PURDUE SIGBOTS

BLRS2 NOTEBOOK

OVER UNDER

AUGUST '23 – MAY '24

BLRS2 Skills Robots			
181	Task and Timeline Update: September	Design Cycle 1	9/4/2023
183	Planning: Robot Skills Challenge Strategies	Design Cycle 1	9/7/2023
188	Design Brief: 15" Iteration 1	Design Cycle 1	9/11/2023
190	Design Brief: 24" Iteration 1	Design Cycle 1	9/14/2023
192	Brainstorming and Decision Matrix: Chassis	Design Cycle 1	9/17/2023
200	Brainstorming and Decision Matrix: Endgame Hanging Mechanism	Design Cycle 1	9/19/2023
208	Brainstorming and Decision Matrix: 15" Triball Intake and Launcher	Design Cycle 1	9/25/2023
214	Planning: 15" Motor and Pneumatic Distribution	Design Cycle 1	9/30/2023
218	Task and Timeline Update: October	Design Cycle 1	10/1/2023
221	Prototyping and Testing: Winch Hang Mechanism	Design Cycle 1	10/2/2023
225	Brainstorming and Decision Matrix: 24" Triball Scoring Mechanism	Design Cycle 1	10/11/2023
230	Design and Building: 24" Chassis	Design Cycle 1	10/18/2023
236	Planning: 24" Motor and Pneumatic Distribution	Design Cycle 1	10/26/2023
240	Task and Timeline Update: November	Design Cycle 1	11/3/2023
242	Brainstorming and Decision Matrix: 24" Chassis Redesign	Design Cycle 1	11/9/2023
247	Design and Building: 15" Chassis	Design Cycle 1	11/16/2023
251	Design and Building: 24" Chassis V2	Design Cycle 1	11/20/2023
257	Testing: 15" Chassis	Design Cycle 1	11/28/2023
265	Task and Timeline Update: December	Design Cycle 1	12/4/2023
267	Task and Timeline Update: January	Design Cycle 1	1/2/2024
268	February Update: Development on Pause	Design Cycle 1	2/10/2024
BLRS2 Team and Project Management			
270	BLRS2 9/24 General Meeting	Design Cycle 1	9/24/2023
277	BLRS2 10/1 General Meeting	Design Cycle 1	10/1/2023
281	BLRS2 10/15 General Meeting	Design Cycle 1	10/15/2023
285	BLRS2 10/22 General Meeting	Design Cycle 1	10/22/2023
289	BLRS2 11/5 General Meeting	Design Cycle 1	11/5/2023
293	BLRS2 11/26 General Meeting	Design Cycle 1	11/26/2023
296	BLRS2 12/3 General Meeting	Design Cycle 1	12/3/2023
299	BLRS2 1/8 General Meeting	Design Cycle 1	1/8/2024
302	BLRS2 1/15 General Meeting	Design Cycle 2	1/15/2024
305	BLRS2 1/29 General Meeting	Design Cycle 2	1/29/2024
307	BLRS2 2/9 General Meeting	Design Cycle 3	2/9/2024

Team Organization and Structure

Date	@August 21, 2023
Category	BLRS2 Pre-Season
Authors	Ⓒ Conner Siebert Ⓙ Jacob Zawacki

Who We Are

We are team BLRS2, a Purdue University team competing in the VEXU Robotics Competition. As a large team boasting members with a large array of experience levels and skillsets, we aim to utilize our strengths to perform at the highest possible level.

Our mission as a team is to achieve overall excellence through effective communication, management, and creating a positive team atmosphere.

Our Goals

1. **Utilize more customized build methods**
    a. This will help our members learn more about engineering with various materials and methods going into professional fields
    b. In addition, unique building methods may give us a competitive edge with more efficient robots
2. **Plan an effective strategy for two robots**
    a. As VEXU teams field two robots that compete in matches and skills together, we have the ability to plan complete strategies for both robots
    b. Developing a solid strategy for both robots will leave less variables up to chance in match play, which could improve our performance
3. **Score in the top five in the world skills ranking by December 2023**
    a. On December 31, 2023, the top five teams in the world skills ranking will automatically qualify for the World Championship
    b. Qualifying through skills early in the season will allow us to focus more on developing our team without needing to worry about a qualification later on

4. **Win 90% or more of our qualification matches**
    a. Qualification rank correlates exactly to tournament seed, meaning that every qualification win gives us a higher chance to win the tournament
5. **Rank in the top five in our division at the World Championship**
    a. Ranking well in the qualification rounds at the World Championship will provide a strong pathway to progressing through the elimination round, improving our odds to win the World Championship

Our Structure

This season, our team will be organized with task-specific subteams, as well as a weekly overall team meeting. Each subteam, as managed by an elected leader, hosts meetings, plans projects, and carries out related tasks.

- BLRS2 Team General Meeting
    - This weekly meeting will bring together members across all subteams to discuss overall team matters
        - Each subteam will provide updates and progress from the prior week
    - **Overall team timelines and goals will be set in this meeting**
    - Any member is welcome to discuss any matter about the team
- Competition and Skills Mechanics
    - These meetings focus on the brainstorming, CAD, and construction of our team's robots
    - **Team members collaborate on a shared concept and goal, following the timeline set in the BLRS2 team general meeting**
    - Members of the mechanics team are assisted by strategy members in assessing the game, effective ways to score, and how other teams are developing match strategies

- Software
  - **This subteam is tasked with creating the drive controls, autonomous routes, and other projects**
  - Software members should work with the strategy to develop efficient routes for both match autonomous and programming skills
  - The software also works with mechanic members to learn the functions of the robots and to create drive controls to the driver's preference
  - Software members should communicate with mechanic members any issues that arise with the functionality during programming
- Strategy
  - **Strategy provides a backbone for all team decisions**
  - Strategy members assist in game analysis, match strategies, robot designs and goals, and autonomous routing and functions of the robots
  - This subteam also completes all documentation of our engineering design process

Team Members

Software	**Mechanics**	**Strategy**
Brandon Liu	Max Johnson	Conner Siebert
Stephen Honholt	Sean MacDonald	Jacob Zawacki
Mihir Laud	Connor Nguyen	Kathleen Lowe
Joey Krejcie	Thomas Cyze	Noah Domogalik
Nathan Smith	Dominic Holifield	Alex Lam
	Joe Lach	
	Wesley Hayes	

Our Engineering Design Process

Date	@August 28, 2023
Category	BLRS2 Pre-Season
Authors	Ⓐ Alex Lam Ⓒ Conner Siebert Ⓙ Jacob Zawacki

In using a defined Engineering Design Process through the course of the season, our team will establish a robust, consistent method to research and develop solutions for defined problems.

Step 1: Define the Problem

Definition

Understanding the problem at hand is essential towards building well-thought-out solutions.

Implementation

- Identify the problem using relevant descriptions and pictures
- Set goals and requirements for possible solutions
- Establish a timeline and tasks to ensure timely progress

Step 2: Research Solutions

Definition

By combining unique ideas with past knowledge, we can ensure that we consider all relevant solutions to the defined problem.

Implementation

- Brainstorm solutions to the defined problem
- Weight the benefits and drawbacks of each solution
- Collect testing data to prove the viability of solutions
- Consider and cite ideas from outside sources

Step 3: Choose the Best Solution

Definition

Make an informed decision using the benefits, drawbacks, and data collected when researching solutions.

Implementation

- Decision matrix to weigh the factors of each possible solution
- Break down solution into tasks to optimize workflow
- Gantt Chart to organize the timeline for completing tasks

Step 4: Prototype and Design

Definition

Take steps to ensure the chosen solution is completed in an effective, timely fashion.

Implementation

- Create sketches and CAD to plan the solution
- Record the steps taken and parts needed to build

Step 5: Test and Refine

Definition

Ensure the completed solution meets defined expectations, and take steps to correct if needed.

Implementation

- Define a set process for testing the solution
- Thoroughly test all aspects of the solution, recording the results
- Analyze the testing data and make adjustments as needed

Game Analysis

Date	@September 6, 2023
Category	BLRS2 Game Analysis Pre-Season
Authors	C Conner Siebert A Alex Lam K Kathleen Lowe J Jacob Zawacki
Design Cycle	Design Cycle 1

Game Overview

This year's game for the VEX Robotics Competition is Over Under.

Figure 1: The Three colors of Triballs, the Scoring Object in VRC Over Under

Game Objectives:

- Place Triballs into your alliance's goal
- Move Triballs into your alliance's portion of the field
- Prevent your opponent from scoring Triballs into their goal and portion of the field
- In the endgame, hang your robot on the alliance elevation bars

Field Layout

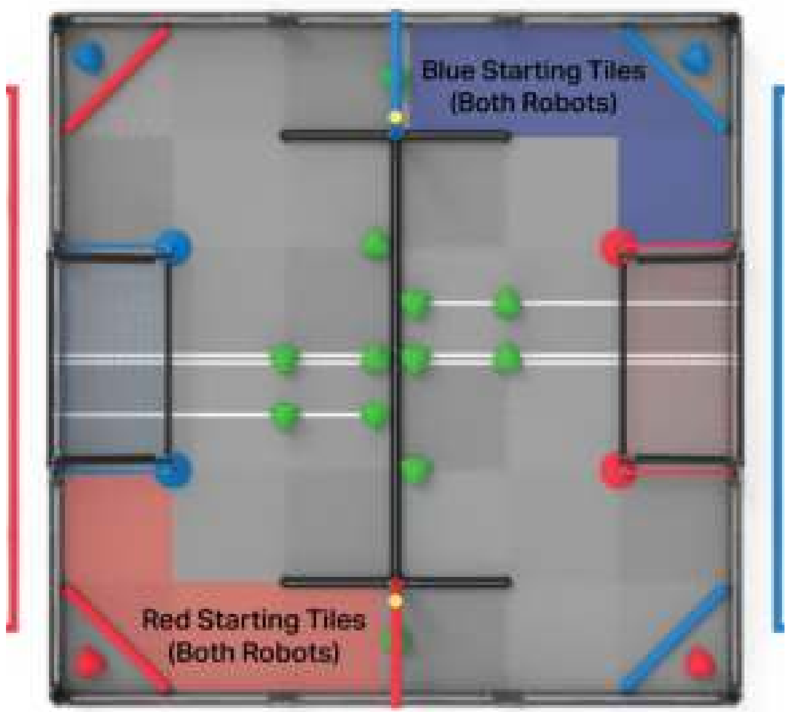


Figure 2: Vex U Field Starting Configuration with Robot Starting Positions

- In a VRC Over Under match, there are a total of 60 Triballs (Appendix C):
  - 16 Triballs that start on the field
    - As seen in Figure 2,
      - 10 Triballs begin scored in both alliance's offensive zone (evenly split 5 and 5)
      - 2 Triballs on the field that are not scored, located under the alliance hang bars
      - 4 alliance Triballs located in the match load zones
  - 44 match load Triballs
    - 22 per alliance
    - 10 per match may be introduced during the autonomous period for each alliance <VUG5>

- Elevation Bar
    - Comprised of a vertical and horizontal bar
    - Each alliance has their own elevation bar
- Goals
    - Located on opposite sides of the field in each alliance zone
- VEX Field
    - 12′ x 12′ field, with a metal or plastic border
    - 6 × 6 grid of 24″ foam field tiles make up the field surface

Game Object Analysis

Triballs, the main game objects, as seen in Figure 1, are the primary source of points in a VRC Over Under match. Therefore, we came to the following early season conclusions on the importance of the Triball:

- There is going to be a challenge associated with scoring Triballs into a goal due to the goals dimensions (see goal below)
- A Triball is manufactured out of a textured plastic that can still be slick when handled by hand
    - While match loading Triballs, the slick texture could be hard to grip, creating issues while attempting to load
    - **A grippy material will be required to handle Triballs, due to the slick nature of the game object**
- A Triball is 6.18″ in height at all times with an outer diameter of 7.00″ as seen in Figure 3
    - A Triball is a 4 sided object with sides resembling a triangle shape. This game object possess the unique geometry that remains the same height regardless of the orientation.

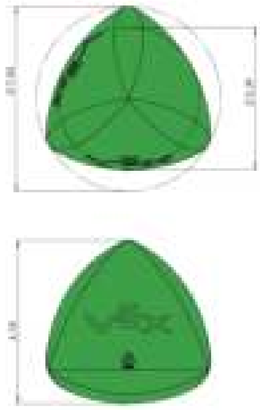


Figure 3: Dimensioned Triball

- The large size will limit robot design by
  - **The space required to intake a Triball into a robot (especially a 15″ robot)**
  - An intake would have to be even wider to be efficient at fielding a Triball by providing a larger margin of error for aligning the robot with the Triball

Figure 5: Goal Containing All Possible Triballs (58)

Goals are the key way to score Triballs in a Over Under match

- The height from the field tile to the bottom of the goal cross bars is 5.78″
  - The goal opening is 0.40″ smaller than a Triball is in height. **Therefore, a Triball cannot be lightly be put into a goal and will require force to be scored.**
    - Robots will have to push a Triball into the goal or strongly shoot Triballs into it

- The goal has a 39.37″ opening along the front and two 23.08″ openings along the sides
  - As seen in Figure 5, all 58 possible Triballs in the goal is nearing maximum capacity of the goal. **This means as more Triballs are scored, the harder they become to push under the bars of the goal.**

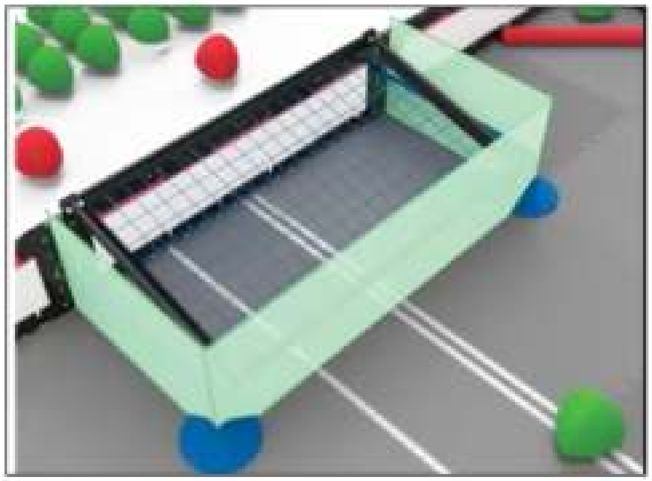


Figure 4: Alliance Goal with Planes Showing Scored Barrier

- For a Triball to be considered scored, **two of the four corners must break the plane of the goal**, as seen in Figure 4 above.
  - It will be important to make sure each Triball is pushed under the goal.
  - Making sure Triballs are completely scored is important to preventing the opponent from de-scoring your Triballs (outside of <SG8>)

Figure 6: Match load zones highlighted in the VRC Field

Figure 7: Match Load Zone Dimensions

Each Alliance has two match load zones, as seen in Figure 6.

- As stated above, each team is allotted 22 Triballs to be introduced during a match
  - **In autonomous, the 10 Triballs allowed to be introduced will play a key role in gaining the autonomous bonus**
- As seen in Figure 7 above, the match load zone is guarded by 2.37″ diameter pipe
  - This pipe poses a challenge while match loading, as as robot will have to lift a Triball over the pipe while removing it
- Figure 7 also displays how much larger the match load zone compared to a Triball
  - This poses a risk of Triballs getting wedged in the back of the match load zone
  - If a robot does not have the range to intake a Triball from the rear of the zone, it will be unable to utilize these Triballs

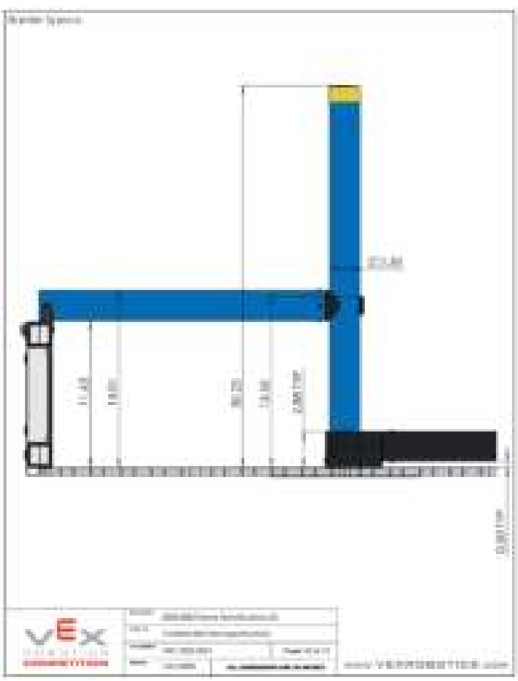


Figure 8: Elevation Bar Dimensioned

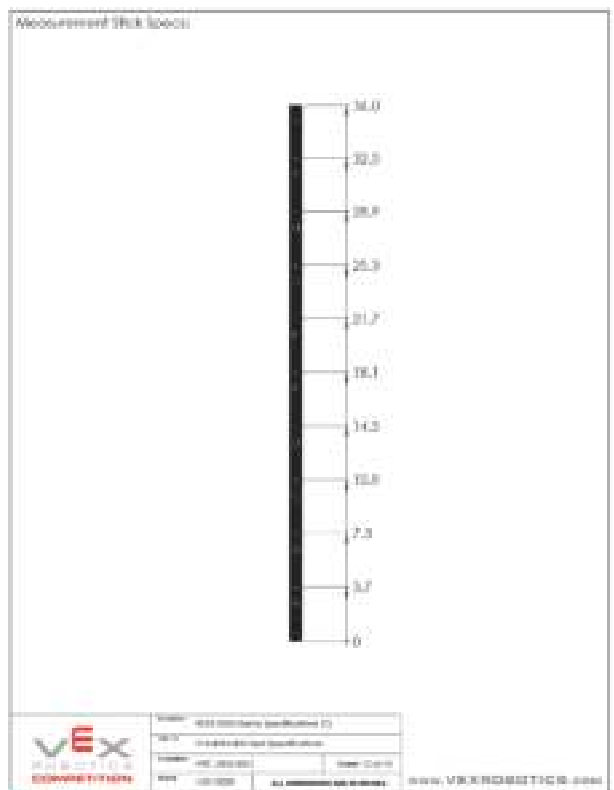


Figure 9: Measuring Stick Dimensioned

The elevation bars are composed of two components: a horizontal and vertical bar. These bars are used to attached your robot for endgame elevations.

- The elevation bars are composed of the same 2.38″ pipe, as seen in figure 8
    - To elevate your robot, it must be able to attach to the pipe by some sort of claw or grabber
- As seen in Figures 8 and 9, the highest scored zone is above the top of the elevation bar at a minimum height of 32.5″
    - **As the cap of the vertical elevation is only 30.23″ from the field, it will be necessary for a robot to lift a partner robot above its lowest point to reach elevation height J.**

Point Value Analysis

In a VRC Over Under match, points may be scored in the following ways:

- A Triball is placed in...
    - An Alliance Goal
    - An Alliance Zone
- In the endgame, a robot hangs on the elevation bars
- The alliance wins the autonomous bonus

POINT BREAKDOWN			
	POINT VALUE	TOTAL OBJECT AMOUNT	MAXIMUM POINTS POSSIBLE
TRIBALL IN ALLIANCE ZONE	2	56	112
TRIBALL IN ALLIANCE GOAL	5	56	280
ALLIANCE TRIBALL IN ALLIANCE ZONE	2	2	4
ALLIANCE TRIBALL IN ALLIANCE GOAL	5	2	10
ENDGAME HANG	20	2	40
AUTONOMOUS BONUS	8	1	8
HIGHEST TOTAL POSSIBLE			338

Table 1: Point Breakdown of Each Scoring Method in Over Under. Highest possible score assumes all Triballs are scored in the Alliance Goal

- As seen in Table 1, scoring Triballs in alliance goals are the highest proportion of total score in an Over Under match.
    - If an alliance scores all 58 Triballs, the maximum amount, in their goal, the 280 points is **82% of the 338 total possible score**.
        - Based on this, it should be a **priority for a team to score Triballs in an alliance goal**.

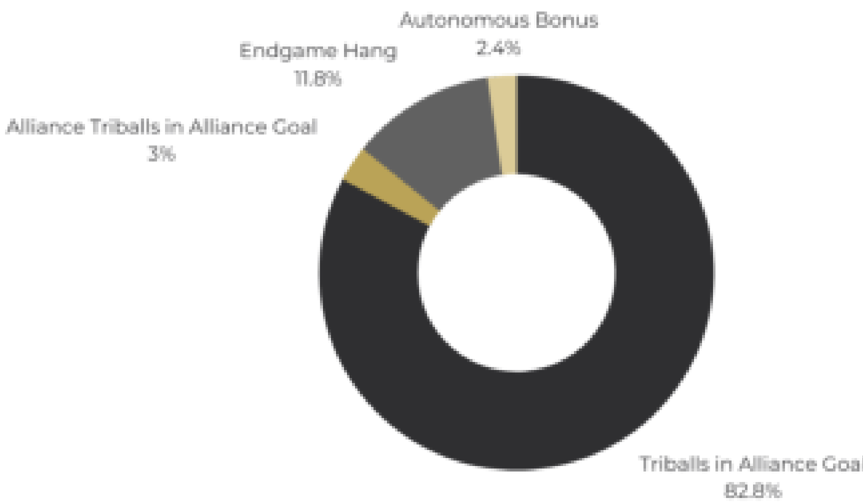


Figure 10: Percentage of total points per object

TIED MATCH BREAKDOWN

	RED ALLIANCE POINTS	BLUE ALLIANCE POINTS
TRIBALL IN ALLIANCE GOAL	150 (30 SCORED TRIBALLS)	150 (30 SCORED TRIBALLS)
ENDGAME HANG	20	20
AUTONOMOUS BONUS	8	8
TOTAL SCORE	150 - 198	150 - 198

Table 2: Theoretical Match Breakdown with Tied Triballs

- Table 2 shows a theoretical match where each team scores an equal amount of Triballs to show the effects of the endgame and autonomous bonus

- Endgame Hanging Points
    - Endgame points can be a **0 to 40 point swing** based on the hang heights of each alliance. Therefore, 0% to 20% of the points in this match would come from endgame hanging
        - In a lopsided match, as seen in Table 1, the endgame hanging points can be overlooked. However, the 40 possible point difference from a maximum and zero hang are equal to 8 Triballs scored in an alliance goal. **These points will prove important in the late stages of the game when teams are likely to score near the same amount of Triballs in an alliance goal.**
    - Autonomous Bonus
        - On a smaller scale than the possible maximum vs. minimum endgame hang, the autonomous bonus provides a 16 point swing. This is because winning autonomous takes 8 points from the opponent and rewards it to you, assuming a team will win the bonus.
        - In a match where both Triballs in alliance goals and with equitable endgame hangs, the autonomous bonus is the only remaining points to be gained.
        - To gain the autonomous bonus points, an alliance must score the most points in the 45 second autonomous period**. Scoring more points than our opponent in autonomous is a crucial goal of the match**, as teams start the 1:15 driver period with an advantage and with bonus points.

Important Rules and Their Effects

- <SG2> Horizontal Expansion is Limited
    - Robots can not exceed 36″ horizontally at any point in a match
    - This rule, along with the rest of the game manual, places no limit on vertical expansion
        - Robots may expand as high as necessary to reach the elevation bar
        - In the event a game strategy evolves to launching Triballs across the field, this **allows us to counter-strategize by raising something to block their attempts** to move match loaded triballs to their alliance zone
- <SG7> Possession is Limited to one (1) Triball
    - During a match, each robot may not have possession of more than 1 Triball at a time

        - **This means each robot will need to be fast in scoring Triballs**, as it will be a race to quickly match load or intake another and return to scoring
        - Robots must be designed to not be able to be stuck in a position where they are in position of more than 1 Triball, as this could lead to a minor or major rule violation.
- <SG8> Stay out of your opponent's goal unless they are Double-Zoned
    - According to SG8, you are only allowed to de-score Triballs from your opponent's goal if they have two robots on the same side of the field at a time.
        - **While an opponent is double zoned, a robot would have to be short enough to reach or drive under the goal to effectively de-score Triballs.**
- <VUG5> Different Match Load introductions
    - Vex U requires all match loads to contact a field tile, and only a field tile, before it may be acquired by a robot.
        - **This means robots will be required to have an intake that can easily full a Triball out of the match load zone and over the bar to match load efficiently**
        - The intake will have to move between match loads to allow a second match load to be entered

Potential Strategies

Offensive Strategies

- Quickly launch all Triballs from the match load zones across to your alliance zone
    - This strategy would allow for all Triballs to be quickly introduced into the match and into a scored position for your alliance
    - From your alliance zone, the Triballs would be able to be quickly moved into your alliance goal
- Launch Triballs from one robot to another with a catcher/net that directly scores into your alliance goal
    - This strategy would allow for the removal the opponent from de-scoring and taking your match loaded Triballs
    - With this strategy, you would not risk doubling zoning as the two robots are always on opposing sides of the long bar

- Intake one Triball at a time with both robots
    - With this strategy, both robots would travel from the match load zone to the alliance goal with one Triball at a time
    - This strategy prevents opponents from stealing your match loads, as they are always in your robot or your alliance goal
    - However, this strategy creates opportunities to be double zoned, as both robots could be at the match loads or scoring in the alliance goal at the same time, which would allow the opposing team to de-score Triballs from our goal
- BLRS2 will also collectively meet on October 11th to discuss offensive strategies before settling on our robot design. This meeting will be recorded in the Competition Robots segment of the notebook.

