot identifies the white border of the ring and moves away from the ring on its own. This will not be a failure if another robot pushes it out of the ring.
4. Design a robot that will identify another robot within 2 feet of the front of our robot.
   a. This will be a success if the robot drives at a robot if it is within two feet of the front of the robot and the front distance sensor.
5. Design a robot that will defend itself against other robots during a sumo match.
   a. This will be a success if the robot prevents other robots from limiting the functionality of our robot.
6. Design a robot that will force other robots out of the ring prior to forcing itself out of the ring.
   a. This will be a success if the robot pushes another ring out of the ring during the competition.

2.5 Problem Statement

The problem that the group is looking to address is how to design and build a robot that remains within the circular boundaries of a sumo mat for a longer period than an opposing robot. Our solution will be successful if it: operates autonomously, detects the marked boundaries of the sumo ring, remains within the marked boundaries of the sumo ring, and detects an opposing robot. The solution must fit within a 12 inch by 16 inch space and not exceed 3.5 pounds.

2.6 Background and Technical Research

[1] This article describes the complex strategies used in a Japanese sumo robot league as well as the different types of design. In this league, some of the robots can seem to weigh up to hundreds

of pounds due to the strong magnets holding the robots to the metal floor. It is common for robots in these leagues to use frontwards facing wedges to disrupt the traction of opponents. Additionally, some robots use a flag-waving strategy to disrupt the opponents’ robot sensors.
[2] This article discusses a Japanese sumo robot league and some of the common strategies that are used in competitions. Some of these existing solutions include spinning robots, moving the robot around the ring in a star shape, using wedges to decrease the opponents' traction, and using wings. Some of these ideas and strategies could be implemented into the team's final design.
[3] This article describes the common parts of the sumo design that are easy to do but widely overlooked. Examples of this include building the robot to the maximum weight, incorporating many light sensors to stay in the ring, and an aggressive coding strategy. A large emphasis of this article is building a skirt or shield on your robot in order to disrupt the movement or traction of your opponents by lifting them off the ground.

3. Generate and Decide

3.1 Brainstorming

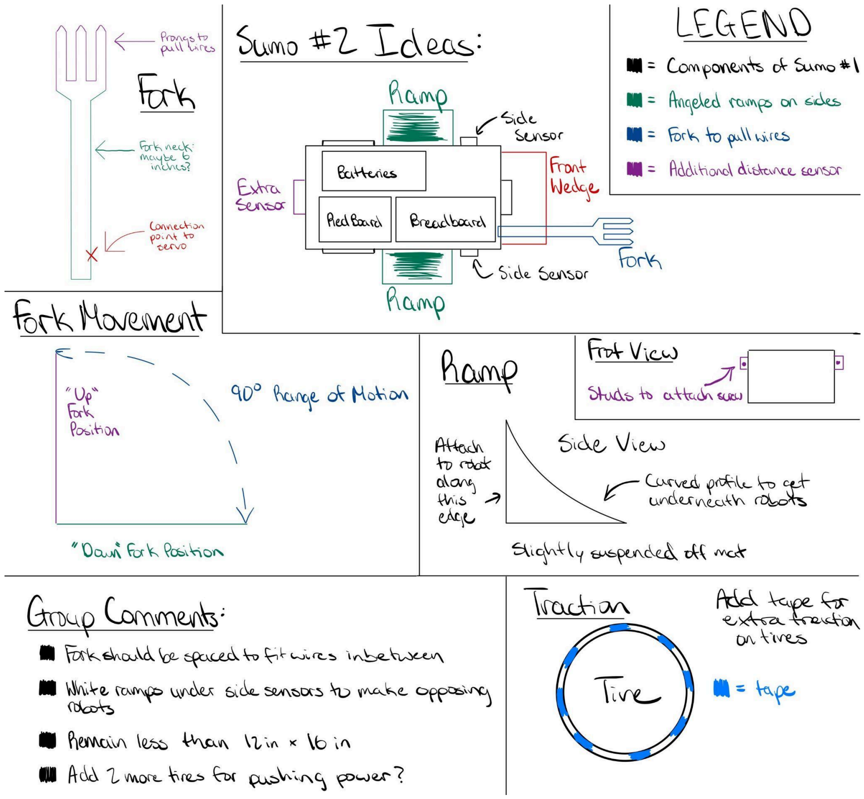
The team first used a traditional brainstorming element where we reflected on past failures and past areas of success. The group shared ideas and concluded with the following list of ideas:

1. 4 motors for more power
2. Fork on the front to disconnect wires
3. 4 distance sensors: left, right, front, back
4. Front wedge

3.2 C-Sketch

The team used a collaborative sketch method to expand upon the ideas that were developed. In the upper left-hand corner of Figure 2, the group designed and laid out how the fork might look. Located below that portion is the drawing of ideas for how the fork will rotate up and down, likely using a 180-degree servo. In the top right is the robot design with the new ideas added on and separated by color. Below is the design for the side ramps with a front and side view. In the bottom left corner, group comments on the ideas and new ideas were written. In the bottom right corner is an idea to remedy traction issues for the robot.

Figure 2 – C-Sketch



3.2 Evaluating Design Alternatives

To decide which ideas to implement, a design matrix was utilized. The team assigned weights of 7, 10, and 5, to the categories of Ease of Implementation, Effectiveness, and Time Consuming respectively. The team prioritized the ideas which would contribute most to the success of the robot. The least prioritized category was time-consuming as there would be enough time to implement each idea. Upon ranking each idea, the top 3 ideas to include were the fork, millis timer, and side wedges.

Table 1 – Decision Matrix

	Ease of Implementation	Effectiveness	Time Consuming	Total
Weight	7	10	5	
4 Motors For Power	4	4	4	88
Fork	6	10	8	182
4 Distance Sensors	2	2	2	44
Wedge	8	6	10	166
Millis Timer	10	8	6	180

4. Implementation and Iteration

4.1 Iterative Design Process

The development of the sumo robot involved an iterative design process, where various versions of the robot were conceptualized, prototyped, and tested. This section outlines the key prototypes, the design concepts they were meant to explore, and the insights gained from each iteration. In essence, this section documents how the robot was truly designed to function.

4.2 Prototypes

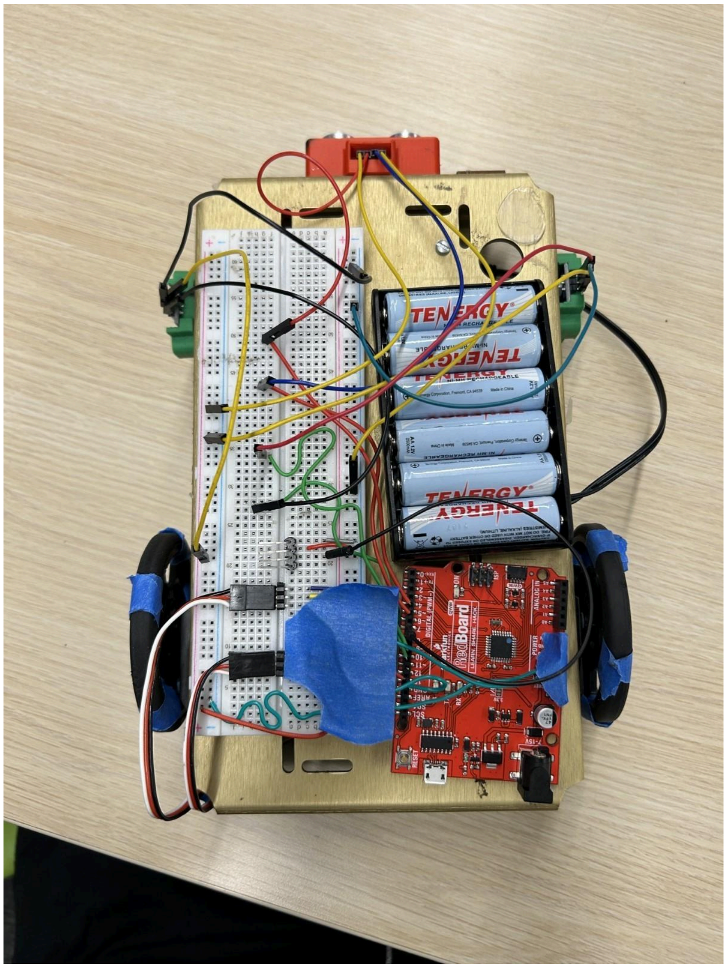
4.2.1 Prototype 1: Initial Concept Exploration

The first prototype focused on the base we already developed from sumo one. It incorporated the same chassis design, same motors, and same basic sensors. However, testing revealed issues with the wires remaining in place, a lack of uber-aggressive strategy, and problematic code delays.