Defensive Strategies

- De-scoring Triballs from the opponents goal
    - Building a robot that is short enough to reach under the opponent’s goal would allow for their opposing goal to be emptied while the opponent is double zoned.
- Build a wall
    - If the opposing teams are attempting to launch Triballs across the field, creating a blocking structure would allow for us to prevent them from launching them across the field
    - As highlighted in <SG2>, any blocker could expand vertically as there is only a horizontal expansion limit
- Blocking your opponents from their match load zone
- BLRS2 will also collectively meet on October 11th to discuss defensive strategies before settling on our robot design. This meeting will be recorded in the Competition Robots segment of the notebook.

Our Main Takeaways

- Considering that only 16 out of 60 Triballs start on the field, we believe that **possession of the limited Triballs** will be key in match play.
    - This is compounded by the ease of interfering with match loads, as robots are **not protected when scoring** the 22 match loads per alliance.
- Skills, on the other hand, leans much more into scoring **as much as possible, as fast as possible**.
    - Between two robots, we will have to be able to **match load 44 Triballs**, in addition to the Triballs starting on the field.
    - We will also need to allocate time for **scoring all 56 Triballs** in the goal, as well as **elevating** our robot.
- Key Differences:
    - Where Skills strategies aim to score Triballs **as fast as possible**, Competition robots focus more on scoring more of the limited Triballs on the field **without losing possession** to the other team.
    - As the two strategies are very different, we believe that **developing four separate robots, two each for Skills and Competition play**, will enable each pair to be more successful at their respective tasks.

PURDUE SIGBOTS

BLRS2 COMPETITION ROBOTS

Task and Timeline Update: October

Date	@October 2, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Conner Siebert
Design Cycle	Design Cycle 1

Major Plans for October

Team Planning Meeting

To begin the development of the competition robots, our team plans to **start the season with a general meeting focused on the match strategies**.

- Each team member should attend the meeting with a match and autonomous strategy
  - A power point presentation with their strategy to present
  - By developing match strategies independently, our team feels it will lead to a diverse pool of options
- After each member has had the opportunity to present on their idea, the meeting will become an open discussion for all members
  - This portion of the meeting will allow us to discuss the viability of each strategy, and lead to a **conclusion of the direction we would like the competition robots to go towards**

Begin Brainstorming, Prototyping, and Design

After the team has determined our early season strategy, work will begin on the robot design

- Breakout meetings will occur with mechanics members for the comp team to iron out details for the robot
  - This will include the chassis type, Triball manipulators, and an endgame hanging mechanism
  - These meetings will allow all mechanics members to contribute to the design of the robots

- Prototyping and Testing
    - To assist in the design decisions of the robot, our team plans to spend time **prototyping several options for each mechanism**
    - If we have determined an option using a decision matrix, testing several configurations of the mechanism will optimize our efforts in the design phase
- CAD Design
    - Concurrently with the testing and prototyping, our team will begin to **CAD several aspects of the robot**
    - By beginning with CAD, the robots will be optimized in space, as each subsystem will be designed together instead of around an already existing subsystem

Key Dates and Timeline

- **October 11th: Team Strategy Meeting**
    - Match strategies are due on this day, as it's when the team strategy meeting will occur
- **October 16th: Build and Test Prototypes**
    - With a match strategy developed, the team will begin to **prototype subsystems that will best meet the requirements for matches**
- **October 23th: Begin CAD**
    - As the design of the robot will begin, the chassis will be the first thing to **develop in CAD**
- **October 29th: Begin Subsystems CAD**
    - With a chassis design began and prototypes built, the **end of the month serves as the target date to begin**

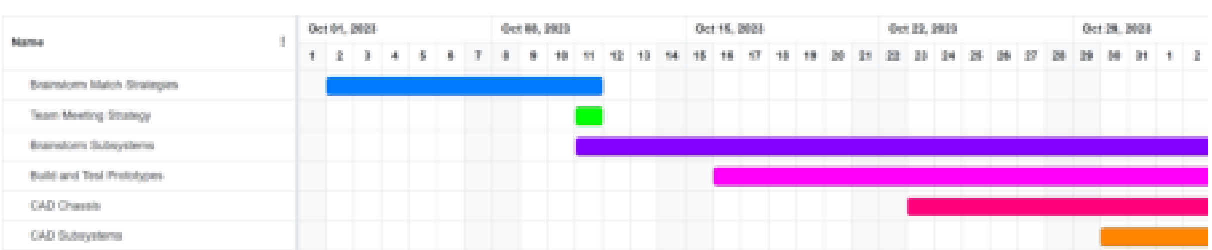


Figure 1: BLRS2 Comp Team October Gantt Chart

Planning: Match Strategies

Date	@October 11, 2023
Category	15" Robot 24" Robot BLRS2 Game Analysis
Authors	Ⓚ Kathleen Lowe Ⓝ Noah Domogalik Ⓐ Alex Lam
Design Cycle	Design Cycle 1

Define the Problem

In this meeting, BLRS2 came together to present on and discuss different strategies for Over Under. We held this meeting so that we could **finalize our initial strategy for the 2023-2024 season** before we began designing our robot as to fully commit to our ideas and hopefully eliminate excessive full robot rebuilds throughout the season.

- Criteria and Constraints:
    - Strategies must plan for both the autonomous and driver control periods, accounting for the differences in play style for each period.
    - At least one robot musts be able to match load.
    - At least one robot must be able to score.
    - Horizontal expansion is limited to 36″ <SG2>.
    - Vertical expansion is unlimited. <SG2>.

Potential Strategies:

Mihir:

- **Autonomous Strategy:**
    - 15″
        - Takes out far alliance Triball
        - Scores both alliance Triballs
        - Parks at goal and scores
    - 24″
        - Sits at the match loader
        - Waits for 15″ to assume park
        - Sweeps Triballs over from the match load zone

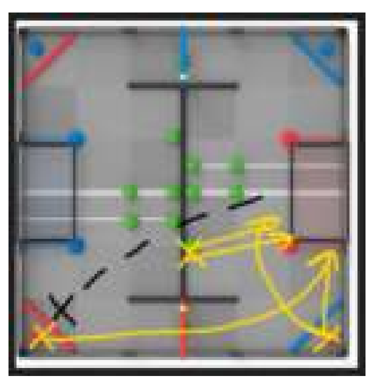


Fig. 1: Mihir's Autonomous Strategy Route (gold is 15″, black is 24″)

- **Driver-Control Strategy:**
    - 15″

        - Plays more offensively
        - Intake grabs Triballs and scores
        - Utilizes a partner hang to work with the 24″
    - 24″
        - Plays more defensively
        - Able to launch match loads to 15″
        - Uses wings to push stray Triballs and de-score opponents

Joe:

- **Autonomous Strategy:**
    - 15″ (Fig. 2)
        - Uses wings to rush for the middle four Triballs
            - Knocks Triballs over the median bar to provide an advantage during driver control
            - Match loads, deploys wings, and then drives along the yellow arrow, pushing all Triballs
    - 24″ (Fig. 3)
        - Clone of 15″ bot
        - Drives from orange spot, turns 180 degrees, create a wall at the purple line

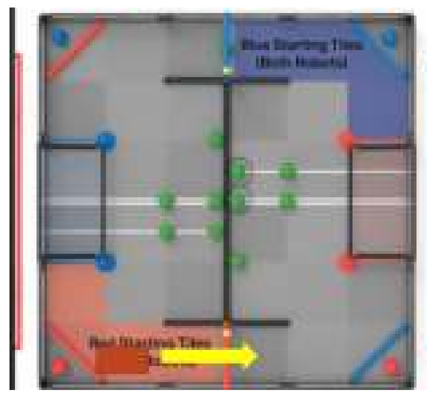


Fig. 2: Bot A’s Autonomous Route

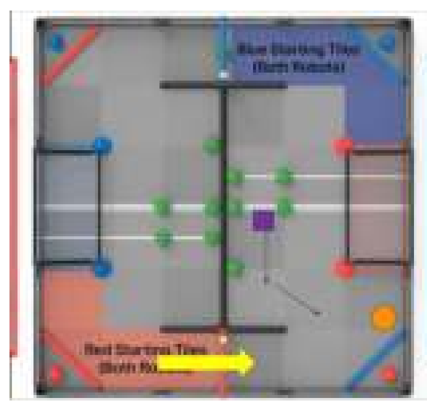


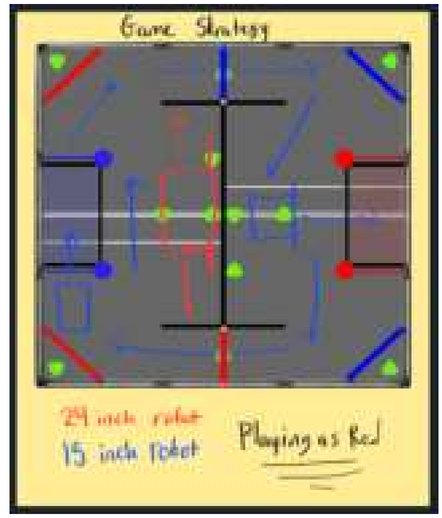
Fig. 3: Bot B’s Autonomous Route

- **Driver-Control Strategy:**
    - Both robots
        - Utilizes a carry-handoff strategy instead of launching Triballs
        - Limit the number of Triballs introduced onto the field as to limit the amount of available points
            - Only introduce enough match loads to have us win if we lose autonomous
        - Possible partner hang

- Large walls made out of mesh for defense

Nathan and Brandon:

- **Match Strategy:**
  - Fig. 4 (Left): Pictured is Nathan and Brandon's route diagram for an Over Under match.
  - Their driver control strategy revolves around having one larger, more defensive bot and one smaller, more nimble bot for offense.



- 15″
  - Circles around field
  - Alternates between scoring Triballs in our goal and de-scoring our opponents
  - Capable of driving into the goal

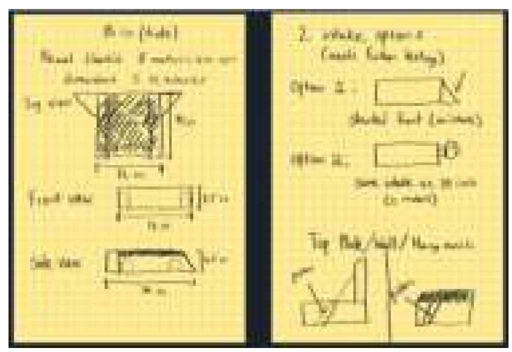


Fig. 5: Diagram of the Kobe Bot

- 24″
  - Use the 36″ mesh wall to play defense on the barrier
  - Shove Triballs over the barrier
  - Can shoot if necessary, but will mainly play defense

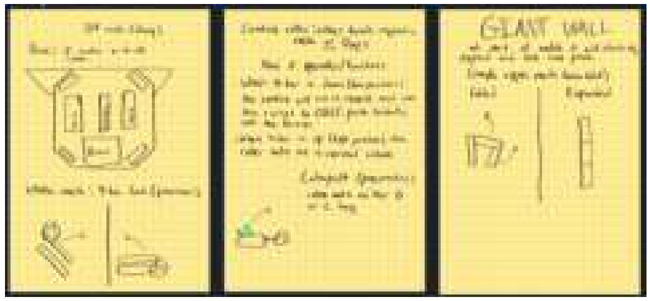


Fig. 6: Diagram of the Shaq Bot

Choosing The Best Solution

To compare each of the three strategies, our team created a list of pros and cons for each option then created a decision matrix.

Pros

Mihir	Joe	Nathan and Brandon
auton-heavy strategy	consistency from not shooting cross-field	both bots are unique so we have higher capabilities
24″ bot has strong defense	defense-centric strategy	strategy throughout driver control is repetitive and replicable
15″ bot is speedy	clone bots for replicability	innovation and creativity for the cyclical movement
	low-scoring matches limit number of triballs in play	

Cons

Mihir	Joe	Nathan and Brandon
15″ bot will encounter heavy defenses	slower cycle time	strategy relies partially on de-scoring
24″ bot will be more massive, limiting elevation	losing autonomous also loses setup for driver control	autonomous is not heavily considered
robots operate at different speeds	opponent could enter a stalemate	strategy could easily lead to us double-zoning
doesn't utilize clone bots		doesn't utilize clone bots

Decision Matrix

The following criteria were considered while we decided upon our team's strategy

- Autonomous Strength (10/40)
  - This season, the autonomous period is longer than normal, giving it more influence in the match.
  - A higher score will indicate a stronger strategy for the autonomous period.
- Offensive Strength (10/40)
  - This criteria is as a measure of how well our robots can score, drive, and interact with game elements.

    - A higher score will indicate higher offensive strength.
- Defensive Strength (10/40)
    - This criteria is a measure of how well our robots can block others, protect our goal, and aid each other throughout a match.
    - A higher score will indicate higher defensive strength.
- Cycle Time (5/40)
    - Especially later in the season, shortening the amount of time it takes for our robots to score will provide us a stronger advantage.
    - A higher score will indicate faster cycle time.
- Buildability (5/40)
    - While we may have big ideas, we need to consider the plausibility of our ideas and strategies.
    - A higher score will indicate more plausible designs.

Strategy	Autonomous Strength (x/10)	Offensive Strength (x/10)	Defensive Strength (x/10)`	Cycle Time (x/5)	Buildability (x/5)	Total (X/40)
Mihir	10	5	6	3	3	27/40
Joe	8	9	8	2	5	**32/40**
Nathan and Brandon	3	8	8	4	5	28/40

After considering each of the three solutions, our team has decided to move forward with Joe's strategy with some modifications. This is because of the following considerations:

- Autonomous Plan
    - Joe's autonomous both sets us up for the win point and provides a solid advantage during Driver Control.
- Defense-Centered Strategy
    - Joe's strategy centers around not introducing match loads onto the field because it provides more unpredictability, while the fewer Triballs helps us guarantee a win.
- Consistency
    - By favoring a carry-handoff strategy instead of shooting across the field, we would be able to reduce variability in scoring and maintain possession of Triballs.

Implementation Plan

- With the core concept of the robots in mind, mechanics members will begin to brainstorm and design specific mechanisms for the robot.
    - Following ideation, **we will move into the Prototype and Design phase of the Engineering Design Process**.
- As decided in our team Gantt Chart, this will take place largely in the rest of October.
    - A more thorough breakdown of our **timeline and overall plan can be found in the next entry**.

Design Brief: Iteration 1

Date	@October 13, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Ⓒ Conner Siebert Ⓚ Kathleen Lowe
Design Cycle	Design Cycle 1

BLRS2 Competition Robot: Iteration 1

Problem Statement

To compete in VRC Over Under, our team needs to develop two robots that are able to meet all constraints of the game manual and complete the task of the game. **These robots should be able to intake Triballs, move them towards an alliance goal, and score them into the goal**. They also need to be able to elevate in the endgame period.

Constraints

- Begin in a 15×15×15 box
- Intake Triballs from the match load zone
- Score Triballs into the alliance goal
- Elevate both robots onto the elevation bars

Criteria

- Be able to score **all 10 match loads during the 45 second autonomous period**.
    - To enter matches in a strong position, our team feels it is necessary to optimize the robots to score Triballs in the autonomous period.
- Avoid defense from opposing robots.
    - As walls and defensive mechanisms become popular in match gameplay, being able to **score around these will prevent opponents from taking control of a match**.
    - This will likely rely more on driver practice as we complete the robots, meaning that we should **thoroughly test each mechanism** to ensure quality performance.

- Achieve elevation at the end of the match.
  - While our ideas of a successful elevation may change as our strategy develops, having the ability to elevate robots will be key in playing at a high level.
  - For our initial robots, **we will prioritize achieving an A-tier elevation** as we research more unique, higher scoring designs.

Timeline For Iteration 1

- October
  - Robot idealization
    - Develop a match strategy using collaborative brainstorming and decision matrices
  - Begin robot CAD
    - Begin designing the robots in CAD 3D design software
- November
  - Finish robot CAD
    - Have a completed model of the robot to begin building
  - Fabricate and print any needed parts
    - As custom parts are legal in VEXU, we will take advantage of this to design a robot exactly to our specifications
  - Build robot
    - Construct the robot following the steps created in the CAD model
- December
  - Complete robot
    - Finish any building still in progress
  - Driver practice and programming
    - Make sure the robot and drivers are ready, and program an autonomous route for match play
- January
  - Prepare for Nuke tournament
    - Complete any final minor adjustments needed before the tournament

- Improve robot based on performance at Nuke
  - Evaluate our performance and plan necessary changes

Iteration 1 Timeline and Gantt Chart

Brainstorming and Decision Matrix: Chassis

Date	@October 18, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Kathleen Lowe Conner Siebert
Design Cycle	Design Cycle 1

Identify the Problem

In order to create the highest quality bots for this season, we want to first design a chassis that will give us **optimal speed, offensive performance, and defensive performance**. We have decided upon the following criteria and constraints in designing our chassis:

- The chassis must work well with the other components of our bot, such as our Triball launcher and endgame mechanism.
  - While this is hard to quantify, we must **design the chassis with the intention of adding additional mechanisms** as we complete brainstorming.
- The chassis must have high traction to push other bots and maximize our defensive prowess.
  - **A successful chassis design will have a pull force of at least 5 pounds**, so that we are able to play effective defense on opposing designs.
- The chassis must have high maneuverability in order to quickly adapt to the varying play styles of our opponents.
  - As we are able to design each chassis type with the speed and gear ratio we prefer, we will set a goal that the **chassis be able to drive effectively at minimum 300RPM**.

Brainstorm Solutions

Solution 1:

- 4 Wheel Mecanum Drive
  - Our Mecanum drive would use 4 Mecanum wheels, one in each corner, and would **allow our robot to drive sideways in either direction** along with forwards and backwards.
  - The larger wheels of the Mecanum drive may increase our chances of smoothly driving over the center barrier.
  - The strafing capabilities of this drivetrain may **help us in aligning our Triball launcher**.

Example of a Mecanum Drive from Servo Magazine

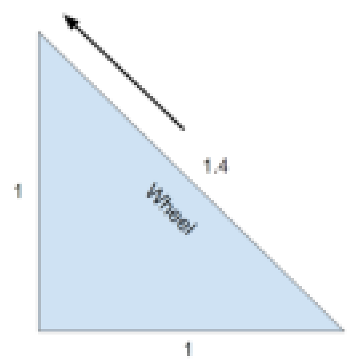
Solution 2:

- 6 Wheel Tank Drive
  - Our tank drive would use 4 omni-wheels, one at each corner, with traction wheels in the middle of either side.
  - This drivetrain would **increase the robot’s defensive power with the traction wheels**.
  - Additionally, the smooth movement of the tank drive would **increase our adaptability during a match.**

Example of 6- Wheel Tank Drive from Game Manual 0

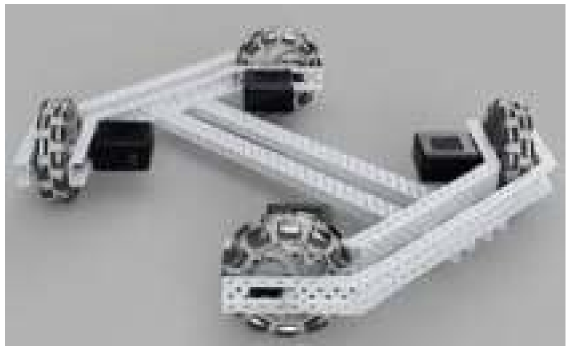
Solution 3:

- 4 Wheel X Drive
  - Our X drive would have an omni-wheel in each corner, each angled at 45 degrees.
    - This **allows for side-to-side movement** in addition to the forward and backward.
  - This drivetrain would allow for enhanced flexibility and ease of movement.
    - With the wheels oriented at an angle, this provides for a 1.4x speed increase, shown in the geometry below.



Triangle geometry

  - The angles of the wheels provides inherent defense against our opponents.



Example of X Drive from GrabCAD

Prototype Solutions + Testing

To assist in determining the best chassis type, our team built prototypes of each of the chassis types listed above, **which were used to test possible robot chassis designs**. The goal of these test were to determine the viability of the chassis to withstand and perform defense.

Test 1: Forwards Force

To determine each chassis' ability to withstand head on pushing, **our team tested the pushing force of each chassis type**.

Testing Procedure

1. The chassis is set onto the field
2. A force scale is attached to a point at the rear of the chassis
3. The chassis is driven at full power forward, as the peak force of the chassis is recorded
4. The chassis is stopped, and the test is reset
5. The test is repeated until 5 test have been completed for each chassis type

Test Data

	Test 1 (lbs)	Test 2 (lbs)	Test 3 (lbs)	Test 4 (lbs)	Test 5 (lbs)	Average (lbs)
Mecanum Drive	5.21	4.78	4.91	5.03	5.23	5.03
Tank Drive	7.23	6.97	7.12	7.35	7.21	7.18
X-Drive	2.51	2.97	2.64	2.79	2.61	2.70

Test 2: Side Pushing Distance

In a secondary defensive biased test, each chassis was tested for it's ability to **resist a push from the side**, at a 90 degree angle from it's forwards direction.

Testing Procedure

1. The chassis is placed onto the field
2. The force scale is hooked onto the middle of a chassis, along the left edge
3. Force is applied to the end of the scale, until the chassis begins to move in a direction perpendicular to it's forward's direction
4. The maximum force is recorded and force is removed
5. The test is reset and reran until 5 test are completed for each chassis type

Test Data

	Test 1 (lbs)	Test 2 (lbs)	Test 3 (lbs)	Test 4 (lbs)	Test 5 (lbs)	Average (lbs)
Mecanum Drive	3.87	3.98	3.88	3.60	3.72	3.81
Tank Drive	4.76	4.60	4.81	4.90	4.70	4.75
X-Drive	1.54	1.23	1.39	1.49	1.35	1.40

Testing Conclusions

From the two test conducted, the **tank drive appears to be a better option**. This is because of the following factors:

- Higher pushing force
  - From test 1, it is seen the **tank drive has 2lb and 4lb higher pushing forces than the mecanum drive and X-drive** respectively
  - This difference in forces are of a large margin, showing that the tank drive would be significantly better when facing defense compared to the other two drives
  - As the test occurred, each of the three chassis would reach their maximum force right as the wheels began to slip, **indicating potential traction issues instead of a lack of power**
- Better Side Resistance
  - Resulting from test 2, the **tank drive stood up to theoretical defensive pushing from the side better** compared to the other two chassis, with at least a 0.9lb difference.
  - While being pulled from the side, the **x-drive was easy to pull due to the omni-wheel rollers**. In comparisons, the tank and mecanum drive wheels would start to skid along the field tiles

Select the Best Solution

To compare each of the three drivetrains, our team created a list of pros and cons for each option then created a decision matrix.

Pros

Mecanum Drive	Tank Drive	X Drive
Largest wheels	High traction	Angled wheels
Strafing capabilities	Largest number of wheels	More defensive build
Simpler robot design	Flexible number of motors	

Cons

Mecanum Drive	Tank Drive	X Drive
Fewer rollers means less traction	Largest number of motors	Less capable of conquering middle barrier
Slower movement speed	More challenging to program due to possible drift	Fixed number of motors
Fixed number of motors		

Decision Matrix

The following criteria were considered while we decided upon the design of our chassis.

- Speed (x/10)
  - Because Over Under heavily relies on how fast a team can reach and score a Triball, **speed is a key factor in our robot design**.
  - The faster we could theoretically make a working chassis, the higher the score.
- Maneuverability (x/10)

- This season, **our robot must have a high maneuverability** to not only dodge game elements but also drive over the middle barrier and other bars on the field with ease.
  - The more maneuverable a chassis is, the higher score it receives.
- Match Flexibility (x/10)
  - We want our robot design to reflect the adaptability that we prioritize in our match strategy.
  - A flexible chassis design would drive with smoothness and strength, **complementing the other components of our robot and allowing our driver more control** over the match.
- Ease of Integration (x/5)
  - Due to the limited amount of space on our robot, we want to ensure that our choice of chassis **does not restrict our options** as we add more components to our robot design.
  - The easier it is to integrate new subsystems, the higher score.
- Forwards Pushing Power (x/10)
  - Pushing power **enhances the defensive capabilities of our robot**, which will grow crucial during defensive play.
  - This score is taken directly from testing results.
- Side Pushing Resistance (x/5)
  - As we intend for these robots to be pushed from the sides while attempting offensive movements, a **chassis that can withstand forces from the side would be ideal.**
  - This score is taken directly from testing results.