4.2.2 Prototype 2: Fixed Wires

In response to the limitations from sumo one where our wire fell out mid-tournament, the second version prioritized a better mechanism to hold the wires in place. This involved going to FYELIC to upgrade the types of wires. Flat wires were the logical choice because it made them less likely to be pulled out accidentally. Flat wires also made the board easier to read and change throughout the competition. We also added tape over all the “important” wires—the ones that connected directly to the redboard.

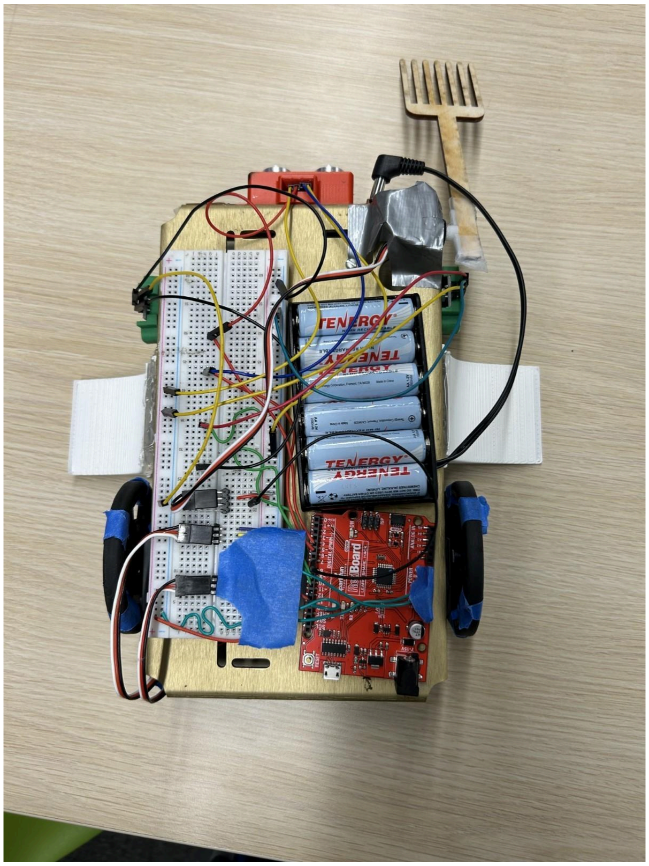
Figure 3 – Prototype 2



4.2.3 Prototype 3: Increased Offensive Strategy

The third prototype addressed the lack of offensive strategy observed in the previous versions. A more aggressive approach was designed using "The Fork" and side wedges. The Fork was created in AutoCAD and laser cut using 3mm wood. The goal of the fork was to catch other robot's wires with its prongs, ripping them from the board. This iteration also marked the inclusion of white side wedges meant to lift opponents and mess with their side sensors.

Figure 4 – Prototype 3



4.2.3 Prototype 3: Code Delays

Our original code for Sumo Competition 1 involved many delays. These delays were necessary for our robot to stop and change directions. However, these delays proved to be a detriment to the functionality of both the distance sensor and the side sensors. If an opposing robot was in front of the distance sensor, or the side sensors were above the edge of the ring during a delay, there would be no output, thus making the robot inconsistent. We integrated the millis() function to update the time based on how long the program has been run and created individual timers for each system on our robot: changing direction and the fork movement.