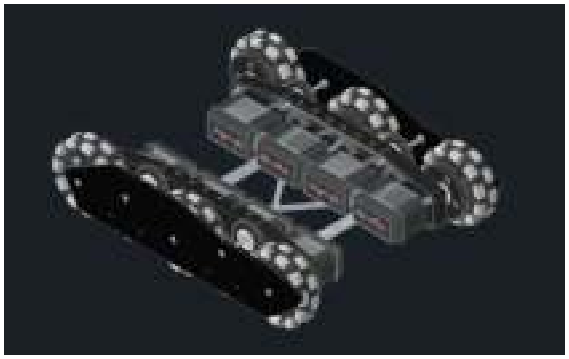
	Speed (x/10)	Maneuverability (x/10)	Match Flexibility (x/10)	Ease of Integration (x/5)	Pushing Power (x/10)	Side Pushing Resistance (x/5)	Total (x/50)
Mecanum Drive	6	6	5	2	5	4	28
Tank Drive	8	7	8	3	8	5	39
X Drive	9	4	6	4	2	1	26

After considering each of the three solutions, our team has decided to move forward with a tank drive. This is because of the following considerations:

- High match flexibility
  - The tank drive allows us to easily adjust our math strategy as necessary. Because of this drivetrain's high maneuverability, **we can play a more defensive game without sacrificing any of our mobility or speed.**
- Wheel number and size
  - Our tank drive will have six wheels, all of which are larger than half of the height of the center barrier.
  - The larger number of wheels on our robot will improve our traction, allowing us to **play a stronger defensive gain and more easily push our opponents**.
  - The size of our wheels ensures that our chassis can clear the middle barrier without excessively relying on additional sleds.
- Customizations
  - While we are not dependent on external sleds, the tank drive's setup means that we can **still easily integrate sleds to aid in our crossing of field barriers**.
  - Additionally, the tank drive includes ample room for additional guards, braces, and wedges to further enhance the defensive capabilities of our chassis.

Implementation Plan

- To begin the Prototype and Design phase of the Engineering Design Process, we have made a rough CAD of a possible chassis, showcased below.



Our Initial Chassis CAD

- The chassis uses 8 V5 Motors to ensure that we have ample pushing power, as well as three wheels for maneuverability.
  - As we progress further into the Prototype and Design phase, **we will decide on and record more specifics in a later entry.**
- As we wish to spend considerable time on deciding other mechanisms and making sure the chassis will be able to support them, **we will aim to have the chassis completed in late November.**
  - To guide our construction of the chassis, we have included a timeline below:
    - **October 30**
      - Decide the specifics of what we would like to include in the chassis
    - **November 13**
      - Complete the chassis in CAD according to desired specifications
    - **November 21**
      - Complete the initial build of the chassis design
      - Record the development process in Notion

Brainstorming and Decision Matrix: Triball Launcher

Date	@October 24, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Noah Domogalik Jacob Zawacki
Design Cycle	Design Cycle 1

Identify the Problem

While our match strategy will largely consist of carrying Triballs from the match load zone to the scoring goal, having a Triball launcher will **allow us additional flexibility should we need to quickly introduce match loads**. A successful launcher should meet the following:

- Launch Triballs within an acceptable grouping across the field
    - For our initial decision, we will consider acceptable grouping a **12″ diameter circle**
- Integrate well with existing subsystems
    - A launcher will need to fit well within the chassis and **feasibly work under VEXU rules**, as described in our Game Analysis
- Quickly launch match loads without downtime
    - For reference, a speed of **1 Triball per second (for a total of 22 seconds) will be our benchmark for testing**

Brainstorm Solutions

Solution 1:

- Catapult
    - The most widely-used mechanism by teams in competition so far this year, the catapult **utilizes a slip-gear mechanism to launch Triballs across the field.**
        - A powered gear drives a slip-gear, a gear with a portion of the teeth shaved down. The catapult is attached to the slip-gear, which is tensioned upward.
        - As the powered gear drives over the slip-gear, the **shaved teeth disengage and allow the tensioned catapult to launch upwards**, launching the payload.
    - The limiting factors of the catapult would likely be the cycle time as the **catapult will have to reset its position after each launch.** Additionally, it may not integrate well with other subsystems due to the large form factor.

Figure 1: Catapult - BLRS Wiki, 240P

Solution 2:

- Passthrough/Flywheel
    - While the flywheel portion is fairly common, the passthrough mechanism would be more specific to our design. **The interior of the robot would be completely open, leaving Triballs entering the system from the intake able to move completely through** the robot to the other side.
        - This would enable the robot to intake from the front and launch directly out of the back, **speeding the cycle time considerably and integrating well with the rest of the robot.**
    - The flywheel itself would consist of a **powered wheel spinning at a high RPM**, with either another roller or a solid hood to create compression and launch the Triball.
    - This option seems fairly strong, with a fast potential cycle time and easy integration with the rest of the robot.

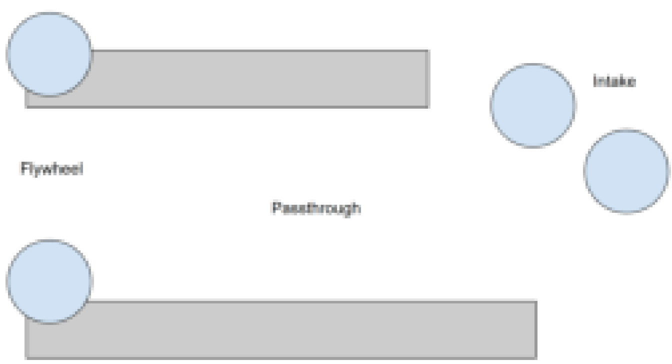


Figure 2: Diagram showing potential passthrough mechanism

Solution 3:

- Kicker
    - A kicker is very similar to the catapult mechanism as far as overall construction, utilizing the same sort of slip-gear mechanism to create a fast launching motion.
        - The main difference between the kicker and the catapult is that rather than place Triballs directly into the catapult mechanism to launch them, the **kicker would impact Triballs placed on a solid, unmoving surface to launch them**.

- While this would likely be a faster mechanism than the catapult, as the kicker mechanism could be **smaller and faster than the catapult** as less momentum is needed for a powerful launch, the kicker would be a **much more difficult mechanism to integrate with the rest of the robot**.
    - This is due to the solid platform for the kicker likely being fairly small, and having to move Triballs from the intake to the small platform.

Figure 3: Kicker mechanism - 5203G

Prototype Solutions + Testing

To assist in determining the best launcher type, our team built very rough prototypes of each mechanism. Two tests were conducted, to **determine the speed and accuracy of each possible solution** to assist in determining the best solution.

Test 1: Firing Speed

To find the highest rate of speed possible our team conducted a speed test on each launcher type. This was conducted using the following steps:

Testing Procedure

1. The launcher was placed at the match load zone, in the same location it would be positioned on the robot
2. A timer is started as the first Triball is placed into the match load zone or launcher
3. 22 Triballs were introduced into the launcher in the same manner that they would be in a match
    a. For the flywheel, as there was not a prototyped intake, it was assumed a wait time of 0.75 seconds would occur between Triballs being entered into the robot
    b. For the catapult, Triballs were placed in a legal match load position to be picked up by the catapult
    c. For the kicker, a 1 second wait time was assumed before placing into launching position to assume time for intaking.
4. The timer is stopped as the 22nd Triball is shoot

Test Data

	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Flywheel	16.75	16.67	16.49	16.56	16.73	16.64 seconds
Catapult	14.52	14.67	14.64	14.56	14.72	14.66 seconds
Kicker	18.32	18.41	18.33	18.34	18.30	18.34 seconds

Test 2: Firing Accuracy

To determine how accurate each launch will be in getting Triballs across to the field close to the scoring goal, we tested the range of which Triballs would end after being launched.

Testing Procedure

1. The launcher was placed at the match load zone, in the same location it would be positioned on the robot
2. A Triballs is placed into the launcher, aimed towards the alliance goal where the partner robot would be pushing in Triballs
3. As a Triball is launched, it is recorded if it lands within the 12″ diameter circle target placed in front of the alliance goal where the partner robot would be positioned
4. This process is repeated until 22 match loads have been fired, completing the test

Test Data

	Test 1 (x/22)	Test 2 (x/22)	Test 3 (x/22)	Test 4 (x/22)	Test 5 (x/22)	Average
Flywheel	18	17	18	16	17	17.2
Catapult	21	20	22	18	19	20
Kicker	19	19	18	18	19	18.6

Testing Conclusions

By the data, the catapult appears to be a better option. This is because of the following factors:

- The catapult mechanism has both a **higher average rate of fire and a higher accuracy** than the other two mechanisms.
    - The kicker trails just behind the catapult, with the flywheel just behind.
    - It is important to note that while the kicker and catapult are fairly rigid in their definitions, **different variations of flywheel may lead to better overall performance**.
- Aside from the testing data, there are numerous other factors to keep into account, which will be summarized below:

Select the Best Solution

To compare each of the three launchers, our team created a list of pros and cons for each option then created a decision matrix:

Pros

Catapult	Flywheel/Passthrough	Kicker
Many examples available, easy to develop	Very fast cycle time between Triball launches	Small form factor and easy to construct
Large flexibility when feeding Triballs from intake	Easy to implement with other subsystems	Fast cycle time between Triball launches
	Fits well into our overall game strategy	

Cons

Catapult	Flywheel/Passthrough	Kicker
Large form factor	Requires more development due to being more experimental	Hard to integrate into other subsystems and receive Triballs
More difficult to integrate with existing subsystems		Likely slower cycle time between Triball launches
Likely slow cycle time between Triball launches		

Decision Matrix

The following criteria were considered while we decided upon which launcher variation to use for our robots. **Higher scores indicate more favorable valuations of the criteria.**

- Integration with subsystems (x/15)
  - A very large part of our strategy this season involves carrying the Triball directly to the goal to maintain possession at all times.
  - The launcher needs to be able to quickly match-load at any given time, and work well with the intake to maintain possession at all times.
  - Because this is so important, it has the highest weight of all criteria.
- Accuracy (x/5)
  - We will need the launcher to accurately launch Triballs when necessary, so we don't accidentally lose possession.
  - This score is taken directly from the above testing data.
- Speed (x/5)
  - We will need to quickly match-load into the goal, so speed is key.
  - This score is taken directly from the testing data.
- Ability to launch under defense (x/10)
  - We will likely be under heavy defense throughout the match, we will need to be able to launch Triballs under defense.
  - Because this is likely to happen in every match, it has a high weight.
- Ease of loading (x/10)
  - With speed and keeping possession being key elements, it must be easy to load Triballs into the launcher.
  - Because this is key to making a quick, successful launcher, it has a high weight.

	Integration	Accuracy	Speed	Defense	Ease of Loading	Total (x/45)
Catapult	12	5	5	8	6	36
Kicker	10	4	4	8	4	30
Flywheel	15	3	3	10	10	**41**

After considering each of the three solutions, our team has decided to move forward with a flywheel and passthrough. This is because of the following considerations:

- Ability to Integrate
  - As the flywheel uses the same medium to launch Triballs as it does to intake them, roller wheels, we will be able to make a **passthrough mechanism of an uninterrupted chain of rollers through the robot.**
  - This will allow us to have a fast, quick method of **ensuring possession of the ball at all times** (encased within the robot) as well as a quick method of launching should the need arise.
- Testing Results
  - While the flywheel did not perform as well in testing as the catapult and kicker, it did not perform poorly enough to be unacceptable.
    - Additionally, **we plan to test additional variations of the flywheel to try for better performance.**
  - We believe that the ability to passthrough, as well as the weaker abilities to maneuver Triballs with the catapult and kicker, outweigh the lower performance data.

Implementation Plan

- As the testing results were passable, but not great, for the flywheel, we will need to test different variations of the flywheel to determine how we can improve performance.
  - To do this, **we will construct different variations of flywheel mechanisms and test a standard procedure**.
  - We will record this in a separate entry, likely following this one.
- As **we will need to have the chassis CAD completed to construct the passthrough**, we will have a good amount of time to test flywheel variations and determine a course of action.
  - After the chassis has been completed, we will begin the design of the flywheel mechanism
  - **We will likely begin to construct other mechanisms before the flywheel**, so that our robot has more basic functionality unlocked before complex launching.
- Here is a timeline breakdown for the launcher mechanism:
  - **November 2**
    - Have the specific flywheel variation decided upon.
  - **December 7**
    - Have the flywheel designed in CAD
  - **January 1**
    - Have the flywheel built on the robot

Prototyping and Testing: Flywheel Variations

Date	@October 30, 2023
Category	15" Robot 24" Robot BLRS2
Authors	B Brandon Liu C Conner Siebert J Jacob Zawacki
Design Cycle	Design Cycle 1

Overall Objectives

With the overall design choice of making a flywheel mechanism to launch Triballs having been chosen, **we now need to take steps to decide which variation of flywheel would be most effective**. We will be testing the following flywheel variations:

- Rubber Band Hood
    - Half-Hood
        - Shorter hood for a more compact launcher.
    - Full Hood
        - Longer hood for more contact and direction on the Triball.
- Lexan Hood
    - 5.375″ spacing
        - Less compression for a quicker launch.
    - 4.875″ spacing
        - More compression for more powerful launches.

We will consider the following metrics in our testing:

1. Launch Distance
    a. A higher launch distance will give us more flexibility when launching Triballs towards the goal.
    b. A successful variation will have an average launch distance of **at least 7 feet (84 in)**.

2. Triball Grouping
    a. As we plan to launch Triballs over and then push a group into the goal, having a compact grouping as they land will be ideal.
    b. A successful launch will have **90% of Triballs land within a circle diameter of 2 feet**.
3. Backspin/Topspin
    a. The compression for the flywheel can either be enacted from the top or bottom, changing the spin the Triballs are launched with.
    b. Each variation will be tested with both backspin and topspin in mind.
    c. Topspin and Backspin will have a large effect on how the Triball is launched out of the flywheel, **either launching it higher up (Backspin) or lower down (Topspin)**
        i. This will drastically effect if the Triballs roll on the ground before settling, as well as the overall spread.
        ii. We do not have a predicted metric, so **we will test both and observe which has the better grouping.**

Distance Testing Procedure

Step 1: Place the launcher prototype in the match load zone, facing across the field.

Step 2: Power the launcher prototype to the maximum velocity. To do this, we will use the brain screen to turn velocity up to 100%.

Step 3: Feed a Triball into the launcher prototype by hand. Repeat 10 times.

Step 4: Adjust the launcher prototype to test either backspin or topspin, whichever has not been tested yet.

Step 5: Repeat the testing steps for backspin or topspin.

Step 6: Repeat the process for additional launcher variations.

Distance Testing Data

Table 1: Rubber Band Hood

	Full Hood Topspin Distance (in.)	Full Hood Backspin Distance (in.)	Half Hood Topspin Distance (in.)	Half Hood Backspin Distance (in.)
Test 1	54	6	48	4

	Full Hood Topspin Distance (in.)	Full Hood Backspin Distance (in.)	Half Hood Topspin Distance (in.)	Half Hood Backspin Distance (in.)
Test 2	66	41	52	34
Test 3	73	42	55	42
Test 4	86	47	55	42
Test 5	86	47	62	50
Test 6	98	52	79	56
Test 7	107	53	85	61
Test 8	107	58	86	61
Test 9	108	98	93	64
Test 10	109	94	108	79
Average	89.4 in.	53.8 in.	72.3 in.	49.3 in.

Table 2: Lexan Hood

	5.375 in. Topspin Distance (in.)	5.375 in. Backspin Distance (in.)	4.875 in. Topspin Distance (in.)	4.875 in. Topspin Distance (in.)
Test 1	68	14	4	7
Test 2	75	25	66	8
Test 3	79	27	80	48
Test 4	80	31	100	48
Test 5	85	34	100	54
Test 6	92	48	107	65
Test 7	103	53	107	81
Test 8	103	59	112	86
Test 9	112	62	112	91
Test 10	112	64	113	91
Average	90.9 in.	41.7 in.	90.1 in.	57.9 in.

Table 3: Horizontal Flywheel

	Standard Configuration
Test 1	85
Test 2	87
Test 3	91

	Standard Configuration
Test 4	91
Test 5	91
Test 6	92
Test 7	100
Test 8	101
Test 9	108
Test 10	111
Average	95.7 in.

Distance Testing Observations

In the distance testing, we noticed the following points of interest:

- To easily compare the different flywheel variations, with so many numbers, we have included a table with the average distance for each variation below:

Flywheel Variation	Average Distance (inches)
Rubber Band Full Topspin	89.4
Rubber Band Full Backspin	53.8
Rubber Band Half Topspin	72.3
Rubber Band Half Backspin	49.3
Lexan 5.375 in. Topspin	90.9
Lexan 5.375 in. Backspin	41.7
Lexan 4.875 in. Topspin	90.1
Lexan 4.875 in. Backspin	57.9
Horizontal Flywheel	95.7

- In looking at the data, the **Horizontal Flywheel had the furthest average distance**, followed by both Topspin Lexan Hood Flywheels and the Topspin Full Rubber Band Hood. These options were the only ones to meet the distance threshold.
- Triballs launched with **topspin tended to roll more than other Triballs** (as shown in the grouping tests), leading to a tradeoff with their strong launch distances. **The Horizontal setup did not share this issue.**

Grouping Testing Procedure

Step 1: Complete the distance test using the outlined procedure.

Step 2: Record the positions of each Triball launched.

Step 3: Observe the grouping shown and look for trends.

Grouping Testing Data

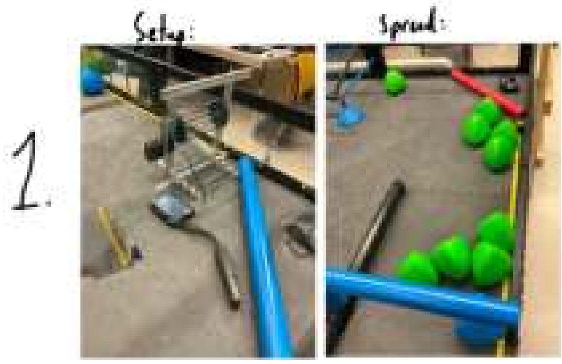


Figure 1: Topspin Half Hood Grouping

Figure 2: Backspin Half Hood Grouping

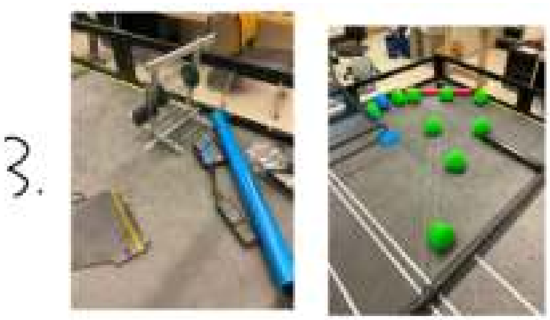


Figure 3: Topspin Full Hood Grouping

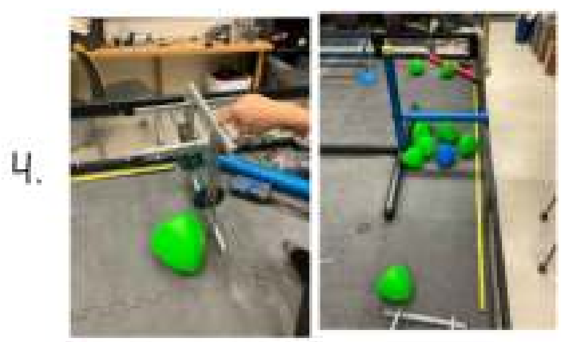


Figure 4: Backspin Full Hood Grouping

5.

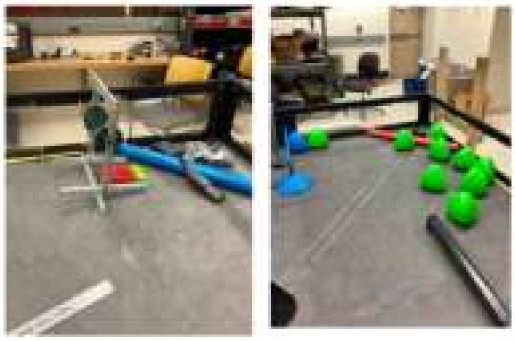


Figure 5: 5.375″ Topspin Grouping

6.

Figure 6: 5.375″ Backspin Grouping

7.

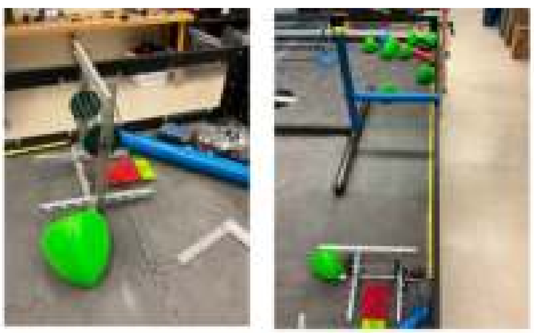


Figure 7: 4.875″ Topspin Grouping

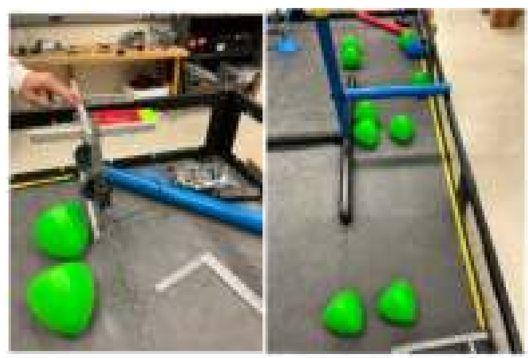


Figure 8: 4.875″ Backspin Grouping

Figure 9: Horizontal Flywheel Grouping

Grouping Testing Observations

In the grouping testing, we noticed the following points of interest:

- The grouping for the **horizontal flywheel was noticeably tighter than most other variations**, as well as launched the Triballs in the area needed for our match strategy
    - Most other variations with a close spread used topspin, which tend to roll as observed in previous testing, meaning that **may not be a consistent representation of overall performance**
- Variations using **backspin were either too spread out or too far from the goal**, meaning they do not fulfill the criteria.

Conclusions

- The **grouping with the horizontal flywheel is noticeably better than any other variation**.
- The horizontal flywheel also launched the furthest distance, with others having to roll the Triballs further to catch up.
- With this data in mind, we believe that the **Horizontal Flywheel** is the best choice for our launcher type, and we will begin to design and implement a refined version of our current prototype.

Task and Timeline Update: November

Date	@November 1, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Conner Siebert
Design Cycle	Design Cycle 1

October Recap

October was a productive month for the competition robot team. Some major accomplishments includes:

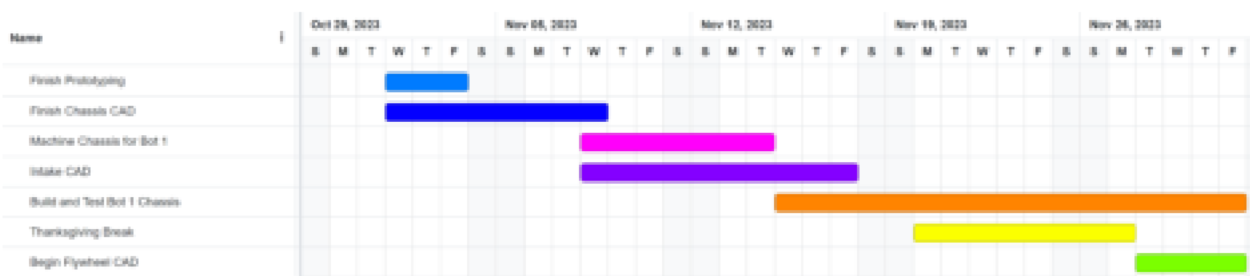
- Match strategy was developed
  - Following a successful planning meeting, **our team decided upon the criteria and constraints for the robot design**
  - It was decided that both robots will follow the same design
- Chassis Design was began
  - After considering 3 possible options for the chassis, **our team began to design the chassis in CAD**
- Prototypes were built and tested
  - Following the strategy meeting, it was determined that our team needed to decide on a Triball launcher type and an intake
  - Several prototypes were built, for example a **catapult and flywheel were tested against each other**

Compared to our overall timeline, October saw the team be set behind the original schedule. **This will be adjusted in November, to allow more time to be dedicated to prototyping and design work.**

November Task and Timeline

For the month of November, our goal is to advance upon the design decisions made in October along with beginning to build the chassis for bot 1.

- Finish Prototyping and Chassis CAD
  - These first tasks from October should be completed within the first week of the month, allowing for further design work to occur
- Design of the intake and flywheel
  - These two subsystems were determined to be important to the design of the robot, **our goal is to complete the design of both of these subsystems in CAD this month**
- Manufacturing and machining bot 1 chassis
  - With the chassis CAD to be completed early in the month, **we are going to strive to complete the manufacturing and building of the chassis in November** to ensure there is time for any issues that arise along with the design



Brainstorming and Decision Matrix: Triball Intake

Date	@November 8, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Alex Lam Stephen Hohnholt
Design Cycle	Design Cycle 1

Define the Problem

In order to score points, the robots need to be able to **grab and manipulate Triballs from**:

- Either Offensive Zone.
- Any Match Load Zone.

This needs to be done in **both the Autonomous Period and Driver-Controlled Period** in order to

- Score Triballs in our Alliance's Goal and Zone for additional points.
- Win the Autonomous Bonus.
- Win the Autonomous Win Point.