Figure 5 – Code Delays Screenshot 1

```
if (fork_down) { //when the fork is down
  last_fork_read_time = millis(); //update timer for when fork is down
  fork_can_down = false;
  servo_fork.write(25); //turn servo down
  if (last_fork_read_time - start_fork_time > fork_interval) { //when down is long enough
    start_fork_time = last_fork_read_time; // reset timer
    start_fork_hold_time = millis(); //start holding up timer
    fork_down = false;
    servo_fork.write(130); //bring fork up
    //start fork hold timer
  }
} else {
  last_fork_hold_read_time = millis(); //update timer for when fork is held up
  if (last_fork_hold_read_time - start_fork_hold_time > fork_hold_interval) { //when up is long enough
    fork_can_down = true;
    start_fork_hold_time = last_fork_hold_read_time;
  }
}
if (fork_can_down && !fork_down && getDistance() < fork_dist) { //fork can go down when its within distance and after its been up for some interval
  start_fork_time = millis();
  fork_down = true;
}
```

For example, the code above controls the fork movement. When the sensor is within a certain distance, the fork is not down, and there is no hold on the fork going down, the servo will turn to bring the fork down. This will start a timer, which when it ends, the servo will turn to bring the fork up, and a timer for the hold on the fork will begin. Once a certain time has passed, the fork is ready to go down again if it fits the conditions.

Figure 6 – Code Delays Screenshot 2

```
if (is_backing_up) {
  driveBackward();
  last_back_up_read_time = millis();
  if (last_back_up_read_time - start_back_up_time > back_up_interval) {
    start_back_up_time = last_back_up_read_time;
    is_backing_up = false;
    stopDriving();
    delay(100);
    start_turn_time = millis();
    if (turn_direction == 1) {
      is_turning_left = true;
    }
    else if (turn_direction == 2) {
      is_turning_right = true;
    }
  }
}
```

When any one of the side sensors is activated, “is_backing_up” will be true, and the code above will proceed. The robot will move backward until a certain time interval is met, and will then turn away from the line depending on which light sensor, the left or the right, is activated, as indicated by the “turning_direction” value.

Figure 7 – Code Delays Screenshot 3

```
if (is_turning_left) {
  turnLeft();
  last_turn_read_time = millis();
  if (last_turn_read_time - start_turn_time > turn_interval) {
    start_turn_time = last_turn_read_time;
    is_turning_left = false;
    stopDriving();
    delay(100);
  }
}
```

In the instance where the left sensor is activated, the "turning_direction" would equal 1, so "is_turning_left" will be true. This starts a timer to turn the robot to the left for a certain amount of time. Once the time interval is over, the robot will stop turning and return to its original state. The logic for turning right when the right sensor is activated is the same.

4.3 Concept Validation

The prototyping phase allowed for the validation of key design concepts:

- **Fixed Wires**: Flat wire with tape attached significantly improved the robot's design, keeping the wires in the robot throughout all testing.
- **The Fork**: The efficacy of the fork was proven in "competition" with one of our two original sumo one designs, CP. The Fork was able to stick its prongs under any non-flat wires and extirpate them from the robot.
- **The Wedge:** The wedge did not achieve the goals we set for it. It was not able to lift other robots that rolled onto it, and it never messed with any side sensors. However, the design included a happy accident. Whenever an opponent tried to push our robot out of the ring from the side, our wedges would dig into the mat making it difficult for us to be moved.
- **Code**: The code worked how we intended. The robot was moving fluidly, without any sudden stops when moving forward or spinning. Additionally, the sensors will be working very consistently. During the competition, if the opposition got within range, they were detected every time and our robot charged forward at them. Although our side sensor logic was not fully displayed during the competition, during our final testing just minutes before the competition, our robot showed the ability to stay within the ring by backing up and turning away from the circle.

4.4 Make or Buy Decisions

The decision-making process for components involves a cost-benefit analysis. Custom components, including The Fork and white wedges, were fabricated in-house using laser cutting and 3D printing to meet specific design requirements. We sought to spend the least amount of money possible on the robot, yielding the results below in the Bill of Materials.

4.5 Bill of Materials (BOM)

The Bill of Materials provides a comprehensive list of components, materials, and associated costs for the final design. This document serves as a reference for procurement and future maintenance.

Table 2 – Bill of Materials

Component	Quantity	Unit Cost	Total Cost
Wood For Fork	1	$5	$5
Total Cost			**$5**

4.6 Drawings and Schematics

Detailed photos and schematics outlining the final design, including the chassis structure, component placements, and wiring diagrams, are provided in the following section "Final Design Implementation and Performance Evaluation". The AutoCAD design of The Fork as well as a 3D model of the white wedges are provided below.