Potential Solutions

Solution 1:

- Claw intake
  - This solution involves using a **claw-like mechanism made with either C-channels or 3D-printed material** that would corral the Triball (either in the load zone or on the field) and lift it into the launcher.
  - Powering methods could be either pneumatic or motors

This is the standard Vex Claw Bot. It has a motor-powered claw that can grab a lot of different (smaller) items.

Solution 2:

- Flex Wheel Intake
  - Flex Wheels could be **horizontal or vertical rollers that grab Triballs as the robot runs near them**. Flex Wheels are made out of a rubbery material and are flexible with fantastic grip.
  - Motor(s) will be needed as the powering method. Using Motors, the robot can spin the Flex Wheels to grab the Triballs and manipulate them how we would like.

This is the BLRS Change Up 15″ robot named Aldrin. It used Flex Wheel Intakes (although modified to the game elements from a previous year), to grab balls off of the field and score them into the goals.

Solution 3:

- Rubber Band Roller Intake
    - Rubber Band Rollers are Sprockets (which may be 3D-Printed, or the standard Vex Sprockets), with rubber bands tensioned so that they will stay on the teeth of the sprocket. **The bands then help with gripping onto the ball and then manipulating them**.
    - Motor(s) will be needed as the powering method. Using Motors, the robot can spin sprockets linked to the rubber band rollers.

This is 7701T's Turning Point Robot. Turning Point was a VEX Game that required you to shoot balls at flags. The Rubber Band rollers shown grabbed the (significantly smaller) balls and primed them to shoot.

Select the Best Solution

To compare each of the three intake solutions, our team created a list of pros and cons for each option then created a decision matrix

Pros

Claw	Flex Wheels	Rubber Band Rollers
Most Consistent	Great Fielding	Wide Fielding Range
Easy to implement for Match Loading	Solid for Match Loading	Fast
	Fast	Easier to power

Cons

Claw	Flex Wheels	Rubber Band Rollers
Slow	Difficult to Control	Take up a lot of space
Narrow Fielding range	Expensive	Hard to Match Load
Difficult to pick up during the Driver-Controlled Period	Take up a lot of space	

Decision Matrix

The following criteria were considered while we decided upon for the intake solutions:

- Speed (x/10)
  - The robot needs to be able to intake Triballs quickly in order to be efficient in Matches.
  - A higher score indicates a faster intake.
- Powering (x/5)
  - If it takes a lot to power the mechanisms, it may be inefficient and cause more problems than do good.
  - A higher score indicates more efficient power consumption.
- Manipulability (x/5)
  - If we cannot control efficiently and easily how the Triball is used within the robot, then that is an issue that needs to be identified.
  - A higher score indicates higher ability to manipulate Triballs.
- Size (x/5)
  - If the intake takes up too much space, we may have issues fitting other mechanisms within the size limit for the robot.
  - A higher score indicates a more space-efficient intake.
- Ease of Use (x/5)
  - The intake should not be difficult to use in the Match, if it requires a lot of precision, another option may need to be considered.
  - A higher score indicates an easier to use intake.

Option\Criterion	Speed (x/10)	Powering (x/5)	Manipulability (x/5)	Size (x/5)	Ease of Use (x/5)	Total (x/30)
Claw	2	4	2	4	1	13
Flex Wheels	8	2	3	2	4	19
Rubber Band Rollers	10	2	4	2	5	23

After considering each of the three solutions, our team has decided to move forward with rubber band rollers. This is because of the following considerations:

- The speed of rubber band rollers are unmatched when it comes to spherical shapes.
  - In past games (Nothing but Net, Turning Point, Change Up) rubber band rollers are often used. In our experience, this is because the **tension pushes the ball forward quickly while still having complete control of the ball inside the robot.**
- It is much easier to manipulate spherical shapes with rubber band rollers to fall along certain paths.
  - With the rigidity of Flex Wheels and Claws, it is hard to control the Game Elements in their entire path without over-complicated designs.
- Drivers and Programmers will have no issues with picking up balls from the field.
  - **Rubber Band Rollers are very easy to both program and drive** as seen in our experience in the mentioned games (Nothing but Net, Turning Point, Change Up)

Implementation Plan

- To implement the Rubber Band Roller Intake method, **our team will begin to design the intake within CAD** before creating a prototype to further ensure the design remains viable
- After the design is completed within CAD, **the intake will be built and tested on the robot** in preparation for competition (both driving and autonomous)
- Here is a rough **timeline showing our plans** for the intake:
    - **November 14**
        - Complete the Intake CAD
    - **December 12**
        - Build and test the intake

Comp Software: Driver Controls

Date	@November 15, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Ⓜ Mihir Laud
Design Cycle	Design Cycle 1

Goals

This meeting, we aimed to accomplish the following:

- [ ] Define the driver controls as they relate to robot software
- [ ] Describe the general flow of driver controls
- [ ] Create a plan for implementation

Summary

- Guiding principles:
  - Driver controls must use all mechanisms and subsystems on the robot to their fullest extent
  - Driver controls must be intuitive and easy to use for the driver
- List of mechanisms and subsystems on comp robot:
  - Differential drive chassis
  - Roller intake
  - Flywheel launcher
  - Pneumatic wings
- Chassis controls:
  - Differential drive can move forward and backwards as well as point turn
  - Three options:
    - Tank — left joystick controls left side of chassis, right joystick controls right side of chassis

        - Single stick arcade — either left or right joystick controls chassis movement, with vertical axis controlling forward/backward motion and horizontal axis controlling turning
        - Split arcade — one joystick controls forward/backward motion and one joystick controls turning
    - Due to driver preference and the lack of another mechanism that could require a joystick to control it, split arcade is the best option, as it is easy for the driver while allowing full range of motion
- Intake controls:
    - Roller intake can be off with no motion or it can spin forwards or backwards
    - Three options:
        - One button — toggles between all three states (off, forward, backward)
        - Two buttons — one toggles between forward and off, the other toggles between backward and off
        - Two buttons — one is held to spin forwards, one is held to spin backwards, and if neither are held, the intake is off
    - The third option will be used as it is easiest to implement in logic and does not require the driver to mentally keep track of the intake's state when toggling
- Launcher controls:
    - Flywheel launcher can have a range of speed from 0% to 100% of the motors' max speed. It can also index a triball to be launched
    - Four options:
        - One button is held to index triballs, another two are used to increase or decrease the flywheel speed incrementally
        - One button is held to index triballs, another one is used to toggle between two or more preset speeds
        - Two buttons are used to spin the indexer either forwards or backwards, another two are used to increase or decrease the flywheel speed incrementally
        - Two buttons are used to spin the indexer either forwards or backwards, another one is used to toggle between two or more preset speeds
    - The second option will be used — since the indexer will only need to feed triballs in one direction, only one button is necessary for it. Also, two to three preset flywheel

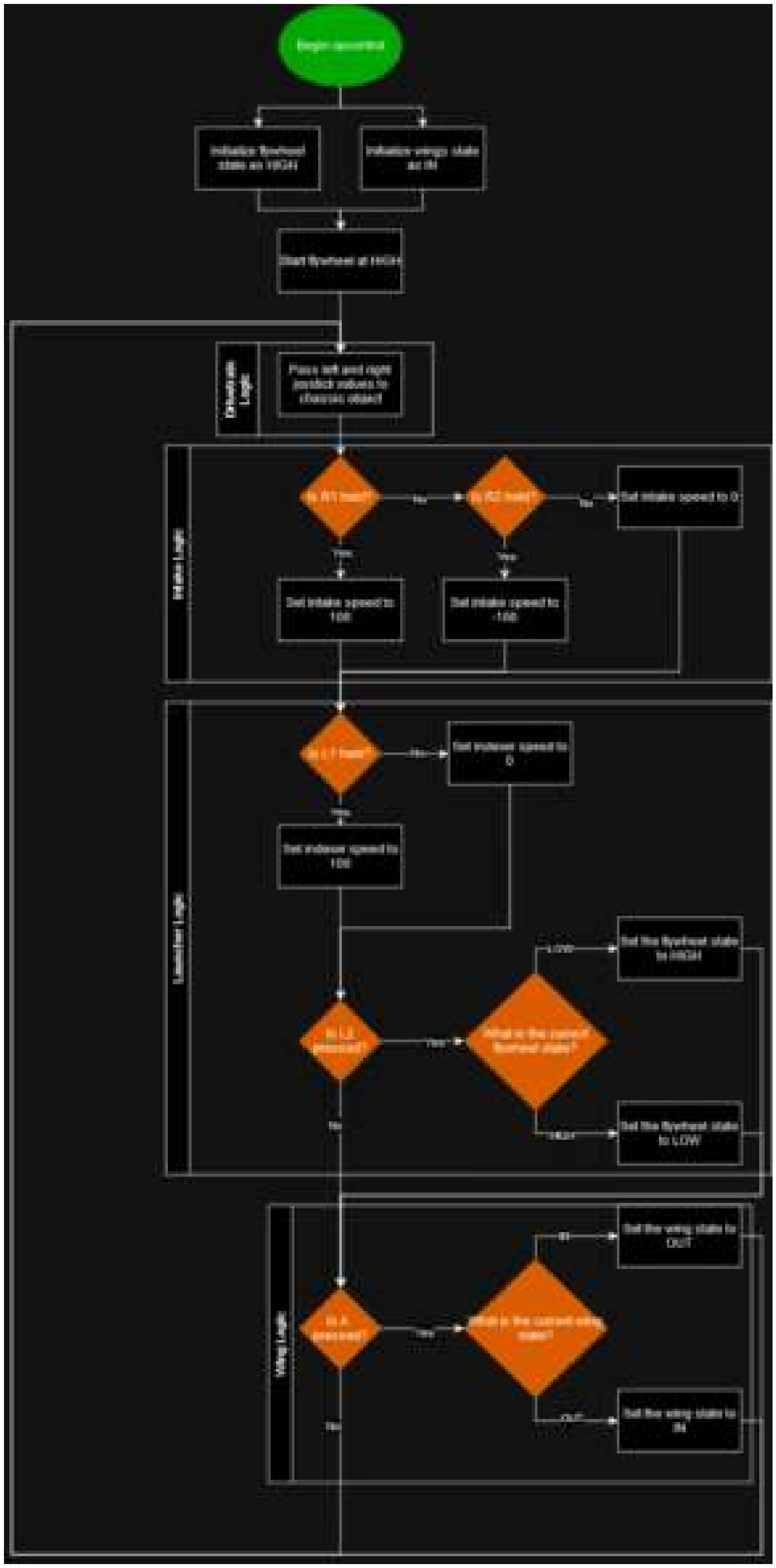
speeds are easier for a driver to use and practice with than having to precisely pick a “right” speed for each shot

- Wing controls:
  - Pneumatic wings have two states, deployed or retracted.
  - Two options:
    - One button toggles between the two wing states
    - Two buttons control the wing state, one always brings the wings to the deployed state and one always brings the wings to the retracted state
  - One button toggle will be used to make the code easier to implement and decrease mental load for the driver by having a dedicated “wings button”
- Preliminary controls:
  - Left joystick → Chassis forward/backward
  - Right joystick → Chassis turning
  - R1 → spin intake forwards
  - R2 → spin intake backwards
  - L1 → index triball into launcher
  - L2 → toggle flywheel speed
  - A → toggle wings
- Control flow (runs inside of the loop in opcontrol):
  - Drivetrain
    - Get joystick values and pass them to the chassis as input
  - Intake:
    - If R1 is held, spin the intake forwards
    - Else if R2 is held, spin the intake backwards
    - Else, do not spin the intake
  - Launcher:
    - If L1 is held, spin the indexer forwards
    - Else, do not spin the indexer
    - If L2 is pressed, toggle between the low and high flywheel speeds

- Wings:
    - If A is pressed, toggle between the deployed and retracted wing states
- State required for driver controls:
    - One variable keeps track of flywheel state (LOW or HIGH)
    - One variable keeps track of wing state (OUT or IN)

Flowchart

To assist in the driver control logic above, our team created the flow chart seen below.



Initialize flywheel state as HIGH
Initialize wings state as IN
Start flywheel at HIGH
Drivetrain Logic
Pass left and right joystick values to chassis object
Intake Logic
Set intake speed to 0
Set intake speed to 100
Set intake speed to -100
Launcher Logic
Set indexer speed to 0
Set indexer speed to 100
What is the current flywheel state?
LOW
Set the flywheel state to HIGH
HIGH
Set the flywheel state to LOW
Wing Logic
What is the current wing state?
IN
Set the wing state to OUT
OUT
Set the wing state to IN

Design and Building: Chassis

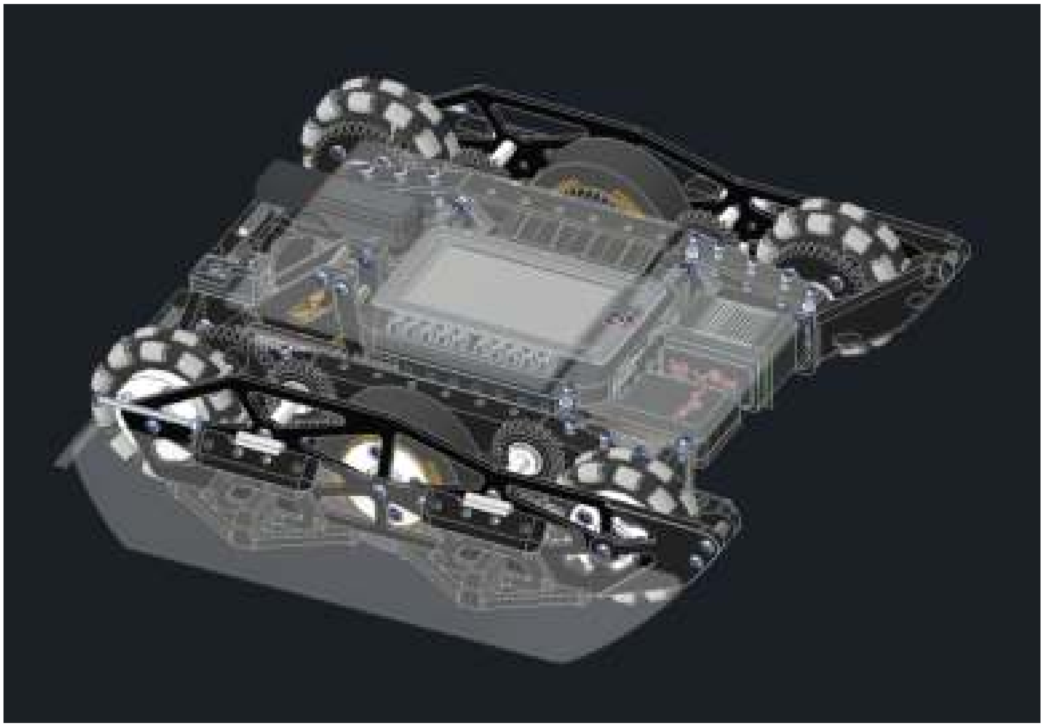
Date	@November 21, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Conner Siebert Sean MacDonald
Design Cycle	Design Cycle 1

Goals

- We will design the robot chassis using the Autodesk Inventor CAD package
- Using the model developed in the CAD we will build the chassis.

Design/Development Process

- Design considerations:
  - **Maneuverability.** Ability to cross over the center barrier and hang bars.
    - Our robot's outer dimensions are vital to ensuring these capabilities. Additionally, **the design of "sleds" on the front of our chassis aids in driving over the center barrier.**
  - **Gearing.** How our drives gear ratio aligns with desired match play.
    - We want a balanced drive with **enough speed to keep up with match play.**
  - **Weight.** We want to remove all unnecessary weight from our chassis so our maneuverability and speed are not adversely impacted.
    - Using mostly custom-fabricated components on our chassis can help us achieve this
  - **Fabrication.** We are designing components to allow for the most simple fabrication possible.
- Final Product
  - As decided using Decision Matrices and Brainstorming, **we have designed the chassis in CAD according to our previous constraints.**



- With the chassis modeled and planned, we can proceed to the build phase.

Build the Solution

To build the chassis, our team used the following steps:

1. To begin the chassis build we first assembled the traction wheel as seen in Figure 1. **We molded the outer silicon wheel and attached it to a 3D printer center hub.**
2. We then used custom adapters to mount plastic bearings to all 4 omni wheels on the chassis. **To insert the bearings into the adapters, we press fit them using a vice.**
3. Next, **we attached the traction wheels and omni wheels to the outer aluminum plates.** Both sides of the chassis were then attached

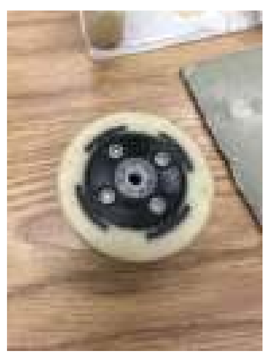


Figure 1: Assembled traction wheel

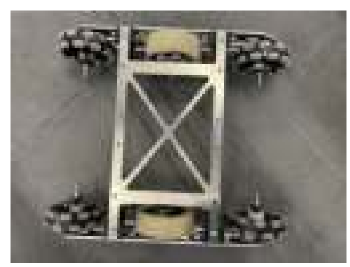


Figure 2: Chassis outer plates attached to the cross brace with wheels

to the baseplate as seen in Figure 2.

4. We then **mounted our 3D printed inner plates to each side of the chassis.** These plates provide mounting locations for our drive motors.
5. The sleds were then mounted to the base as seen in Figure 3. **To form the sleds, we used 2 3D prints mounted to the inner and outer plates of the chassis** which ramp the chassis over the middle barrier.
6. We then attached the skirt mounts to the base as seen in Figure 4. **The skirt mounts attach to the outer metal plate and will later support our polycarbonate side skirts.** Next we attached the gears to both sides of the drive as shown in Figure 5. We also attached motors and the brain during this step.
7. Acting as a final brace across the chassis, w**e attached our back metal plate** between the two inner drive channels.
8. We then added the rear wedge mounts to the chassis which are displayed in Figure 6. **The rear wedge mounts attach via the back metal plate and will later support our rear wedge** similarly to our side skirts.
9. Finally in Figure 7 we have completed the chassis assembly.

Figure 3: Front Sleds mounted to the outer chassis plate

Figure 4: Side skirts mounted to the outer chassis plate

Figure 6: Chassis rear cross brace with wedge mounts

Figure 5: Inner chassis plate mounted with drive gears and motors

Figure 7: Underside of completed Chassis

Test Solution

To ensure that the chassis meets our established goals, we will conduct testing in various areas.

The criteria we established for a successful chassis in the Brainstorming and Decision Matrix entry are:

- Pull force of at least 5 pounds
- Drive at a minimum of 300RPM

Drive Speed

- To meet the second criteria, **we designed the chassis with the speed of 375 RPM in mind.**
- In driving the chassis on the field, this seems to be quick enough for now, though **we may wish to increase the speed in the future.**

Pull Force

To test pull force, we use the hook scale attached to the back of the drive, with the drive moving away from the scale. **The specific procedure is outlined below:**

1. Attach the hook of the scale to the back of the chassis.
2. Position the chassis to be facing straight away from the scale.
3. Drive the chassis at full speed away from the scale.
4. Record the pull force as shown on the scale display.
5. Repeat four more times.

Results

	Pull Force (lbs)
Test 1	6.83
Test 2	7.12
Test 3	6.94
Test 4	6.27
Test 5	7.18
Average	6.87

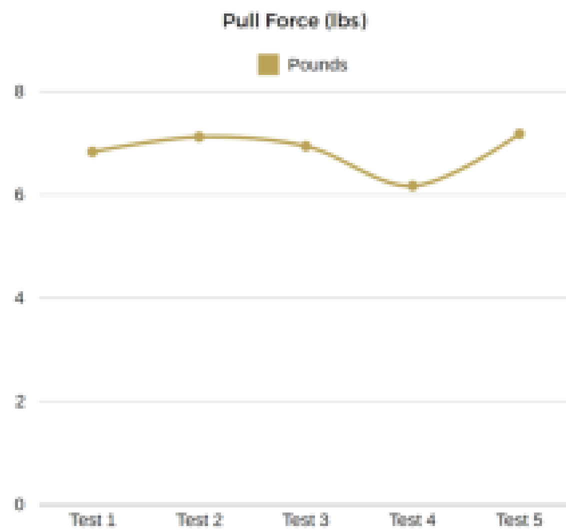


Figure 8: Testing chart

As the average pull force of the chassis exceeds our criteria by 1.87 pounds, **we can conclude that the chassis successfully meets both of our criteria.**

Comp Software: Driver Control Implementation

Date	@November 23, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Mihir Laud
Design Cycle	Design Cycle 1

Objectives

This meeting, we aimed to accomplish the following:

- [ ] Implement the driver control flowchart in code

Summary

- Drivetrain implementation:
    - This season, BLRS2 will be using VOSS, a library designed by our software team to make driver-control of the robot chassis and autonomous movements much simpler to use
    - The following snippet is the setup needed to run our 8-motor differential drive with PD control on linear and angular movement and odometry positioning.

```
 auto odom = voss::localizer::IMELocalizerBuilder::new_builder()
    .with_left_motors({-1,-3,-6,-5})
    .with_right_motors({11, 12, 19, 20})
    .with_track_width(9.75)
    .with_left_right_tpi(17)
    .build();

auto pid = voss::controller::PIDControllerBuilder::new_builder(odom)
        .with_linear_constants(4, 0, 4)
        .with_angular_constants(0.1, 0, 0.1)
        .with_tracking_kp(0.1)
```

```
        .with_exit_error(2)
        .with_angular_exit_error(0.1)
        .build();

auto chassis = voss::chassis::DiffChassis({-1,-3,-6,-5}, {11, 12, 19
```

- Inside of opcontrol, we are able to run our arcade controls with the following code. Note that the turn parameter is cubed and then scaled down to achieve a custom acceleration curve according to driver preference.

```
double throttle = master.get_analog(ANALOG_LEFT_Y);
double turn =  ((pow(master.get_analog(ANALOG_RIGHT_X), 3) * 0.0001)
chassis.arcade(throttle, turn);
```

- Intake implementation:
  - This code will allow the intake to run in both directions using only two buttons:

```
if (master.get_digital(DIGITAL_R1)) {
    intake::move(100);
} else if (master.get_digital(DIGITAL_R2)) {
    intake::move(-100);
} else {
    intake::move(0);
}
```

- Launcher implementation:
  - This snippet takes care of the indexer using the L1 button:

```
 if (master.get_digital(DIGITAL_L1)) {
    indexer::move(100);
} else {
    indexer::move(0);
}
```

  - This code facilitates the toggling of the flywheel between multiple speeds:

```
if (master.get_digital_new_press(DIGITAL_L2)) {
    if (flywheel_state == LOW) {
```

```
        flywheel_state = HIGH;
    } else {
        flywheel_state = LOW;
    }

    flywheel::toggle_speed(flywheel_state);
}
```

- Wings implementation:
  - This block of code lets the wings toggle after every A-button press:

```
if (master.get_digital_new_press(DIGITAL_A)) {
    if (wings_state == OUT) {
        wings_state = IN;
    } else {
        wings_state = OUT;
    }

    wings::toggle_state(wings_state);
}
```

Future Goals

Before the competition, we would like to accomplish the following:

- Test the code to make sure the logic works as intended
- Add in macros to make operating the robot easier for the drivers

Design and Building: Intake

Date	@November 27, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Brandon Liu J Jacob Zawacki
Design Cycle	Design Cycle 1

Goals

- We will design the intake assembly using Inventor
- Using the steps developed in the CAD we will build the the intake
- There are 2 main goals for this intake:
    - Match loading
        - **The robot should be able to pull Triballs out of the match load zone** and manipulate them without excess movement of the robot chassis itself.
        - The orientation of the robot should be like so:

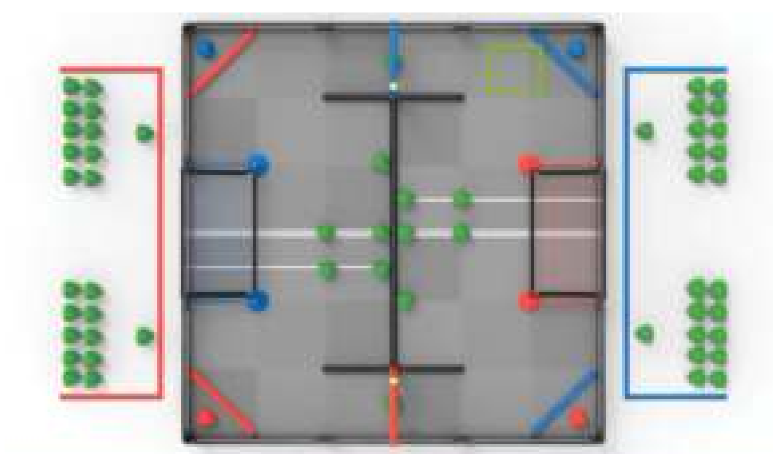


Figure 1: Orientation of Robot at the upper blue match load zone

- Fielding
    - **The robot should be able to "field" the Triball by quickly intaking the Triball into the robot** and keeping the Triball in possession in order to score loose Triballs during the driver controlled portion of a match.
- In addition to the above, **the intake should be light and durable** enough to withstand match play without breaking.

Design/Development Process

- Step 1
    - In order for the robot to match load effectively in the orientation listed above, **we measured that the intake would have to extend out of the robot by ~12 inches minimum (reach)** in order to be able to manipulate the Triball.
- Step 2
    - With the reach established, we can now design an intake geometry that will be able to achieve at least 12 inches of expansion. **We decided to use a 4-bar geometry for the intake since this type of mechanism is able to achieve the reach we need within a relatively small space.**

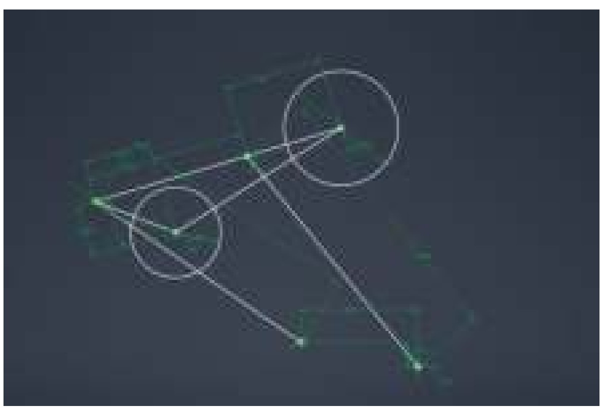


Figure 2: Intake Geometry in the Retracted Position

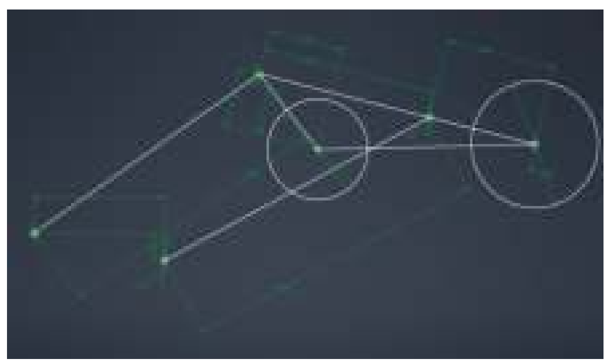


Figure 3: Intake Geometry in the Extended Position

- Step 3
    - Using this geometry we will now CAD our intake. **The current plan is to cut the bars out of polycarbonate (3/16" thick) and 3d print the top bar** that houses the intake rollers and the motor.

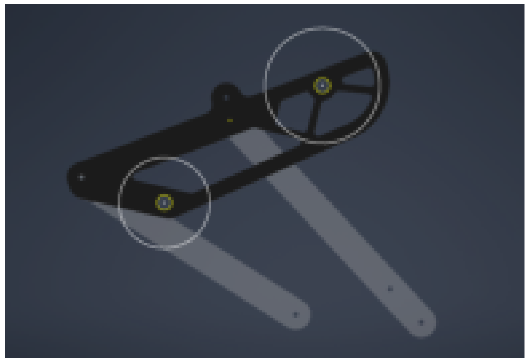


Figure 4: Version 1 of the intake

- Step 4
    - With the base of the intake, **we can now CAD the motor mount and the intake wheels to finish the intake**. The current plan is for both of the intake rollers to be spinning at 1200 rpm.

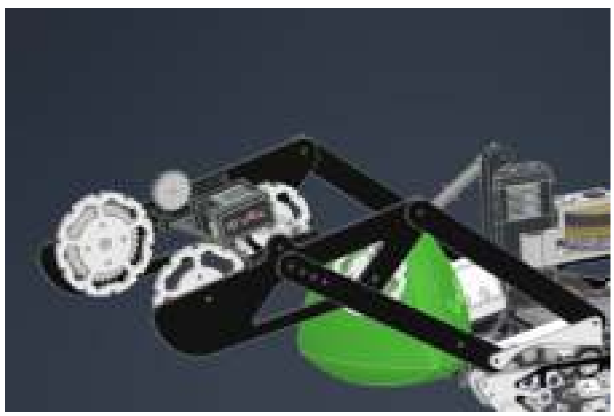


Figure 5: Side view of the intake on the robot

- Step 5
    - After further discussion, we have determined as a team to add a “3rd roller” which free floats behind the current 2 intake rollers in order to have a **more reliable passthrough mechanism** as well as a mechanical way of stopping the Triball from passing through the bot for the purpose of fielding.

Figure 6: Folded Intake with 3rd roller designed

Figure 7: Completed Intake CAD

- With the intake modeled and planned, we can proceed to build the solution.

Build the Solution

- Step 1
    - To build the intake, we had to **3d print the top bar and the 3rd roller bar along with the motor mounts, braces, and rollers** that are within the intake structure.

Figure 8: Completed 3D prints for the intake

- Step 2
    - Now we had to cut the 3/16″ polycarbonate to size and drill holes for the screws to pass through. The corners of the bar have to be filleted in order to be used in the intake mount that has been designed for the chassis. **We designed a jig that allows us to drill holes in the exact location necessary for the intake bars.**

Figure 9: Intake bar with drilling jig

- Step 3
    - Now that all the parts are prepared, we can assemble the intake according to the CAD model. The **polycarbonate bars are screw-jointed into the intake mounts** and should be able to freely move.

Figure 10: Intake mounts installed onto the chassis with intake arms

- Step 4
    - **Screw the top bars to the cross brace and screw the motor with the motor mount into the cross brace.** Some light sanding on the cross brace may be required for the parts to easily fit with each other.

Figure 11: Assembled Top Bar

Figure 12: Front roller and chain mounted to the intake

- Step 5
    - **Press the bearings into the intake top bars and mount the intake rollers** and spacers to the top bar using low strength axles and shaft collars. Attach and size the 6P chain between the rollers and the motor.

- Step 6
    - **Screw the 3rd roller bars to the cross brace** for the 3rd roller and attach the motor and motor mount to the cross brace/
- Step 7
    - **Attach the 3rd roller bars via screw joint to the intake bars and top bar.** Then screw joint the outer intake bar to the top bar.

Figure 13: 3rd Roller Mount installed onto the intake cross bars

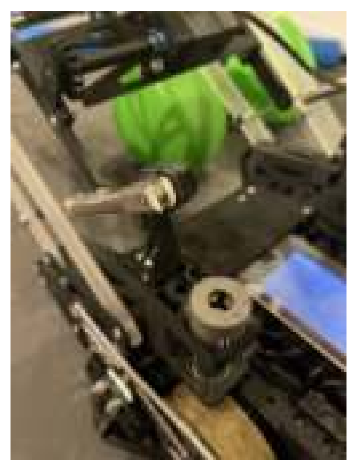


Figure 14: Intake Piston attached to the mount

- Step 8
    - Attach the roller to the 3rd roller bar and attach and size the 6p chain.
- Step 9
    - **Mount pistons on screw joints to the mounts** attached to the flywheel motor mount.
    - With this, the intake is complete (Figure 15).

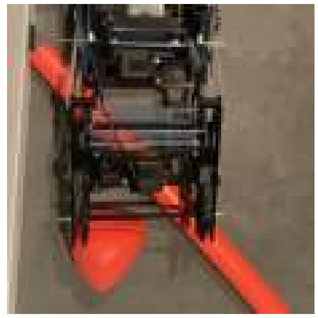


Figure 15: Top view of completed intake

Test Solution

To ensure that the intake meets our established goals, we will conduct testing in various areas.

The criteria we established for a successful intake in the Brainstorming and Decision Matrix entry are to manipulate Triballs from:

- Either Offensive Zone
- Any Match Load Zone

Offensive Zone

To test intaking from either Offensive Zone, **we used the following procedure:**

1. Place a Triball flat on the surface of the field tile, 12 inches away from the robot.
2. Drive the robot forward at full speed, with the intake spinning at maximum velocity.
3. Record whether the intake successfully fields the Triball into the passthrough location.
4. Repeat the process four additional times.

Results

	Success (Y/N)
Test 1	Y
Test 2	Y
Test 3	Y
Test 4	Y
Test 5	Y
Average	Y 5/5

Figure 16: Triball as it moves through the intake

As the intake was able to field Triballs from the Offensive Zone with no trouble, **we can conclude that the intake fulfills this set criteria.**

Match Load Zone

To test the ability to intake from the Match Load Zone, **we used the following procedure:**

1. Place a Triball flat on the surface of the field tile, 12 inches away from the robot.
2. Drive the robot forward at full speed, with the intake spinning at maximum velocity.
3. Record whether the intake successfully fields the Triball into the passthrough location.
4. Repeat the process four additional times.

Results

	Intake Time (seconds)
Test 1	1.83
Test 2	1.12
Test 3	1.94
Test 4	1.27
Test 5	1.18
Average	1.47

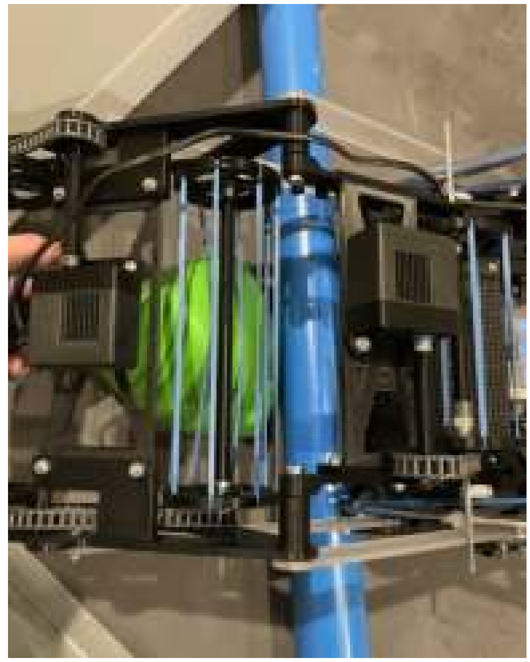


Figure 17: Tested Match Load Position

While **the intake was able to intake from the Match Load Zone successfully each trial**, the time for the Triball to reach the passthrough zone was inconsistent throughout. Additionally, we observed that **intaking in directions other than directly on to the bar was considerably more inconsistent**, sometimes failing.

The intake does fulfill the set criteria from earlier in the process, **making it a successful mechanism**. We do plan to revisit the intake after constructing the rest of the robot, to tune and improve the overall efficiency of intaking from the Match Load Zone.

Task and Timeline Update: December

Date	@December 4, 2023
Category	15" Robot 24" Robot BLRS2
Authors	Ⓒ Conner Siebert
Design Cycle	Design Cycle 1

November Recap

November saw major progress on both the robot design and building of the first robot

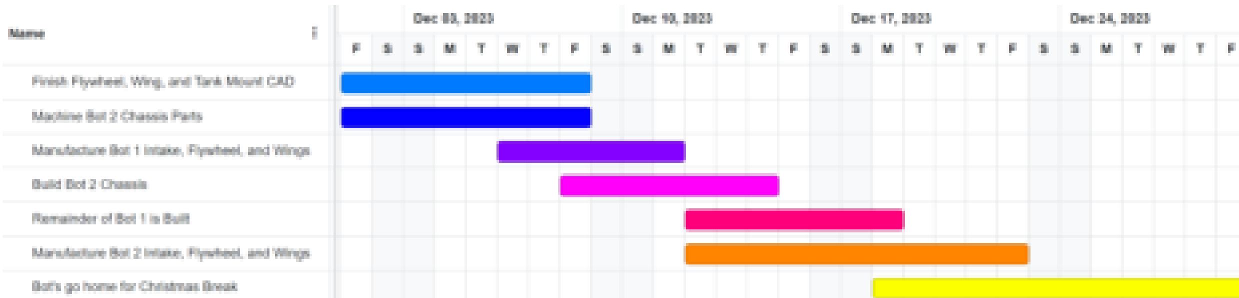
- Chassis and Intake CAD was completed
    - Through the first half of the month, the design for the chassis was completed
    - **After the chassis was completed, the intake followed by the end of the month**
- Chassis for bot 1 was constructed
    - With the chassis design completed, the **3D printed and aluminum pieces were fabricated** which allowed for the chassis for robot 1 to be competed
    - No issues were found with the chassis, allowing for future construction for bot 2
- Flywheel, tank mount, and wing CAD was began
    - With the chassis and intake design completed, the **ending of the month was spent beginning the last two major subsystems for this iteration of the robot**

December Task and Timeline

The overall goal of December is to **complete the building of both of the robots by the end of the month**. This will lead to a busy month for the competition robot team, because of the following factors:

- The CAD needs completed within the first week
    - To be able to build and test robot 1 before we leave campus, finishing CAD is a high priority

- All parts for the first robot should be manufactured **by the 11th of December**
- The goal is to **have robot one assembled before we leave campus for Christmas Break**
  - As we leave, Conner will take the first robot home for break to finish any building and tuning needed
  - Sean will take robot 2 from campus along with all manufactured parts, allowing for it to be **completed before we return to campus in January**



Decision Matrix and Building: Wings and Tank Mounts

Date	@December 11, 2023
Category	15" Robot 24" Robot BLRS2
Authors	J Joe C Conner Siebert
Design Cycle	Design Cycle 1

Goals

- One important aspect of the game this year is being able to **quickly and efficiently move large amounts of triballs to either score them into the goal, move them in autonomous, or push them over the center barrier.** With our wings, we want to be able to easily do all three of these actions with the wings being sturdy enough to withstand the large amount of force from pushing and driving aggressively.
- We will design an **effective wing that also doubles as our pneumatic tank mount** using prototyping, decision matrices, and CAD.
- Using the steps and designs developed in the prototyping and decision matrix phase, we will use CAD to design a fully flushed out and functional wing and tank mount that fully achieves our goals.
- When starting our brainstorming process, it was important to **define clear goals for what we wanted each component of the mechanism**, these specific goals are outlined here:
    - **Wings:**
        - Wings reach out as wide as possible without going out of size to push max amount of triballs
        - Wings are high enough that triballs do not flip over the wing
        - Wings be strong enough to withstand force from driving and other robots
        - Wings are able to fold in utilizing pneumatics
    - **Pneumatic Tank Mount:**
        - Securely holds tanks in place
        - Able to easily tube the system and replace if necessary
        - Be able to drive under the low bar on the field

Design/Development Process

- Wing Design Decisions
    - For the wings, we first had to decide the way in which we want the wings to fold out, **the three options being vertical, horizontal, or diagonal drop downs.**
    - These options are compared in the table below. For the horizontal wings, we used high school team 21417A as a basis for comparing possible designs.

Figure 1: Vertical Wing Concept

Figure 2: Angled Wing Concept

Figure 3: Horizonal Wing Concept from VRC team 21417A

Choosing The Best Option

Pros

Horizontal Wings	Vertical Wings	Angled Wings
- Low profile	- Easiest to build	- Allows wings to be longest distance
- Get long reach because no vertical limit	- No need for a locking mechanism	- Easier to store wings in size requirement
- Only requires 1 piston per wing	- Can make them wider and therefore stronger	
	- Only requires 1 piston per wing	
	- Can open if balls are in the way	

Cons

Horizontal Wings	Vertical Wings	Angled Wings
- Can struggle to open if many balls in way	Max length is limited by the horizontal bar on field	- Super complex fold out mechanism
- Requires locking mechanism when deployed	- If wings are tall, then they stick wider out on the bot when folded up	- Requires additional 2 pistons
- Gets in way with skirts		- Need locking mechanism when deployed
- Would need to attach to side of chassis somehow, which is difficult with custom side plates		

Decision Matrix

The following criteria were considered while we decided upon which launcher variation to use for our robots.

- Weight (x/5)
  - With this subsystem posing a risk of adding large amounts of additional weight to the robot, limiting this will be important
  - Pneumatic systems can become heavy, so minimizing pneumatics would be optimal
  - A higher score indicates a lighter system.
- Mechanical Complexity (x/5)
  - With such a large span of the wings, minimizing the additional risk of failure is important to us in the design oft his subsystem
  - A higher score indicates a simpler design
- Space Required (x/5)
  - As designing 15″ robot is often a crunch for space, smaller mechanisms tend to be favored to allow for other subsystems to be design around
  - A higher score indicates a more space efficient design
- Integration with other Subsystems (x/10)
  - While looking to attached the mount to the chassis, it is important to ensure that not too much real-estate is removed from the chassis to mount other subsystems
  - A higher score indicates that the design integrates well with other subsystems
- Integration of the tank into the mount (x/10)
  - With the two tanks being fairly large, using them as structural support for the mount would be ideal.
  - As the wings folds into the robot, the tanks should run along with them, as together in the same configuration less space will be required
  - A higher score indicates that the wings integrate well with the tank mount

	Weight (x/5)	Mechanical Complexity (x/5)	Space Required (x/5)	Integration with other Subsystems (x/10)	Integration of the tank into the mount (x/10)	Total (x/35)
Horizontal Wings	5	3	3	1	5	17
Vertical Wings	5	5	4	8	8	30
Angled Wings	3	1	2	5	4	15

- After team discussion and comparing the pros and cons with the overall goals for the system, the team decided to go with the **vertical wings, as it would achieve all the laid out goals for the wings and be the simplest and strongest to build.**
    - For the tank mount portion, we concluded that the **tanks being vertical** would be the best option as it would provide support to the wing mount while still fitting under the horizontal field bar.
    - Additionally, for the tank, the team wanted to try out **lightweight pneumatic tanks which are smaller but 1/4th the weight of a typical VEX pneumatic tank**, which were ordered off a third party website.

Implementation Plan

- Designing the part in CAD
    - To design the part, **the mounting point would be as far back as possible and mounting overtop the wheel**, screwing into the handful of holes that are laid out for the wheels on the side chassis piece.
    - For the wing, we want it **as low to the ground as possible**, therefore it was decided that it sits further out closer to the outside of the wheel to be lower to the ground when folded out, thus making the wings less tall.
    - To make the wings sturdy, we made them fairly thick, and **added support braces** in between which allowed for strong but light wings.
    - To power the wings up and down, a **piston mount was integrated into the vertical tank mount** and connected to the wing such that the max stroke of the piston is used.
    - For the pneumatic tank, we needed it to be **as low to the ground in order to fit under the horizontal bar.**
    - In order to make the piece printable, we decided to make a screw on cap which **makes 3D printing the part possible.**
- Final Product
    - With this design, and its mirrored counterpart, **we will 3D print them and be able to easily assemble the pieces together onto the bot.** The prints include the mount, the wing, tank caps, and piston connectors.

Build the Solution

- Step 1
  - After the design was completed, **all 3D printed parts were manufactured.** The main mounts were the largest pieces to 3D print, with over 300g of filament per mount with print supports!

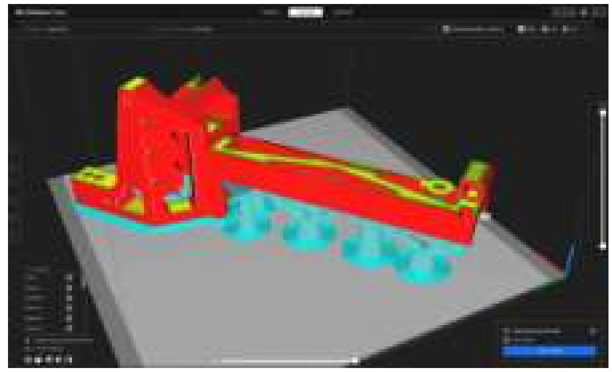


Figure 4: Wing and tank mount in the 3D printer slicer

- Step 2
  - After the mounts were printed, they were attached to the chassis.
  - To do this, **3 screws are attached to the top of the inner chassis plate, and two onto the back side**, also passing though the back chassis brace plate.

Figure 5: Tank and wing mount mounted to the inner chassis piece

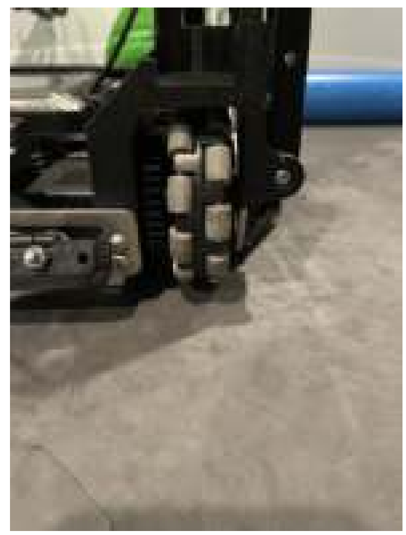


Figure 6: Rear mounting screws passing through the chassis brace plate

- Step 3
    - Assemble the wings onto the mounts with the pistons.
    - To attached the wings, a **2.5″ screw is used as a screw joint,** passing though the pivot on the mount and the wings.
    - To ensure the wing will move frictionless, a **drill is used to remove any excess PLA** from the wing mounting holes.
    - The piston is attached to the mounting hole on the wing **using a screw joint**, in the same manner as the pivot.

Figure 7: Extended Wing on the screw joint

Figure 8: Side View of folded wing

- Step 4
    - Attach tanks and the tank caps.
    - The aluminum tanks are **placed into the bottom of the mounts.**
    - The cap is screwed onto the top of the tank mount, **holding the tank in place.**

Figure 9: Completed wing and tank mount

Test Solution

In order to test the wings and ensure that they deploy without issue as established in our initial criteria, we will use the following procedure:

1. Ensure that the wings are in the upright position as shown in Figure 9.
2. Using the brain screen control panel, activate the pneumatics controlling the wings.
3. Record whether or not the wings deployed.
    a. A successful deploy will have the wings in the horizontal position instantaneously. This is shown in Figure 10.
4. Repeat four more times.

Trial Number	Success/Failure
1	Success
2	Success
3	Success
4	Success
5	Success

Figure 10: Successful wing deploy

Observations and Results:

- As shown in our testing, the **wings successfully deploy using pneumatics to their maximum width, fulfilling the initial set criteria.** Using the 3D-printing settings described in the build transcript, the wings are sturdy and will not break under threat of defensive play.
    - Through multiple iterations and print tests, the best print/support settings were found that **wasted the least amount of filament while still providing the structural integrity that was needed.** Further, we optimized the print time from 18 hours all the way down to 8 for the large mount print.
- In driving around the field, it was seen that the tanks we bought were not super durable, and dented easily. To combat this, we have designed an **additional set of these mounts to fit more sturdy tanks which will be implemented in the future.**

Task and Timeline Update: January Before NUKETown

Date	@January 8, 2024
Category	15" Robot 24" Robot BLRS2 WIP
Authors	Conner Siebert Brandon Liu
Design Cycle	Design Cycle 1

December Recap

The strong push from the competition team members in December paid off, as both robots were assembled for NUKE over the break.

- All Design work was completed after the first
- Manufacturing of all parts for both robots was completed
  - Over a total of 300 3D prints were used total between both competition robots!
- Robot 1 was mostly completed on campus
  - Conner was able to complete the robot before returning to campus on January 7th
- Robot 2 was machined on campus
  - Sean finished robot 2 at home, also returning to campus with it at the end of break

January Task and Timeline

The biggest date in January will be attending the NUKE competition on January 14th. In preparation, the following will occur during the week of January 8th:

- Driver practice
  - Complete skills runs
  - Go head to head against our sister team, BLRS
- Programming
  - Create any driver macros required
  - Write 45 second match autonomous

    - Tune the 60 second programming skills
- Final Tuning
    - Throughout the week of preparation, we expect several aspects of the robot to need small changes

After NUKE, we plan to host a team meeting to discuss the results, and to brainstorm potential changes to the robots for the remainder of the month.

Comp Software: NUKE Match Autonomous

Date	@January 9, 2024
Category	15" Robot, 24" Robot, BLRS2
Authors	Ⓜ Mihir Laud Ⓐ Alex Lam Ⓜ Max
Design Cycle	Design Cycle 1

Objectives

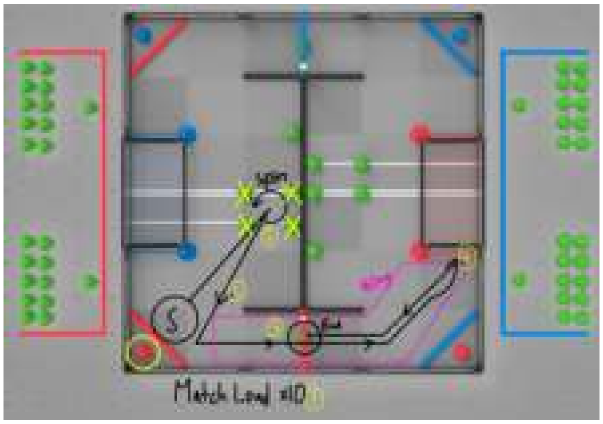
- [ ] Plan match Autonomous for both 15″ & 24″ robots
- [ ] Create flowcharts of autonomous program(s)
- [ ] Implement autonomous code

Planning

The number 1 goal that we have for our match autonomous for each robot is the ability to **score a consistent Autonomous Win Point (AWP)**. This will help us the most throughout the day as we aim to place high in the qualification rankings.