Figure 8 – Side Ramp Schematics

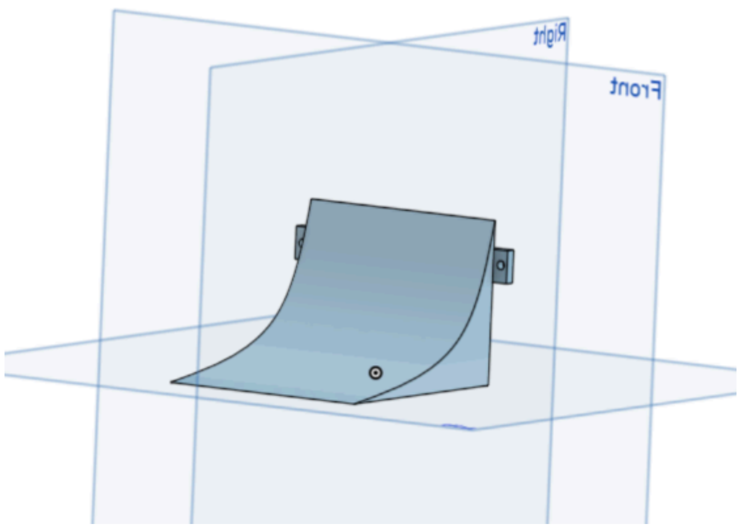
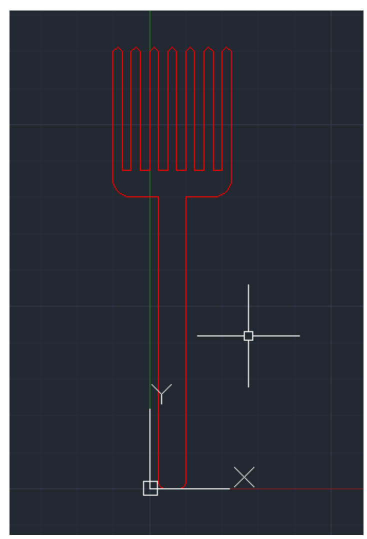


Figure 9 – Fork Schematics



5. Final Design Implementation and Performance Evaluation

5.1 Accomplishment of Design Goals

The final iteration of the sumo robot successfully realized the design goals established during the conceptualization phase. However, it did not accomplish our top outcome goal in winning matches. The integration of The Fork, white wedges, Millis timer, and fixed wires yielded a robot capable of effectively navigating the sumo ring and engaging in dynamic matches—until it came against a superior robot. Other robots were the issue with our design. All our code and designs functioned properly but they were not effective in immobilizing our competitors.