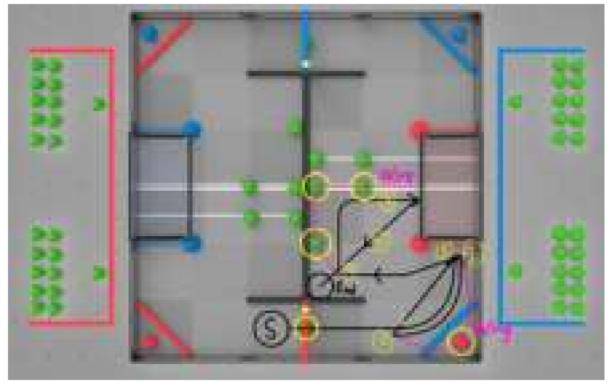
To complete this, both robots will be **responsible for scoring an Alliance Triball in the goal, as well as touching the hanging bars** at the end of the autonomous period.

Before programming and testing the autonomous on the robots, a route is **planned and mapped (as seen below):**



24″ Robot Match Autonomous Planning Sketch

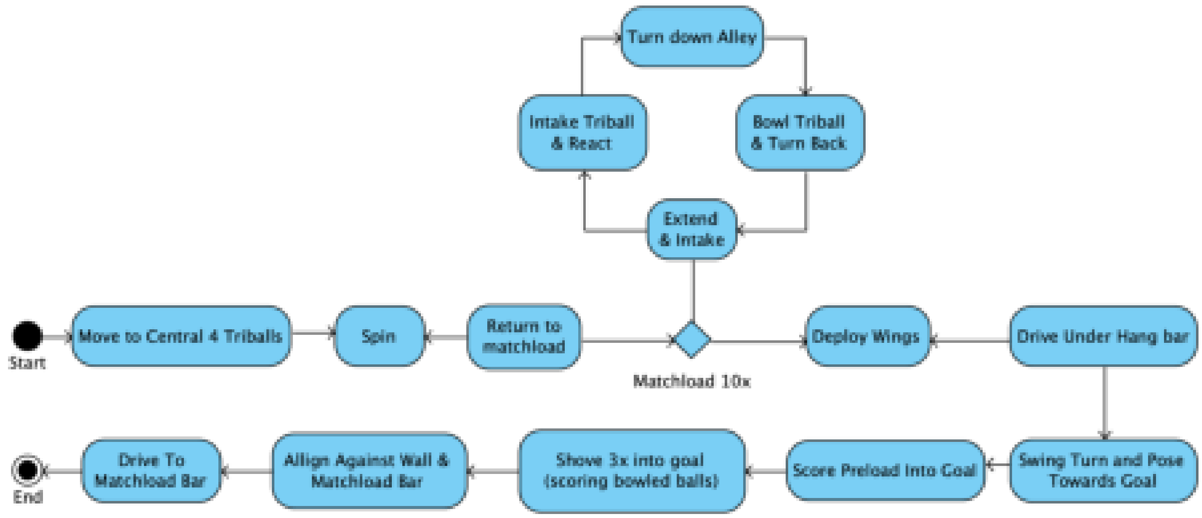
Step of Autonomous	Procedure
Step 1	Move to the center of the field, deploy wings, and spin to grief Triballs so the opponent can’t score them
Step 2	Move into a position to begin match loading.
Step 3	Match load 10 Triballs (&preload) using our pass through.
Step 4	Sweep Triballs into the side of the goal.
Step 5	Move to the horizontal hanging bar to end touching it and finish AWP.



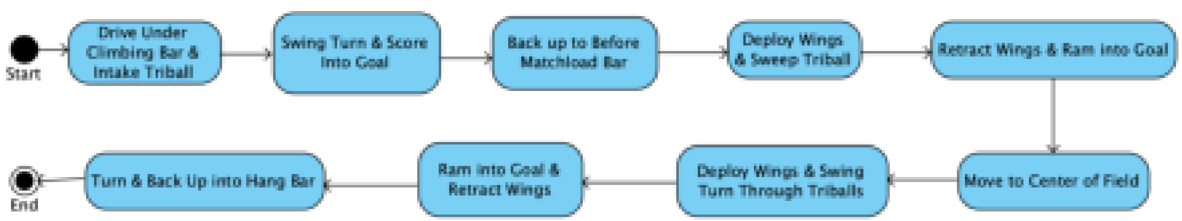
15″ Robot Match Autonomous Planning Sketch

Step of Autonomous	Procedure
Step 1	Drive under the hanging bar and score 2 Triballs into the goal.
Step 2	Back up, deploy the right wing, and sweep the alliance Triball into the goal
Step 3	Ram into the goal forcing Triballs further in.
Step 3/4	Move to the middle of the field, deploy wings, and sweep 3 triballs into the goal.
Step 5	Move towards hanging bar to end touching it and finish AWP.

With this plan, we will be able to score 6 Triballs in the goal, as well as 10 match loads into the offensive zone, and end with both robots touching the hanging bars. **This could potentially score us 50 points during the autonomous period.** A flowchart was created for both robots to convert the route to pseudocode (code procedure):



24" Match Autonomous Flow Chart



15" Match Autonomous Flow Chart

Now that we know the process we need to implement into the code for each of the robots, we can begin to program. The **motors and sensors are first initialized within the code.** Following this, the ARMS library will be used for movement through the use of odometry & PID.

Mechanism/System Powered (15")	Ports Initialized
Left Drive Motors	2, 3, 5, 6
Right Drive Motors	11, 12, 19, 20
IMU (inertial measurement unit) Sensor	13

Mechanism/System Powered (24")	Ports Initialized
Left Drive Motors	1, 3, 4, 21
Right Drive Motors	11, 14, 19, 20
IMU (inertial measurement unit) Sensor	15

Encoder Sensor	5, 5

Below is the code utilized to initialize the motors and sensors utilized for the robots, as well as the constant values utilized for PID:

```
// Negative numbers mean reversed motor
#define LEFT_MOTORS -2, -3, -6, -5 // left drive motors
#define RIGHT_MOTORS 11, 12, 19, 20 // right drive motors
#define GEARSET pros::E_MOTOR_GEAR_600 // RPM of chassis motors

// Ticks per inch
#define TPI 45.7   // Encoder ticks per inch of forward robot movement
#define MIDDLE_TPI 1 // Ticks per inch for the middle wheel

// Tracking wheel distances
#define TRACK_WIDTH 9.75 // The distance between left and right wheels (or tracker wheels)
#define MIDDLE_DISTANCE 0 // Distance from middle wheel to the robot turning center

// Sensors
#define IMU_PORT 13                          // Port 0 for disabled
#define ENCODER_PORTS 0, 0, 0              // Port 0 for disabled,
#define EXPANDER_PORT 0                      // Port 0 for disabled
#define ENCODER_TYPE arms::odom::ENCODER_ADI // The type of encoders

// Movement tuning
#define SLEW_STEP 8             // Smaller number = more slew
#define LINEAR_EXIT_ERROR 0.3   // default exit distance for linear movements
#define ANGULAR_EXIT_ERROR 1    // default exit distance for angular movements
#define SETTLE_THRESH_LINEAR .5 // amount of linear movement for settling
#define SETTLE_THRESH_ANGULAR 1 // amount of angular movement for settling
#define SETTLE_TIME 150         // amount of time to count as settled
#define LINEAR_KP 6 // Linear Proportion constant for PID
#define LINEAR_KI 0 // Linear Integral constant for PID
#define LINEAR_KD 50 // Linear Derivative constant for PID
#define TRACKING_KP 40 // point tracking turning strength
#define TRACKING_KD 200 // point tracking turning strength
#define ANGULAR_KP 1.25 // Angular Proportion constant for PID
#define ANGULAR_KI 0.01 // Angular Integral constant for PID
#define ANGULAR_KD 8 // Angular Derivative constant for PID
#define MIN_ERROR 5 // Minimum distance to target before angular componenet is disabled
#define LEAD_PCT .6 // Go-to-pose lead distance ratio (0-1)

#define MIN_LINEAR_SPEED 5 // lowest linear speed in which robot should go
#define MIN_ANGULAR_SPEED 5 // lowest angular speed in which robot should go
```

Figure 3: Motor & Sensor Initialization + Control Constants for 15″ Robot

```
// Negative numbers mean reversed motor
#define LEFT_MOTORS -4, -21, -1, -3 // left drive motors
#define RIGHT_MOTORS 11, 14, 19, 20 // right drive motors
#define GEARSET pros::E_MOTOR_GEAR_600 // RPM of chassis motors

// Ticks per inch
#define TPI 323.33   // Encoder ticks per inch of forward robot movement
#define MIDDLE_TPI 1 // Ticks per inch for the middle wheel

// Tracking wheel distances
#define TRACK_WIDTH 9.75 // The distance between left and right wheels (or tracker wheels)
#define MIDDLE_DISTANCE 0 // Distance from middle wheel to the robot turning center

// Sensors
#define IMU_PORT 15                         // Port 0 for disabled
#define ENCODER_PORTS 5, 5, 0               // Port 0 for disabled,
#define EXPANDER_PORT 0                     // Port 0 for disabled
#define ENCODER_TYPE arms::odom::ENCODER_ADI // The type of encoders

// Movement tuning
#define SLEW_STEP 8              // Smaller number = more slew
#define LINEAR_EXIT_ERROR 0.2    // default exit distance for linear movements
#define ANGULAR_EXIT_ERROR 1     // default exit distance for angular movements
#define SETTLE_THRESH_LINEAR .5 // amount of linear movement for settling
#define SETTLE_THRESH_ANGULAR 1 // amount of angular movement for settling
#define SETTLE_TIME 250          // amount of time to count as settled
#define LINEAR_KP 4 // Linear Proportion constant for PID
#define LINEAR_KI 0 // Linear Intergral constant for PID
#define LINEAR_KD 12 // Linear Derivative constant for PID
#define TRACKING_KP 60 // point tracking turning strength
#define ANGULAR_KP 2.3 // Angular Proportion constant for PID
#define ANGULAR_KI 0 // Angular Integral constant for PID
#define ANGULAR_KD 20 // Angular Derivative constant for PID
#define MIN_ERROR 5 // Minimum distance to target before angular componenet is disabled
#define LEAD_PCT .6 // Go-to-pose lead distance ratio (0-1)
```

Figure 4: Motor & Sensor Initialization + Control Constants for 24″ Robot

Specific and key coding sections such as griefing autonomous, autonomous loading, and goal scoring for the match autonomous can be found below:

```cpp
// grief 4 balls on the center line. Wings out, and pop balls over barrier
// with wings/back plate
chassis::move(-16, 90, 2, arms::THRU | arms::REVERSE); // move towards triball
chassis::move({18, 44.5}, 75, 3, arms::REVERSE); // move towards triball
wings::toggle(); // activate wings to grief triballs
chassis::turn(-180, 75); //face towards the triballs
chassis::move({48, 47}, 90, arms::REVERSE); // push triballs over bar
chassis::move(6, 80); // back up slightly from the center bar
chassis::turn(-180); // line up to score triball in goal
wings::toggle(); // retract wings after grief
```

Figure 5: Match grief autonomous for 15″ Robot

```cpp
for (int i = 0; i < MATCH_LOAD_NUM; i++) { // loop to match load in autonomous (run till 10 triballs are introduced)
    intake::move(100); // intake triball to grab out of load zone
    intake::move(0); // stop intake as inside robot
    chassis::move(-6, 100, arms::REVERSE); // backup slightly towards alliance park pole
    chassis::turn(-210, 50); // turn robot towards alley
    intake::move(100); // drop triball onto field
    indexer::move(100); // drop triball onto field
    while (!chassis::settled()) { // check if chassis has stopped moving
        pros::delay(10); // if chassis is still moving, delay 10 ms
    }
    pros::delay(550);
    indexer::move(0); // stop rotation of the intake
    chassis::turn(-136, 50); // realign with load zone
    intake::move(100); // run intake to grab triball
    chassis::move(6, 80); // move back into load zone
}
```

Figure 6: Match autonomous loading autonomous for 15″ Robot

```cpp
// score the triball
wings::toggle(); // retract wings
pros::delay(300);
chassis::turn(225); // turn robot around
chassis::move(-20, arms::REVERSE); // line up with the wall
chassis::turn(270); // line up to prepare to push

for (int i = 0; i < SLAM_AND_JAM; i++) { // repeat twice to push triballs in
    chassis::move(-18, arms::REVERSE); // move backwards to push triball
    chassis::move(18); // move forward to prepare for another push
}
```

Figure 7: Match scoring into goal autonomous for 24″ Robot

Comp Software: NUKE Skills Autonomous

Date	@January 11, 2024
Category	15" Robot, 24" Robot, BLRS2
Authors	Mihir Laud, Alex Lam
Design Cycle	Design Cycle 1

Objectives

- [ ] Plan Skills Autonomous for both 15″ & 24″ robots
- [ ] Create flowcharts for autonomous program(s)
- [ ] Implement autonomous code

Planning

We would like to test 2 approaches to the skills challenge at this tournament. With the 24″ robot, we plan to **use the pass-through to score match loads down under the hanging bar to the offensive zone.** With the 15″ robot, we plan to **use the wings to score match loads down under the hanging bar to the offensive zone.** Both robots will then use their barrier hangs at the end although we don't anticipate having time to tune this before the competition and don't believe it will be necessary for the programming skills challenge.

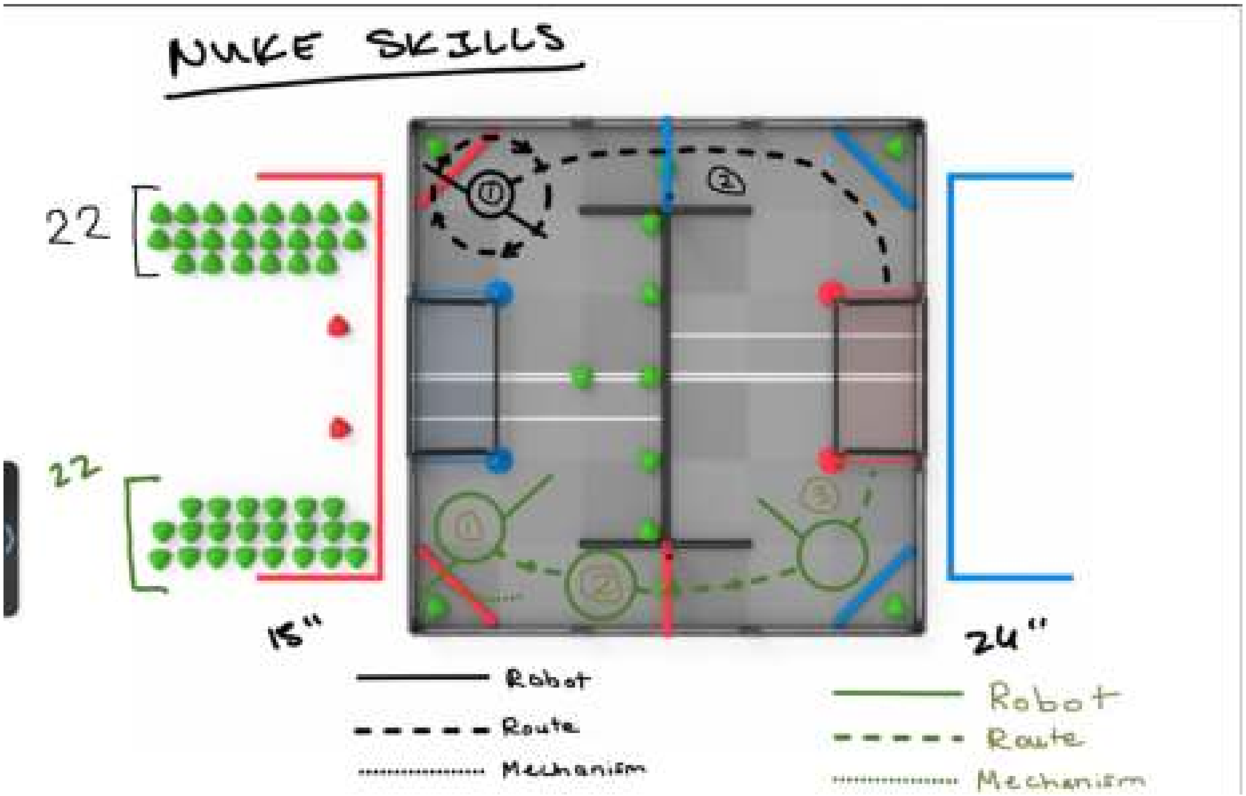


Figure 1: Skills Autonomous Route for Both Robots (15″ & 24″)

With this plan, we will be able to introduce 44 balls onto the field, and **bowl most of the triballs across the field.** We will also attempt to score some of the triballs into the goal, with a guarantee of 2 alliance triballs that will be scored into the goal.

Following the mapped plan, a flowchart was created for both robots to convert the route to pseudocode (code procedure):

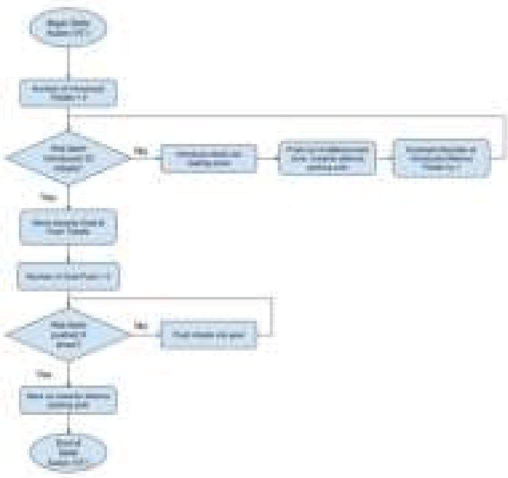


Figure 2a: Flowchart for Skills Autonomous Route for 15'' Robot

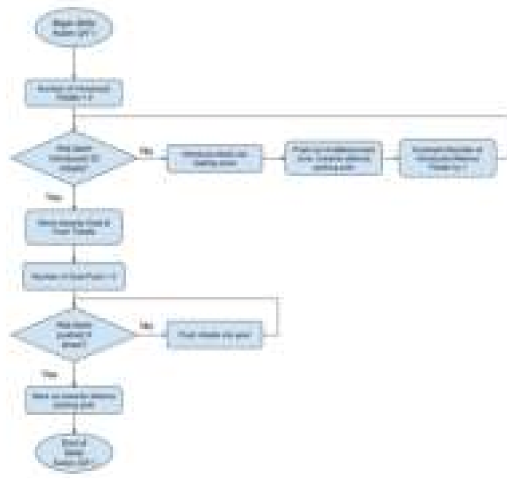


Figure 2b: Flowchart for Skills Autonomous Route for 24'' Robot

Following the flowcharts, implementation of the logic was applied to the robot for competition use. **Initialization of the motors was first done before odometry & PID was utilized to program the robot movements of the route.**

Mechanism/System Powered (15'')	Ports Initialized
Left Drive Motors	2, 3, 5, 6
Right Drive Motors	11, 12, 19, 20
IMU (inertial measurement unit) Sensor	13

Mechanism/System Powered (24'')	Ports Initialized
Left Drive Motors	1, 3, 4, 21
Right Drive Motors	11, 14, 19, 20
IMU (inertial measurement unit) Sensor	15
Encoder Sensor	5, 6

```
// Negative numbers mean reversed motor
#define LEFT_MOTORS -2, -3, -6, -5 // left drive motors
#define RIGHT_MOTORS 11, 12, 19, 20 // right drive motors
#define GEARSET pros::E_MOTOR_GEAR_600 // RPM of chassis motors

// Ticks per inch
#define TPI 45.7   // Encoder ticks per inch of forward robot movement
#define MIDDLE_TPI 1 // Ticks per inch for the middle wheel

// Tracking wheel distances
#define TRACK_WIDTH 9.75 // The distance between left and right wheels (or tracker wheels)
#define MIDDLE_DISTANCE 0 // Distance from middle wheel to the robot turning center

// Sensors
#define IMU_PORT 13                          // Port 0 for disabled
#define ENCODER_PORTS 0, 0, 0              // Port 0 for disabled,
#define EXPANDER_PORT 0                      // Port 0 for disabled
#define ENCODER_TYPE arms::odom::ENCODER_ADI // The type of encoders

// Movement tuning
#define SLEW_STEP 8             // Smaller number = more slew
#define LINEAR_EXIT_ERROR 0.3   // default exit distance for linear movements
#define ANGULAR_EXIT_ERROR 1    // default exit distance for angular movements
#define SETTLE_THRESH_LINEAR .5 // amount of linear movement for settling
#define SETTLE_THRESH_ANGULAR 1 // amount of angular movement for settling
#define SETTLE_TIME 150         // amount of time to count as settled
#define LINEAR_KP 6 // Linear Proportion constant for PID
#define LINEAR_KI 0 // Linear Integral constant for PID
#define LINEAR_KD 50 // Linear Derivative constant for PID
#define TRACKING_KP 40 // point tracking turning strength
#define TRACKING_KD 200 // point tracking turning strength
#define ANGULAR_KP 1.25 // Angular Proportion constant for PID
#define ANGULAR_KI 0.01 // Angular Integral constant for PID
#define ANGULAR_KD 8 // Angular Derivative constant for PID
#define MIN_ERROR 5 // Minimum distance to target before angular componenet is disabled
#define LEAD_PCT .6 // Go-to-pose lead distance ratio (0-1)

#define MIN_LINEAR_SPEED 5 // lowest linear speed in which robot should go
#define MIN_ANGULAR_SPEED 5 // lowest angular speed in which robot should go
```

Figure 3: Motor & Sensor Initialization + Control Constants for 15′′ Robot

```
// Negative numbers mean reversed motor
#define LEFT_MOTORS -4, -21, -1, -3 // left drive motors
#define RIGHT_MOTORS 11, 14, 19, 20 // right drive motors
#define GEARSET pros::E_MOTOR_GEAR_600 // RPM of chassis motors

// Ticks per inch
#define TPI 323.33   // Encoder ticks per inch of forward robot movement
#define MIDDLE_TPI 1 // Ticks per inch for the middle wheel

// Tracking wheel distances
#define TRACK_WIDTH 9.75 // The distance between left and right wheels (or tracker wheels)
#define MIDDLE_DISTANCE 0 // Distance from middle wheel to the robot turning center

// Sensors
#define IMU_PORT 15                            // Port 0 for disabled
#define ENCODER_PORTS 5, 5, 0                  // Port 0 for disabled,
#define EXPANDER_PORT 0                        // Port 0 for disabled
#define ENCODER_TYPE arms::odom::ENCODER_ADI // The type of encoders

// Movement tuning
#define SLEW_STEP 8              // Smaller number = more slew
#define LINEAR_EXIT_ERROR 0.2    // default exit distance for linear movements
#define ANGULAR_EXIT_ERROR 1     // default exit distance for angular movements
#define SETTLE_THRESH_LINEAR .5 // amount of linear movement for settling
#define SETTLE_THRESH_ANGULAR 1 // amount of angular movement for settling
#define SETTLE_TIME 250          // amount of time to count as settled
#define LINEAR_KP 4 // Linear Proportion constant for PID
#define LINEAR_KI 0 // Linear Intergral constant for PID
#define LINEAR_KD 12 // Linear Derivative constant for PID
#define TRACKING_KP 60 // point tracking turning strength
#define ANGULAR_KP 2.3 // Angular Proportion constant for PID
#define ANGULAR_KI 0 // Angular Integral constant for PID
#define ANGULAR_KD 20 // Angular Derivative constant for PID
#define MIN_ERROR 5 // Minimum distance to target before angular componenet is disabled
#define LEAD_PCT .6 // Go-to-pose lead distance ratio (0-1)
```

Figure 4: Motor & Sensor Initialization + Control Constants for 24″ Robot

Specific and key coding sections such as “Bowling” and pushing of triballs into the goal for the skills autonomous can be found below:

```cpp
// bowl across the 22 triball load
for (int i = 0; i < MATCH_LOAD_NUM_SKILLS; i++) { // run till all 22 triballs have been introduced
    intake::move(100); // intake triball into robot
    pros::delay(100);
    chassis::move(-6, 100, 3, arms::REVERSE); // back up from load zone
    chassis::turn(-203, 100); // turn robot towards alley
    intake::move(100); // drop triball onto field
    indexer::move(100); // drop triball onto field
    pros::delay(600);
    indexer::move(0); // stop rotation of the intake
    chassis::turn(-135.5, 100); // face back towards load zone
    intake::move(100); // run intake to grab triball
    chassis::move(6, 80, 3); // move back into load zone
}

intake::move(100); // intake to prevent triballs from reentering robot
pros::delay(250);
chassis::move(-6, 100, arms::REVERSE); // back up from load zone
chassis::turn(-195, 75, arms::ASYNC); // turn robot towards alley
pros::delay(50);
intake::move(100); // intake to prevent triballs from reentering robot
indexer::move(100); // intake to prevent triballs from reentering robot
pros::delay(550);
indexer::move(0); // stop rotation of the intake

// try and shove them in goal
intake::toggle_piston(); // activate wings
chassis::move({20, -3}, 100, 3, arms::THRU | arms::REVERSE); // push triball by driving through the alley
chassis::move({50, -5, -180}, 90, arms::REVERSE); // align with goal and push
chassis::move({20, -5, -180}, 90); // align with goal and push
pros::delay(250);
```

Figure 5: “Bowling” & Scoring of triballs into goal for 15′′ Robot

```cpp
// bowl across the 22 triball load
wings::toggle(wings::BOTH); // activate wings
chassis::move(16); // move towards loading zone

chassis::tank(30, -30); // spin in circle to knock triballs from loading zone
pros::delay(45000); // run the above spinning motion for 45000 ms (45 seconds) to load all 22 triballs
chassis::tank(0, 0); // stop movement of robot
wings::toggle(wings::BOTH); // retract wings
pros::delay(1000);
chassis::turn(340, 50); // rotate to align with alley
chassis::move(-20, 80, arms::REVERSE); // push into pile of triballs to "bowl"
chassis::turn(320, 50); // rotate again to align with alley
chassis::move(-72, 40, arms::REVERSE); // push into pile of triballs to "bowl"
intake::move(100); // intake to prevent triballs from reentering robot
chassis::move(24); // move back to prevent robot from touching triballs
```

Figure 6: “Bowling” & Scoring of triballs for 24′′ Robot

NUKETown 2024 VEXU Qualifier Results

Date	@January 14, 2024
Category	15" Robot 24" Robot BLRS2
Authors	K Kathleen Lowe A Alex Lam
Design Cycle	Design Cycle 1

Premise

Figure 1: Team Picture

On January 14th 2024, we attended the NUKEtown 2024 VEXU Qualifier in Highland Heights, Kentucky, which had a total of 10 teams competing.