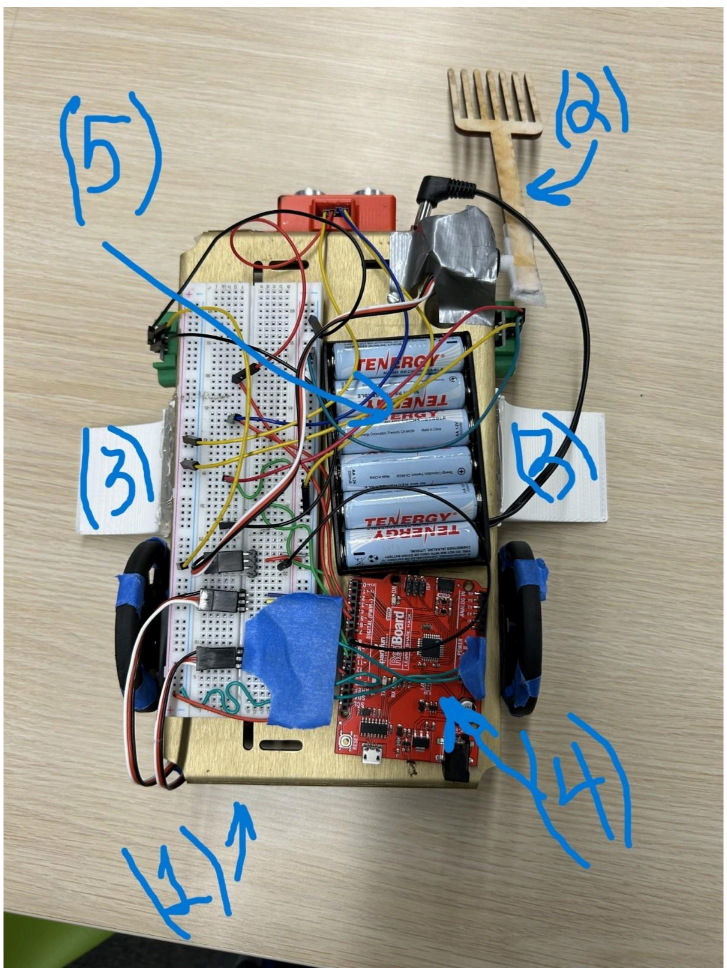
5.2 Design Reflection

5.2.1 Actual Design vs. Initial Prototype

A comparative analysis between the actual design and the initial prototype reveals significant improvements. The incorporation of The Fork, white wedges, Millis timer, and fixed wires addressed the limitations observed in sumo one. The final design not only met our goals for sumo two, but in some areas exceeded those expectations.

5.3 Visual Representation

Figure 10 – Final Design Prototype



Labels:

- (1) Chassis: Durable, metal construction ensures stability and a firm base to push robots out of the circle.

- (2) The Fork: Run by a 180-degree servo to rip out an opposing robot’s wires.
- (3) White Wedges: Placed on the side to lift other robots. It also is white so any side sensor that views it will cause the robot to back away from us.
- (4) Red Board: Enables the robot to run off autonomous code.
- (5) Battery Pack: Power source.

5.4 Performance Evaluation

5.4.1 Chassis

The chassis worked perfectly during the competition. It was big enough to hold all our components together without exceeding any of the constraints for the competition. If we were able to do further iterations, we would have liked to make the chassis heavier. A heavier chassis would have prevented robots from being able to push us out of the circle. If we made the chassis heavier, we also would have attached more powerful motors to maintain mobility from our own robot.

5.4.2 The Fork

The Fork did not work. Most of the robots we encountered had manners to protect their wires. This thwarted the fork and reduced it to a mechanism that could solely slap our opponents. The code functioned well for the fork, but in practice it was useless.

5.4.3 White Wedges

The white wedges also did not work in the way we intended. Robots did not drive up onto them and became unsteady. We also noticed that there were no circumstances in our three battles where the opponent's side sensors read the white as the sideline. However, we are still confident that this coloring was the correct choice. We believe that this did not happen because of the small sample size. Given all the failures of the white wedges, we were pleased with an unexpected happy accident derived from them. Any time another robot would push Gear Up from the side, the white wedges dug into the mat, preventing it from sliding towards the edge easily.

5.4.2 Code

Of all the different aspects of Sumo two, the improvements made to the code were the most successful. Everything about them worked exactly as we planned it. Furthermore, the Millis timer significantly cut down the amount of time it took to find the other robot in the ring. The code was a massive success.

5.4.2 Battery Pack

The battery pack worked perfectly. The batteries were only replaced once throughout all iterations of Gear Up. We felt no need for more power at any point throughout the design process.

5.6 Conclusion of Evaluation

Throughout the design and testing phases, iterative improvements were made based on performance evaluations. These iterations led to refined algorithms, improved hardware, and enhanced programming for better outcomes during matches. The final design of the sumo robot successfully accomplished the set design goals. The visual representations, including drawings and labeled images, provide a clear understanding of the design's key components and their interactions. This section serves as a comprehensive reflection on the success of the final product.

6. Reflection

6.1.1 What We Learned, And What We Could Have Done Better

Reflecting upon the design process, the fatal flaw of our group was that we operated under the assumption that other groups would not be making significant changes to their robot. For this reason, most of our design was created to directly combat the robots from sumo 1. Specifically, the fork that our group implemented to remove the wires from our opponent's breadboards was targeted at designs from Sumo 1. After competing in sumo 2, we quickly realized that the fork would never have been successful, as most groups had added a protective box to their wires within the new design. For this reason, we learned the necessity of testing against your competition, rather than outdated models, to gain a full understanding of how the robotics will operate and in what categories it will be successful come competition day.

Given the opportunity to go through this design process again, our group would have dedicated more time to creating a more effective robot at its core, rather than worrying about small additions before doing this. The first move here would be to scrap the fork, as it added little to no value to our design. Furthermore, we would focus our efforts on developing a wider and heavier chassis, to prevent us from being pushed around, as well as adding two more wheels, which would allow for more pushing power for our robot. Additionally, we now see the importance of adding more distance sensors on the sides of the robot, as this would cut down our turning time by up to 75%. Finally, we noticed that an additional augmentation to our side 11

Given unlimited time and money, it is likely that we would build a replica bot alongside our competition robot to allow for testing against a more updated opponent. Testing CP and Gear Up against each other is what allowed us to develop a robot that was of the highest competency, and the lack of an opponent for testing in Sumo 2 is likely what led to our downfall. Additionally, with access to unlimited funds, we would likely have purchased stronger motors and wheels to give our robot more pushing power, alongside a stronger more lasting power source.

6.1.2 Ethical Considerations

Regarding Gear Up, there are no substantial risks of it hurting someone. Other than the slim possibility of the fork being projected off of the robot and hitting someone in the eye, it is unlikely that Gear Up would cause damage to another human being. For the environment, however, Gear Up is run with battery power, so this could be harmful. Nevertheless, all batteries used in this competition were rechargeable, so our group did not create any nonbiodegradable waste throughout our project. To the knowledge of our group, no engineering codes were violated, or ethical issues surrounding the creation of our robotics. Although this is true, our robot does not contribute anything to society either. Although it could be augmented to provide some sort of service, possibly similar to a Roomba, this model provides no services to society.

7. References

[1] D. P. E. Team, “All About The Sumo Robot Competition And Technology,” *Device Plus*, May 26, 2016. https://www.deviceplus.com/trending/all-about-the-sumo-robot-competition-and-technology/

[2] J. Vincent, “You have to watch these freakishly fast sumo wrestling bots,” The Verge, Jun. 21, 2017. https://www.theverge.com/tldr/2017/6/21/15845032/robot-sumo-wrestling-fast-furious

[3] Tycho. “Sumobot Tips and Tricks – Iron Reign Robotics –.” *Ironreignrobotics.org*, 12 Apr. 2016, ironreignrobotics.org/2016-04-12-sumo-tips/. Accessed 8 Dec. 2023.

8. Appendices

8.1.0 Rules and Regulations

Requirements for Robots

Article 1 [Robot Specifications]

1. A robot must fit within a square tube of the appropriate dimensions for the given class. A robot may expand in size after a match begins, but must not physically separate into pieces, and must remain a single centralized robot. Robots violating these restrictions shall lose the match. Screws, nuts, and other robot parts with a total mass of less than 5 grams falling off from a robot's body shall not cause the loss of match.
2.

Class	**Height**	**Width**	**Length**	**Weight**

Robot Sumo	unlimited	12.00 in	16.0 in	3.5 lb

3. The total mass of a robot at the start of a match must be under the designated weight for the given class.**Autonomous classes:** The robot must be autonomous. Any method of control may be used, as long as it is fully contained within the robot and receives no external signals or directions (human, machine, or otherwise). **Autonomous robot operation must begin automatically no less than five seconds after being started by the user. Robots starting before the five second mark forfeit that Yuhkoh point.**
3. The robot must have a name or number for registration purposes. Display this name or number on your robot to allow spectators and officials to identify your robot.

Article 2 [Robot Restrictions]

1. Jamming devices that would make a mess of the ring or other robots are not allowed.
2. Parts that could break or damage the ring are not allowed. Do not use parts that are intended to damage the opponents robot or it's operator. Normal pushes and bangs are not considered intent to damage.
3. Devices that can store liquid, powder, gas or other substances for throwing at the opponent are not allowed.
4. Any flaming devices are not allowed.
5. Devices that throw things at your opponent are not allowed.
6. Sticky substances to improve traction are not allowed. Tires and other components of the robot in contact with the ring must not be able to pick up and hold a standard 3"x5" index card for more than two seconds.
7. Devices to increase down force, such as a vacuum pump or magnets, are not allowed.
8. All edges, including but not limited to the front scoop, must not be sharp enough to scratch or damage the ring, other robots, or players. In general, edges with a radius of greater than .005", as would be obtained with a unsharpened .010" thick metal strip, should be ok. Judges or competition officials may require edges that they deem too sharp to be covered with a piece of tape.
9. Batteries –rechargeable batteries are available in the learning center to test with however alkaline batteries may be used in competition.