Performance Overview

Matches

Rank	3	Record	4-2-0
WP	8	CCWM	36.2
AP	40	OPR	71.8
SP	242	DPR	35.7

Outcome: Eliminated in quarterfinals, Design Award

Q3 - LOSS	Score
BLRS2	52
BLRS	67

Q6 - LOSS	Score	Q12 - WIN	Score
BLRS2	58	BV1	98
WLDCT	70	KUDOS	13

The first two matches were the first times that our drivers had the time to learn to use the robots hence our losses. They were able to slowly get the hang of it though and practice on a separate field. Our first match also ended with the 15″ robot being beached on the barrier.

Q19 - WIN	Score
BLRS2	75
NUKE	31

Q23 - WIN	Score	Q30 - WIN	Score
BLRS2	93	BLRS2	55
BRYAN2	33	BRYAN1	0

The second half of our qualification schedule was much more successful. We played some stronger teams but performed well against them. During this time, we were also running our skills challenge attempts. There is certainly points to improve on but our strategy seems to be the main point that needs an overhaul (to discuss later)

QF4 - LOSS	Score
BLRS2	52
AYO	75

AYO very clearly had spent time tuning and practicing with their robots. Talking with BLRS before the match, we knew this would be decided in the autonomous period and this is where we lost. We had fallen behind, and could not recover as they played heavy defense.

Robot Skills Challenge

Rank	4	Total Score	107
Driver Score	64	Programming Score	43
Driver Attempts	3	Programming Attempts	2

Throughout the tournament, our **scores improved as our drivers became more secure in their communication and driving abilities and our programmers tweaked the code.** While these scores fell short of our goals for this tournament, we are still glad that we posted our baseline for the rest of the season at NUKE.

Subsystems

Drivetrain

Our drivetrain performed well throughout the tournament, allowing our team high mobility and maneuverability. However, we had a few incidents with **beaching over the center bar that we are trying to fix with our next major design cycle.**

Intake

At NUKE, our intakes had a firm grip on triballs at the beginning of the matches, yet the rubber bands on our intakes often snapped mid-match. Moving forward, **we are moving away from having rubber bands on our intakes to avoid this issue.**

Passthrough

The passthroughs on our robots performed decently, yet they proved to be **less effective than just holding triballs in our intake before releasing them**. Therefore, we have decided to forgo the passthroughs for the rest of the season. This will be shown in more detail in the next entry.

Wings

Our wings, while well-built and effective, did not prove useful for de-scoring during our time at NUKE. The wings will stay on our robot design because they were **helpful in moving large quantities of triballs across the field.**

Identify the Problem

- Match Strategy:
  - Throughout the tournament, the referee called multiple violations of <SG11> on BLRS2, which is a rule that we have since become more familiar. **If a team acquires three or more warnings because of this rule, they are then disqualified.**

<SG11> Elevation is protected. During the last 30 seconds of the *Match*, *Robots* may not contact the following:

a. The opposing *Alliance's Elevation Bars*

b. Opponent *Robots* who are contacting their *Elevation Bars*

c. Opponent *Robots* who meet the definition of *Elevated*

d. The Short *Barriers* adjacent to the opposing *Alliance's Elevation Bars*.

Figure 2: Screenshot of <SG11> From the VRC Game Manual

   - As a team, NUKE was an exercise in effective communication. We have decided to improve our communication skills with **more frequent points of contact both on the field and in the lab.**
   - Additionally, we have decided to budget **more time to practice driving the robots so that our drivers can more effectively maximize the abilities of our robots.** We have learned that no matter how much we redesign our bot, our performance in matches is defined by the relationship between the bot and its driver.
- Robot Skills Challenge:
   - Similar to some of the takeaways from our match strategy, we believe that the skills challenge would also **greatly benefit from more time for driver practice.**
   - The same logic for allowing more time for our programming skills as we build our next schedule.
   - In the same vein, we also believe that a **stronger channel of communication between our teammates will drastically increase our scores.**
- Subsystems:
   - The following ideas will be **explained and more clearly decided upon within our next entries:**
      - The rubber bands on our intake should be replaced with flex wheels to avoid breakage.
      - We should the passthrough on our robot to make more room for potential endgame mechanisms.

Post NUKETown Strategy and Robots Analysis

Date	@January 15, 2024
Category	15" Robot 24" Robot BLRS2
Authors	K Kathleen Lowe C Conner Siebert
Design Cycle	Design Cycle 2

Define the Problem

Following the NUKETown competition, our team has decided to reconsider our match strategies and how that will affect the design of our robots. **We met together in the lab to consider and decide upon our robots for the MCCC competition in two weeks.**

Potential Solutions

Solution 1: Passthrough Bowling of Triballs

- Cloned pass-through robots
    - These robots at NUKETown, while struggling in several aspects of the game, showed promise moving forward with additional adjustments and development. Key factors we would like to see improved moving forward are:
        - Maneuverability
            - While moving around the field, **the current massive intake makes it difficult to pass under the elevation bars.**
        - Fielding
            - **The current long intake has a dead zone while fielding,** created from the geometry to get the match loads over the match load barrier
        - Ability to Tune
            - As the current robots are full custom, **a lack of tuning often leads to longer design times**, as each part must be redesigned in CAD and then manufactured before it can be tested

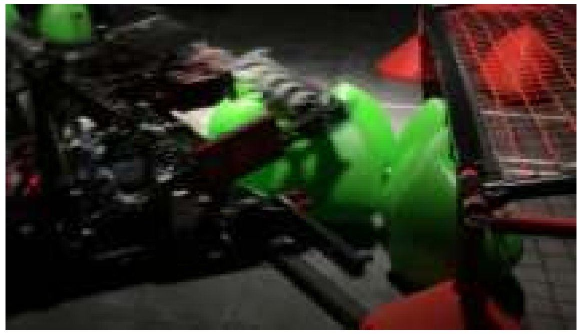


Figure 1: 21417A bowling Triballs into the goal

- This option would keep the pass through robots, created to intake triballs form one side and spit them out the rear while scoring.
  - Using wings and the back of the robot, **the goal of these robots is to push as many triballs from the match load zone to the goal** while passing under the elevation bars.

Solution 2: Triball Shuttling

- As the current chassis is strong at crossing the barrier and has a low profile, **stripping the other subsystems off of the robot would allow for a partial restart**
  - The following subsystems would be re-evaluated for their effectiveness in their current states
    - Intake size, geometry
    - Elevation strategies
    - Wing deploy methods

Figure 2: 9364H shuttling a Triball

- With these changes, our strategy would evolve to **shuttling triballs one at a time from the match load instead of large pushes.**
  - While this strategy would be slower at scoring, it would **allow for more control over all triballs on the field.** While opponents can play defense on the the robot pushing several triballs, holding one in the robot at all times would prevent opponents from stealing the triballs

Solution 3: Triball Catcher and Shooter

- A 3rd strategy our team has considered would start with redefining our autonomous strategy. To score triballs into the alliance goal, **one robot would sit at the match load zone and launch triballs into a hopper system** on the other robot.
    - This strategy in theory would be the most accurate in the autonomous period, as **all match loaded triballs would be scored in the alliance goal without any randomness** that can occur while bowling or shuttling.

Figure 3: Catcher robot from TNTN

- To design robots to meet this strategy, **a complete rebuild would likely be necessary.** That is because the current chassis is not designed for a triball catcher that is able to quickly placed shot triballs into the alliance goal

Choosing The Best Solution

Comparing each of the three strategies, our team considered the following pros and cons. From here, we created criteria and used a weighted decision matrix to determine the best option for MCCC in 2 weeks.

Pros

Passthrough Bowling of Triballs	Triball Shuttling	Triball Catcher and Shooter
Least time required to prepare for MCCC	Able to be started on existing chassis	Higher ceiling in autonomous period
Shown promise of improvement at NUKE	Controls triballs at all times	Highest accuracy
Bowling, while not optimal due to randomness of triball bounce, was able to win autos at NUKETown	Holds up to defense better than the other two options	Cannot lose triballs to opponents while operating correctly
	Simpler Robots	
	Easier to program	
	Small profile, better maneuverability	

Cons

Passthrough Bowling of Triballs	Triball Shuttling	Triball Catcher and Shooter
Losses triballs during auto to randomness, and inability to push into goals	Slowest at scoring while match loading	Complex robots
Difficult to program	Limits possession to 1 triball at at time	Difficult to program
Requires large intake, hindering mobility		Could fail while facing defense

Passthrough Bowling of Triballs	Triball Shuttling	Triball Catcher and Shooter
		Requires a complete rebuild, pushing our 2 week deadline

Decision Matrix

- Time Required (x/3)
    - As we are operating on 2 week timeline for this iteration of the robots, **allowing time for drive practice and programming is important**
    - A higher score indicates less time required to build
- Mechanical Complexity (x/5)
    - As robots that include more moving parts have a higher chance of failure, **minizine the failure points would be optimal**
    - As this game sees heavy defense by opposing robots at times, less is better in terms of holding up to these attacking robots
    - A higher score indicates less mechanical complexity
- Scoring Possibility in Autonomous (x/5)
    - As 38% of a match occurs during the autonomous period, being effective in scoring during this time is important to **optimize entering the driver control period**
    - Starting driver control with more triballs in our alliance goal would allow for several strategies to be deployed in driver control
    - A higher score indicates higher scoring potential in autonomous
- Control over Triballs during driver control (x/8)
    - Once we enter the driver control period, our team has determined the **majority of points will come from scoring triballs** into our alliance goal from the match load zone.
    - A higher score indicates more control over the Triballs
- Ability to hold up to defense (x/8)
    - Also occurring in driver period, we are intending a large amount of defense to be played against our offensive robots. **Controlling triballs while under defense** is also import during driver control.
    - A higher score indicates better ability to stand up to defensive play

	Time Required (x/3)	Mechanical Complexity (x/5)	Scoring Possibility in Autonomous (x/5)	Control over triballs during driver control (x/8)	Ability to hold up to defense (x/8)	Total (x/29)
Passthrough Bowling of Triballs	3	3	4	1	3	14
Triball Shuttling	2	5	3	8	8	26
Triball Catcher and Shooter	1	1	5	8	1	16

After considering each of the three solutions, our team has decided to move forward with Triball Shuttling. This is because of the following considerations:

- This option is the simplest
  - Given both the time constraint and other factors given to mechanical complexity, **we believe robots that are simple for MCCC will allow for stronger match play.**
  - Well built simple robots will also be able to play defense on opposing robots when desired
- Control over Triballs during driver and against defense
  - For reason stated above, this option gives us the **best chance to score in driver without allowing the opponents to benefit from our match loads.**
- The chassis are proven to work
  - Given these robots will be started on the existing competition robot chassis, we have a solid base for new designs. They meet all of our current requirements, **as they are able to cross the barrier, have other subsystems mounted to it, and holds up against defense fairly well.**

Implementation Plan

As we are under a tight two week time crunch until MCCC, we need to have a strict plan in place to accomplish our goals.

To see our plan in detail, please visit the next entry for a Task and Timeline Update.

Task and Timeline Update: January After NUKETown

Date	@January 15, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Conner Siebert
Design Cycle	Design Cycle 2

Pre-NUKETown Recap

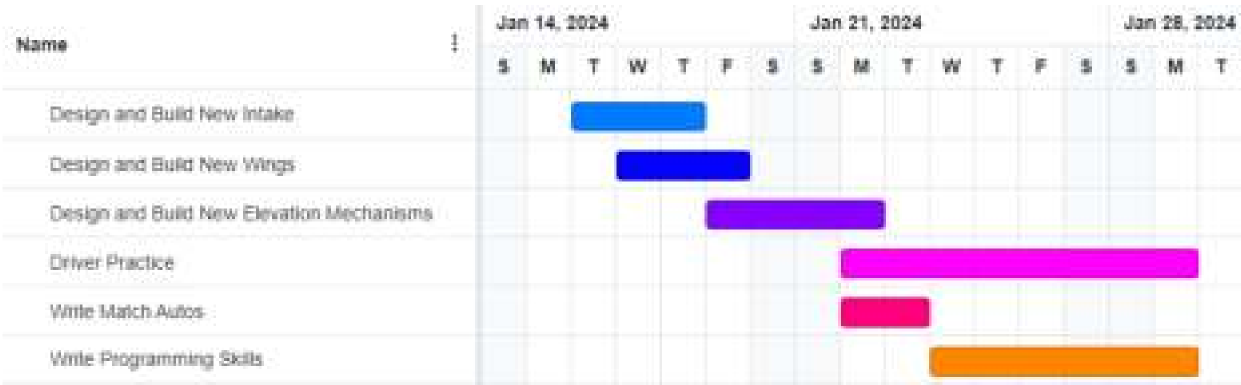
The team's first week back on campus before NUKETown was successful in finishing the robots for the competition. **See the competition recap entry for more information.**

After NUKETown, our team meet and discussed the future strategy

Post NUKETown Task and Timeline

In the next two weeks leading up to the MCCC, our team has set up three main design and build goals along with allowing time to drive practice and program the robots

- **The first week we aim to complete all design and building that needs to occur for both robots.** This will allow time for the drivers and programmers to prepare for MCCC
    - Design and build:
        - New intake
        - New wings
        - New Elevation Mechanisms


Name
Jan 14, 2024
Jan 21, 2024
Jan 28, 2024
S M T W T F S S M T W T F S S M T
Design and Build New Intake
Design and Build New Wings
Design and Build New Elevation Mechanisms
Driver Practice
Write Match Autos
Write Programming Skills

Design Brief: Iteration 2

Date	@January 15, 2024
Category	15" Robot 24" Robot BLRS2
Authors	J Jacob Zawacki S Stephen Hohnholt
Design Cycle	Design Cycle 2

BLRS2 Robots: Iteration 2

Problem Statement

These robots will need to play and win in VRC Over-Under, meaning that it needs to meet all constraints of the game manual and score more points than the opposing alliance. These robots need to be able to **score Triballs quickly and efficiently in the alliance goal, in both the autonomous period as well as the driver control period.** This includes picking them up off the floor and from the match-load zone. They then need to move them towards the offensive zone, and then finally score them in the alliance zone. They will also have an elevation for more points.

Design Statement

These robots will be designed to have a chassis, intake, wings, and hanging mechanism. The chassis will allow it to move around the field. The intake and wings will allow the robot to pick up Triballs quickly off the floor and in the match load zone, and then score them. The hanging mechanism is needed to achieve a high tier elevation for more points.

Constraints

- Begin within the set size constraints
    - One robot starting within a 24×24×24 inch box
    - One robot starting within a 15×15×15 inch box
- Intake Triballs from the match load zone
- Score Triballs into the alliance goal
- Elevate the robot onto the elevation bars

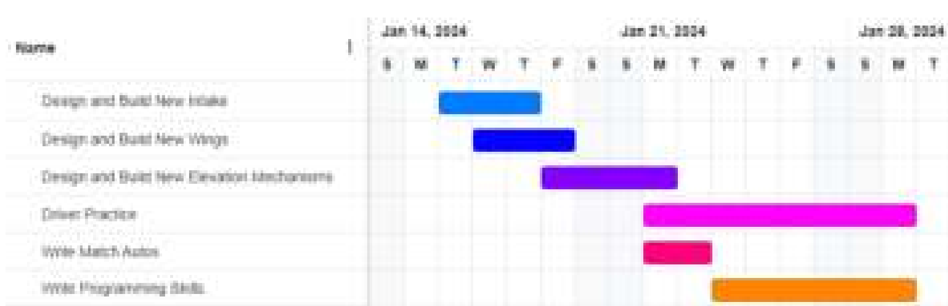
- Pop balls over the center barrier

Criteria

- Be able to intake or pop Triballs from the match load zone at a rate of 1 Triball per second
    - This will allow us to **quickly score all 10 allowable match loads** during the autonomous period, as well as allow us to quickly intake in driver
- Efficiently shuttle Triballs across the field to our scoring goal
    - To do this, we will need to have efficient drives, quick and secure intakes, and a large amount of drive practice going into tournaments
    - By being able to effectively shuttle, **we will be able to score Triballs without fear of losing them to other teams**
- Achieve elevation at the end of the match
    - As shown by our performance at NUKETown, being able to achieve an elevation, even at the lowest tier, **will add a large amount of points to our overall score**
    - We will prioritize both robots being able to elevate in some manner, as decided in a future entry

Important Deadlines

- 1/18/2024: Design and Build New Intake
- 1/19/2024: Design and Build New Wings
- 1/22/2024: Design and Build New Elevation Mechanisms
- 1/27/2024: MCCC Tournament
    - Robots should be fully built and programmed.



Brainstorming and Decision Matrix: Intake Redesign

Date	@January 17, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Conner Siebert Jacob Zawacki
Design Cycle	Design Cycle 2

Identify the Problem

After competing at NUKE last weekend, **our team identified a major point of concern moving forward is our intake.** The current design has the following issues we would like to address:

- The intake has large reach out the front of the robot.
  - This caused **issues maneuvering around the field**, especially around the short barriers
    - Turning around the short barriers would require precise turns to avoid hitting the walls
- There was a dead zone in middle of the intake while fielding
  - While the 2nd roller on the existing intake was raised to allow the Triball to pop over the match load bar, while intaking from the field **Triballs would get stuck between the 1st and 3rd roller** while not contacting the 2nd roller
  - To fix this issue, the driver would have to **release the Triball back onto the field** and attempt to intake it again
  - The robot has an **inconsistent center of gravity**
    - While the current intake is fully extended, the front of the robot is heavy
    - This causes issues while causing the barrier, as a **bigger runup was required to have the momentum to clear the long barrier**
      - The robot struggled to carry triballs across the barrier
        - With the large intake, while crossing over the long barrier with a Triball it would often come out

Figure 1: Top View of Current Intake

- This caused issued with the **opponents being able to take the Triball from us**, and score them into their own alliance goal

The goal is for our robot to **have a strong intake for MCCC competition in two weeks.**

Brainstorm Solutions

Solution 1: Keep Current Intake without changes for MCCC

- By keeping the current intake, our team **maintains the ability to pull over the match load barrier efficiently**
  - This solution would allow for **more time on other subsystems of the robot** in the coming weeks
  - Additional time could be spent on driving practice and programming skills
  - This solution **does not fix the current issue with our intake**

Solution 2: Improve the Current Intake

- With the current 4 bar intake, we feel a possible solution is to further tune the intake.
  - This solution would aim to keep the same intake mount for our robot, along with arms
  - The goal of this rebuild would be to **make the intake shorter, while maintaining the ability to pull over the match load zone**

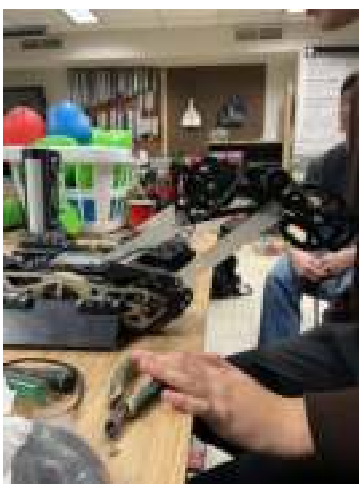


Figure 2: Built intake improvement to be tested

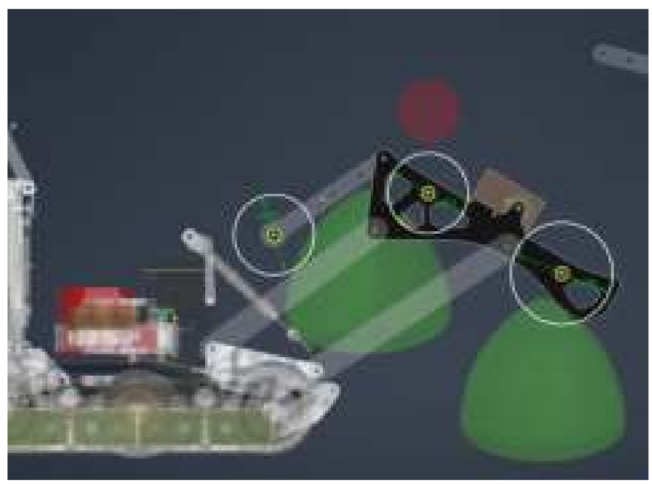


Figure 3: CAD of prototyped intake improvements

Solution 3: Begin to Design a New Intake with Flex Wheels

- We would like to test the use of flex wheels
  - By using **flex wheels on our intake**, we hope to get better grip on the triballs
  - Rubber bands have posed issues in the past, as they can become entangled with opposing robots
  - This is inspired by high school teams, such as 99904B Boogie Woogie

- The third option is to build a new intake with a smaller footprint on the robot
  - The goal of this prototype would be to have the **best possible fielding of triballs**
  - Our team would like to explore the option of the intake not needing to reach over the match load zone

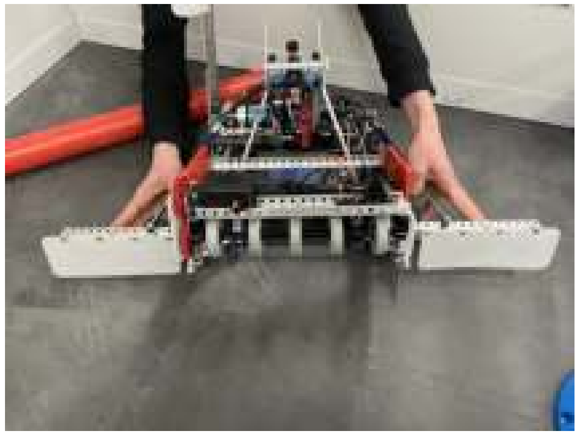


Figure 4: Flex Wheel intake from team 99904B Boogie Woogie

Prototype Solutions + Testing

To assist in determining which course of action we should take, we determined **three possible mechanisms to test:**

1. Our current intake
2. Our current intake, swapping out the arm pieces for shorter, rough plastic
   a. This will simulate the possible changes we can make as described above
3. A flex wheel intake prototype

We can then use the data we gather to **help make our decision.**

Test 1: Ability to pull over the barrier

One aspect of the intake is the ability to intake Triballs from over the match load barrier. **While we may be able to use alternative mechanisms to secure match loads, having data will help us make an informed decision one way or another.**

Testing Procedure

1. The robot is positioned against the match load bar, as close to the bar as possible.
2. The intake is powered on at full power.
3. 5 Triballs are manually fed into the intake in its resting position.
4. The speed at which the Triball is fed into the main body of the robot is recorded.

Test Data

	Test 1 (seconds)	Test 2 (seconds)	Test 3 (seconds	Test 4 (seconds)	Test 5 (seconds)	Average
Current	0.5	0.5	0.6	0.5	0.5	0.52s
Modified	1.1	1.1	0.9	0.8	1.0	0.98s
Flex Wheel	DNC	DNC	DNC	DNC	DNC	DNC

Test 2: Fielding Efficiency

One important factor, as previously determined, is the ability to field Triballs from the field tiles. This well help **provide a measure of overall intake efficiency.**

Testing Procedure

1. A Triball is placed on the field exactly 24″ away from the front face of the intake.
2. The robot drives forward at full speed with the intake powered on until the Triball has been fielded
3. The time it takes from the Triball to move from the front face of the intake to the inner robot is recorded.

Test Data

	Test 1 (seconds)	Test 2 (seconds)	Test 3 (seconds)	Test 4 (seconds)	Test 5 (seconds)	Average
Current	1.8	1.6	1.6	1.7	2.0	1.74s
Modified	1.5	1.4	1.7	1.5	1.5	1.52s
Flex Wheel	1.1	0.9	1.1	1.1	1.0	1.04s

Testing Conclusions

Considering the data, it appears that the **current intake has the most efficient match loading, whereas the flex wheel intake has the most efficient fielding**. The length of the current intake likely has a direct correlation to its ability to match load, while the overall geometry of the flex wheel intake leads to improved fielding.