How to Carry Sumo Matches

Article 3 [How to Carry Sumo Matches]

1. One match shall consist of 3 rounds, within a total time of 3 minutes, unless extended by the judges. This may be shorten to a single round if class time is running short.
2. The team who wins two rounds or receives two "Yuhkoh" points first, within the time limit, shall win the match. A team receives a "Yuhkoh" point when they win a round. If the time limit is reached before one team can get two "Yuhkoh" points, and one of the teams has received one Yuhkoh point, the team with one Yuhkoh point shall win.
3. When the match is not won by either team within the time limit, an extended match may be fought, during which the team who receives the first Yuhkoh point shall win.

Alternatively, the winner/loser of the match may be decided by judges, by means of lots, or by a rematch.

4. One Yuhkoh point shall be given to the winner when the judges' decision was called for or lots were employed.

Start, Stop, Resume, End a Match

Article 4 [Start] Upon the judge's instructions, the two teams bow to each other in the outer ring, approach the ring, and place a robot within their half of the ring on or behind the Shikiri line. (A robot or a part of a robot may not be placed beyond the front edge of the Shikiri line toward the opponent. Note that is not required that a robot be placed directly behind the Shikiri line; it may be offset to the side, as long as it is behind an imaginary line collinear with the Shikiri line.) When the judge announces the start of the round, the teams start their robots.

Article 5 [Stop, Resume] The match stops and resumes when a judge announces so.

Article 6 [End] The match ends when the chief judge announces so. The two teams retrieve the robots from the ring area, and bow.

Time of Match

Article 7 [Time of Match] One Match will be fought for a total of 3 minutes, starting and ending upon the judge's command. The clock shall start ticking five seconds after the start is announced

Article 8 An extended match, if called for by the judge, shall last for a maximum of 3 minutes.

Article 9 The following are not included in the time of the Match:

1. The time elapsed after the judge announces Yuhkoh and before the match resumes. The standard delay before the match resumes shall be 30 seconds.
2. The time elapsed after a judge announces to stop the match and before the match resumes.

Yuhkoh

Article 10 [Yuhkoh] One Yuhkoh point shall be given when:

1. A team legally forces the body of the opposing robot out of the match. If at least 1/2 of the robot goes outside of the ring during the match then that team will lose the point if no robot is fully eliminated.
2. When judges' decision is called for to decide the winner, the following points will be taken into considerations:
3. Technical merits in movement and operation of a robot
4. Penalty points during the match
5. Attitude of the players during the match
6. The match shall be stopped and a rematch started under the following conditions:
7. The robots are entangled or orbiting each other with no perceivable progress for five seconds. If it is unclear whether progress is being made or not, the judge can extend the time limit for observable progress for up to 30 seconds.
8. Both robots move, without making progress, or stop (at the exact same time) and stay stopped for five seconds without touching each other. However, if one robot stops it's

movement first, after five seconds it will be declared as not having the will to fight. In this case the opponent shall receive a Yuhkoh, even if the opponent also stops. If both robots are moving and it isn't clear if progress is being made or not, the judge can extend the time limit up to 30 seconds.

9. If both robots touch the outside of the ring at about the same time, and it can not be determined which touched first, a rematch is called.

Violations

Article 11 [Violations] Players performing any of the deeds described in Articles 6, 16, or 17, shall be declared in violation of these rules.

Article 12 [Insults] A player who utters insulting words to the opponent or to the judges or puts voice devices in a robot to utter insulting words or writes insulting words on the body of a robot, or performs any insulting action, is in violation of these rules.

Article 13 [Minor Violations] A minor violation is declared if a player:

1. Enters into the ring during the match, except when the player does so to take the robot out of the ring upon the judge's announcement of Yuhkoh or stopping the match. To enter into the ring means:
2. A part of the player's body is in the ring, or
3. A player puts any mechanical kits into the ring to support his/her body.
4. Performs the following deeds:
5. Demand to stop the match without appropriate reasons.
6. Take more than 30 seconds before resuming the match, unless the judge announces a time extension.
7. Start operating the robot within five seconds after the chief judge announces the start of the match.
8. Does or says that which disgraces the fairness of the match.