Select the Best Solution

After testing was conducted, we compared the three possible solutions to choose the best option moving forwards.

Pros

Keep Current Intake without changes for MCCC	Improve the Current Intake	Begin to Design a New Intake with Flex Wheels
More time can be spent on other subsystems	Intaking from the field would be better	Intaking from the field would be the best
The intake can pull over the match load bar		The intake would be shortened
		Flex wheels would prevent entanglement with other robots

Cons

Keep Current Intake without changes for MCCC	Improve the Current Intake	Begin to Design a New Intake with Flex Wheels
Intaking from the field is poor	Lesser ability to match load over the barrier	Requires additional time to design and build
The intake pushes the center of gravity far forwards	Still has a fairly large profile	Cannot match load over the barrier

Decision Matrix

The following criteria were considered while we decided upon which decision to make for our intake.

- Overall Maneuverability (x/10)
  - The robot needs to be able to **maneuver around the field quickly and without issue**
  - The intake should not be so long as to massively affect Center of Gravity or get in the way of obstacles
  - A higher score means that the robot is more maneuverable
- Match Loading Efficiency (x/10)
  - The ability to **intake Triballs directly from the match load zone**
  - Taken directly from testing results
- Fielding Efficiency (x/10)
  - The ability to **intake Triballs from the field tiles**
  - Taken directly from testing results
- Time to Construct (x/10)
  - How long the intake mechanism will take to build
  - A higher score means a **faster time to construct**
- Reparability (x/10)
  - The intake should be **easy to repair, adjust, and tune** for optimal performance
  - A higher score means easier reparability

	Maneuverability	Match Loading	Fielding	Time to Construct	Reparability	Total (x/50)
Current Intake	1	9	3	10	2	25
Modified Intake	5	5	5	5	4	24
Flex Wheel Intake	9	0	9	3	9	30

After considering each of the three solutions, our team has decided to move forward with a flex wheel intake. This is because of the following considerations:

- High Maneuverability
  - As the intake with the smallest overall profile and relatively light construction, the **flex wheel intake will be very maneuverable around the field.**

- Ease of Fielding
  - Through our testing, we established that the flex wheel intake has the **highest fielding efficiency compared to all of our prototypes.**
- Reparability
  - As this mechanism would use VEX parts, this **allows us more freedom when it comes to making quick repairs** or tuning for efficiency.

Implementation Plan

- Match Loading
  - As this intake cannot intake from the match load zone, **we will need a mechanism to score match loads during autonomous and driver**
  - This will be decided upon in a later entry
- As we have two weeks until our next tournament at MCCC in Michigan, **we will need to have this intake built ASAP**
  - To guide our design and construction of the new intake, we have included a timeline below:
    - **January 16**
      - Decide the specifics of the intake as far as mounting location and material
    - **January 18**
      - Take off and deconstruct the old intake to leave room for the new iteration
    - **January 19**
      - Build and test the updated intake

Brainstorming and Decision Matrix: Wing Redesign

Date	@January 18, 2024
Category	15" Robot, 24" Robot, BLRS2
Authors	Conner Siebert, Jacob Zawacki
Design Cycle	Design Cycle 2

Define the Problem

After competing at NUKE last weekend, our team identified another point of improvement going into the next design cycle: our wing design. A few of the problems we noticed are:

- The size is limited by the vertical fold-up
    - As the wings fold up vertically alongside the pneumatic air tank mounts, the **wings need to be short enough so as to not hit the horizontal elevation bar** when driving under
    - **This limits the overall reach of the wings**, as the wingspan is equal to the folded-up length of each wing
- The wings could not be easily used for match-loading
    - With our old intake design, the intent was to intake Triballs from the match-load zone directly with our intake
        - As our intake mechanism is going to be changed to reflect the decision made in the previous entry, this will no longer be the case
        - **Using our wings to "kick the Triballs out of the match load zone will be our main strategy going forward**
    - Additionally, the short wingspan of the wing design did not allow much length to reach into the goal

Figure 1: Previous Wing Design

With our intake no longer being able to intake Triballs from the match-load zone, having wings that could remove Triballs upon actuation would be ideal - making this our **main goal for this upcoming design cycle**.

Potential Solutions

Solution 1: Keep Current Wings without changes for MCCC

- By keeping the current intake, our team can focus time into other reworks happening before MCCC
  - This solution would **allow for more time on other subsystems of the robot** in the coming weeks
  - As there was no functional issue with the Wings, they would still work without change

Solution 2: Horizontal Fold Out Wings

- Another option for wings would be a horizontal deploy, with the wings folding up along the side of the drive train
  - This solution would **allow us to extend the wings without worrying about the height limit**
  - This solution would also **allow us to intake from the match-load zone**, as horizontal actuations would swipe the Triball from the match-load zone

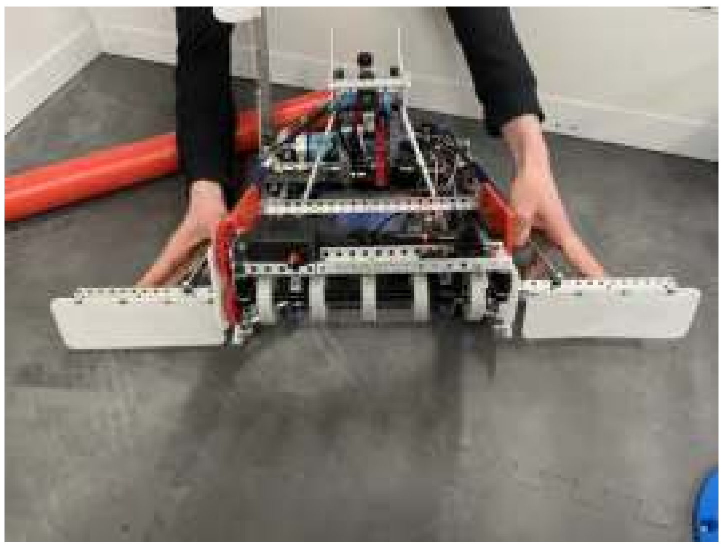


Figure 2: Flex Wheel intake from team 99904B Boogie Woogie

Solution 3: Utilize Both Solutions

- As shown in the image by team 21417A, one possible solution could be to a**dd both types of wings to our robot**, with the vertical wings staying where they are, and adding horizontal wings to the front of the robot
- This solution would allow us to keep much of the work we have now, **only needing to add horizontal wings at the front of the robot**

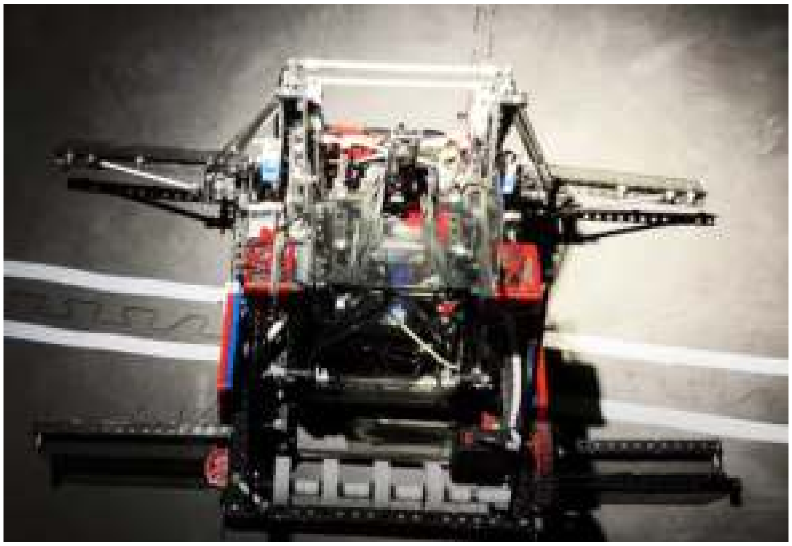


Figure 3: Combination of wings used by team 21417A Robokauz

Choosing The Best Solution

With these three options in mind, we considered the following benefits and drawbacks of each solution:

Pros

Keep Current Wings without changes for MCCC	Horizontal Deploy Wings	Combine Both Solutions
More time can be spent on other subsystems	Ability to intake from match-load zone	Ability to intake from match-load zone
	Increase the overall wingspan	Increase the overall wingspan
		More versatility with which wings to use in matches

Cons

Keep Current Wings without changes for MCCC	Horizontal Deploy Wings	Combine Both Solutions
Shortest overall wingspan	Takes up time from improving other subsystems	More complexity for driver and autonomous
No ability to intake from match-load zone		Would take the most time to build
		Lack of potential use-cases

Decision Matrix

The following criteria were considered while we decided upon which decision to make for our wings.

- Time to Construct (x/10)
    - **How quickly we can complete the wings** to our satisfaction

- A higher score means faster time to construct
- Match-Loading Ability (x/10)
  - Ability to **intake Triballs from the match-load zone**
  - A higher score means a more efficient intake
- Wingspan (x/10)
  - The relative **length of the wings from end to end**
  - A higher score means a wider wingspan
- Ease of Use (x/10)
  - **How simple will the wings be to use** in Driver and Autonomous
  - A higher score means simpler to use
- Usability (x/10)
  - The wings should be **practical and well positioned for use during matches**
  - A higher score means higher usability

	Time to Construct	Match Loading	Wingspan	Ease of Use	Usability	Total (x/50)
Current Wings	10	0	6	8	5	27
Horizontal Wings	8	8	10	10	10	46
Combination	6	8	10	5	8	37

After considering each of the three solutions, our team has decided to move forward with horizontal wings. This is because of the following considerations:

- Match Loading Ability
  - With a horizontal arc, these wings **will have the ability to intake Triballs from the match-loading zone**, meeting our goal established earlier in this entry
- Ease of Use
  - With one set of wings at the front of the robot, it will be **very easy for our driver and programmers to utilize these wings** to score Triballs
- Wingspan
  - With a horizontal fold-up, these wings will have a **much longer wingspan** than the vertical wings currently do

Implementation Plan

- While we need to have **wings completed before MCCC**, they are a relatively low-stakes mechanism
  - The Horizontal Wing concept is fairly simple, and should be **relatively easy to mount to the robot**
- With this in mind, it has the lowest priority of the current pre-MCCC changes to the design. Here is a schedule for when we plan to have the wings attached to the robot:
  - **January 20**

    - Have the concept and ideation for the specific wing implementation decided
  - **January 22**
    - Disassemble the old wings, so that we have room to mount the new wings
  - **January 24**
    - Build and test the new horizontal wings

Design and Build: Intake Iteration 2

Date	@January 18, 2024
Category	15" Robot BLRS2
Authors	Ⓒ Conner Siebert Ⓑ Brandon Liu
Design Cycle	Design Cycle 2

Goals

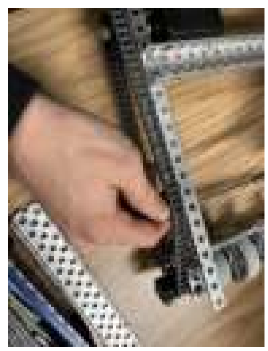
- After prototyping possible improvements to our current intake, **our team is planning to build a flex wheel intake out of vex metal**
  - Given our team is leaving for MCCC in a week, the goal is to build the new intake in a short period of time
  - By using vex metal, the intake can be **tuned and adjusted easier than a full custom intake**
- Improved fielding
  - The current intake struggles to pick up loose triballs from the field
  - A goal for this robot is to **quickly intake the triballs from the field in less than 0.5 seconds**
- Small footprint
  - The first version of the intake was massive
  - By creating a smaller intake, our robot will be able to **maneuver around the field much more efficiently**
- Strong control over the Triball
  - While going from the match load zone to the alliance goal with a Triball, our first intake would often lose the triballs
  - A goal of this intake is to be able to **cross the long barrier while holding a Triball in the intake**

Build the Solution

- Step 1
  - Two 23 long 2-wide C-channels are **mounted together using standoffs.** The spacing between the two c-channels are 7.5.″
- Step 2
  - A **high strength pillow bearing** is mounted to the bottom of the c-channels.
- Step 3
  - A high strength axle with length of 10″ is ran between the two pillow bearings, with a **total of 4 1.625″ and 2 2″ flex wheels** positioned between the c-channels
- Step 4
  - A **11W motor is attached to the insides of both c-channels**, with a flat bearing positioned inside the c-channels
- Step 5
  - A **18 tooth sprocket** is added on an axle attached to the motor
  - A **6 tooth sprocket** is attached to each ends of the high strength axle
  - 6P chain (found for Vex IQ product line) is attached between the two sprockets
- Step 6
  - A 1/8″x3/4″x8 1/2″ aluminum flat bar is mounted to the top of the

Steps 1, 2, and 3: Intake Axle attached to the intake bars

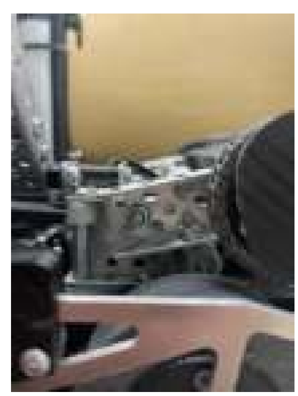
Steps 4 and 5: Motor and sprocket



Step 5: Sprockets and chain run on one side of the intake

Step 5: Motors and chain on the intake

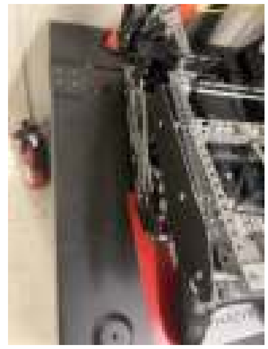
Steps 6 and 7: Top view of cross brace flat bar and standoffs



Step 8: Side view of the screw joint

intake near the front. This is to act as **protection for the flex wheels**

- Step 7
  - 3 3″ standoffs are mounted in middle of the intake to **prevent the triballs from passing through**
- Step 8
  - A c-channel is attached to the inner driver plate, to **allow a screw joint to mount the intake**
- Step 9
  - Polycarbonate covers are attached to the outside of the intake to protect the chain. It is attached by 1.5″ standoffs, which **doubles as a tensioner for the intake chain**



Step 9: Polycarbonate cover to protect the intake chain

Figure 1: Completed Intake

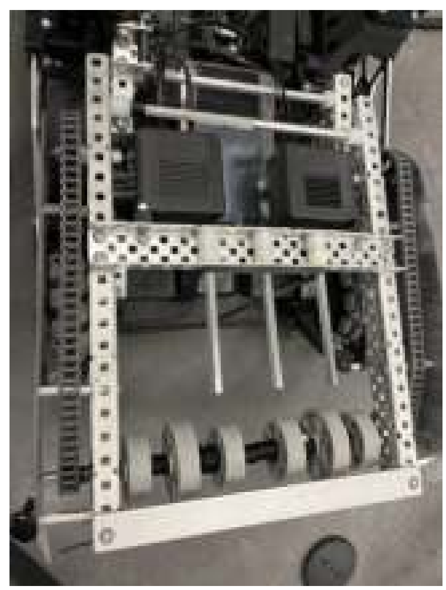


Figure 2: Top view of completed Intake

Test Solution

To ensure that the intake meets our established goals, we will test how quickly the intake can field a Triball from the ground. **The specific testing procedure is outlined below:**

1. Place a Triball flat on the surface of the field tile, 12 inches away from the robot.
2. Drive the robot forward at full speed, with the intake spinning at maximum velocity.
3. Record whether the speed at which the Triball moves from the tile to encased in the intake.
4. Repeat the process four additional times.

Results

	Intake Speed
Test 1	0.44
Test 2	0.44
Test 3	0.40
Test 4	0.45
Test 5	0.43
Average	0.43

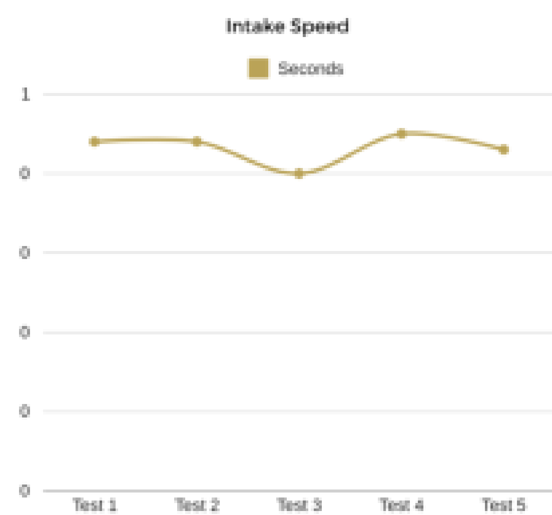


Figure 3: Testing graph

As the intake speed is less than our maximum allowable speed (0.43 < 0.5 seconds), this intake design is a success.

Brainstorming and Decision Matrix: Elevation Mechanism

Date	@January 19, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Conner Siebert Jacob Zawacki
Design Cycle	Design Cycle 2

Identify the Problem

The second major subsystem upgrades determined to be needed between NUKETown and MCCC was the **requirement for endgame elevations**.

In both matches and skills at NUKETown, our team lost out on points due to neither robots having the ability to elevate at the end of matches.

- In matches, we lost a potential of up 40 points due the lack of elevations
- In the robot skills challenge, up to a total of 80 points

Our goals for elevations are the following for MCCC

- Achieve 35 points in matches
    - By being the highest and 2nd highest tiers in matches, our robots can achieve 20 and 15 points each match
    - We believe that to achieve this goal, a B or C tier and B or A tier elevation will be required between the two robots
- Get 15 points in both driver and programming skills
    - As in the robot skills challenge, an A tier elevation is 5 points and B-D tiers are 10 points
    - To archive this, we will need one robot in each of the 5 and 10 point regions

Brainstorm Solutions

Solution 1: Long Barrier Park

- The long barrier park works by a robot **driving up onto one of the black barrier bars in the middle of the field** and balancing their chassis such that no wheels touch the ground.
  - While balancing, the robot must make some sort of contact with the colored alliance elevation bars.

Figure 1: Barrier Elevation - 1095R

Solution 2: Horizontal Elevation Bar

- The horizontal elevation bar hang works by the robot driving underneath the horizontal colored alliance elevation bar and **pulling itself off the ground.** The robot will then balance while hanging such that none of its drive wheels are touching the ground.
  - Since we are grasping the alliance elevation bar to hang off the ground.
  - This could be accomplished by a **passive or active hook mechanisms** which will dictate how high the robot gets off the ground.

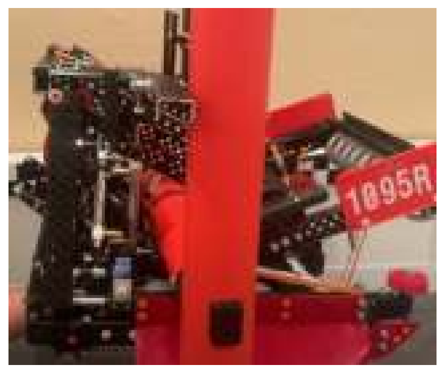


Figure 2: Horizontal Bar Elevation - 1095R

Solution 3: Vertical Elevation Bar

- The vertical elevation bar hang is a hang strategy where a robot uses some form of clamping mechanism to **grasp the vertical alliance elevation bar and pull itself up and off the ground.**
  - Similar to how some teams elevated during VEX Start struck.
  - To be more effective, it is possible to first **drive up onto the low black barrier and then perform the vertical elevation bar hang.** This enables the robot to start higher off the ground.

Figure 3: Vertical Elevation - FrostByte Robotics

Select the Best Solution

To compare each of the three hanging solutions, our team created a list of pros and cons for each option then created a decision matrix

Pros

Long Barrier Park	Horizontal Elevation Bar	Vertical Elevation Bar
Very fast	Can still be quick	Highest point range potential. B to H range.
Simple mechanism	Higher point range. A to C range.	
Large margin of error		

Cons

Long Barrier Park	Horizontal Elevation Bar	Vertical Elevation Bar
Lowest point range. A tier	More complex than a barrier park	Slowest of the three
	Only one robot can hang here	Most complex
		Needs highest amount of precision

Decision Matrix

The following criteria were considered while we decided upon which elevation to add on our robots.

- Feasibility (x/10)
  - **Ability to design and build the mechanism** in the limited time we have
  - 10 is most feasible, 1 is least
- Ability to Use (x/10)
  - Does the mechanism pass the "common sense" test of **holding up under competition stress**
  - 10 is most durable, 1 is least
- Scoring Potential (x/10)
  - Amount of **points the mechanism can score**
  - 10 is most points, 1 is least
- Speed (x/10)
  - The **time it takes for the mechanism to elevate completely**
  - 10 is the fastest, 1 is the slowest
- Precision Required (x/10)
  - The amount of precision it takes **to line up the robot for the elevation**
  - 10 is the least precise, 1 is the most

	Feasibility	Ability to Use	Scoring Potential	Speed	Precision Req.	Total (x/X)
Barrier	10	9	2	10	7	38
Horizontal Bar	8	9	4	8	8	37
Vertical Bar	3	4	10	6	6	29

After considering each of the three solutions, our team has decided to move forward with a Barrier elevation for our 15″ robot, and a Horizontal Bar elevation for our 24″ robot. This is because of the following considerations:

- Both the Barrier and Horizontal Bar elevations scored well in our decision matrix, with **only one point separating the two total scores**.
  - Because of this, and considering that each location only has room for one elevated robot, **we will make use of each mechanism on each robot**.
- Because the Barrier elevation will take up less space, we will build that on the smaller of the two robots. Therefore, the 24″ robot will have the Horizontal Bar elevation.

Implementation Plan

- With the MCCC tournament fast approaching on the 27th, we need to build both mechanisms as fast as possible **in order to thoroughly test and ensure both are ready for competition**.
- As such, we will plan to **build both elevation mechanisms tomorrow**, so that we have the full week to test and practice. For reference, here is a planned timeline:
  - **January 18:** Decide the specifics of how each elevation will be implemented on the robot.
  - **January 19:** Build both elevation mechanisms and test.
  - **January 27:** Compete at MCCC.

Design and Build: Elevation Mechanisms

Date	@January 19, 2024
Category	15" Robot 24" Robot BLRS2
Authors	C Conner Siebert K Kathleen Lowe
Design Cycle	Design Cycle 2

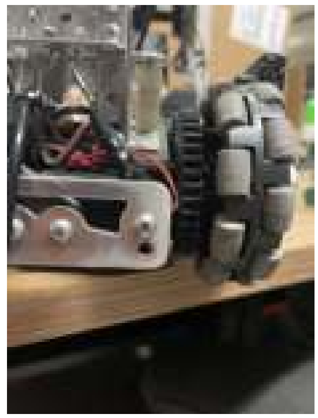
Goals

- To add the “c” tier elevation to the 24” robot, our team plans to use a two bar mechanism with 2 motors powering it.
    - This was determined as the optimal solution due to the **large amounts of torque provided by two motors**
    - Our gear ratio will be 14 rpm, from a 1:7 powered by two 100 rpm motors
    - To determine whether or not the solution is successful, **we will test using a benchmark of 5 seconds for a successful elevation.**
- To allow the 15” robot to elevate on the long barrier, a support along the vertical elevation bar
    - To deploy the arm, a piston should be utilized
        - This is because it is only a 1 time actuation to drop the arm. **It will not be required to be retracted**
    - To determine whether or not the solution is successful, **we will test using a benchmark of 3 seconds for a successful elevation.**
- To design these mechanisms, our teams to prototype and designed biased off of vex parts, **diverging from the prior full custom design method**
    - This design philosophy change was driven by two main factors: the **inability to tune things and the time required**
        - While tuning items on a full custom project, any changes to hole locations requires a new machined piece

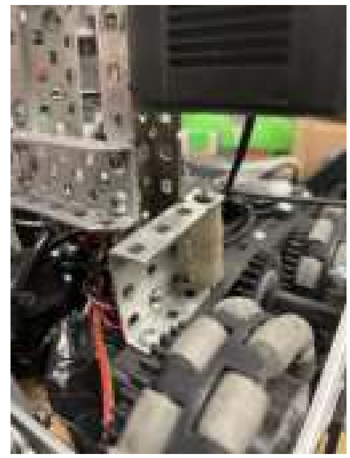
- This was getting costly, as the old piece required scrapped when replacing
- Anytime a new piece was needed, lead times went from 6 to 24 hours for a 3D print and 1 to 3 days for machined metal
  - This lead to timeline issues, which we can not afford in the coming week preparing for MCCC

Building the 24″ Elevation Mechanism

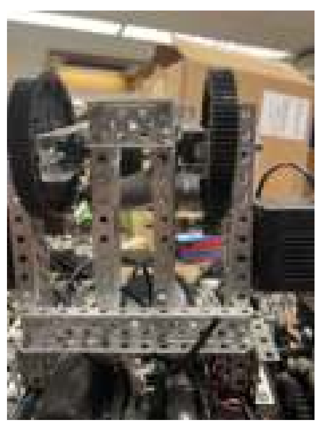
- Step 1
  - To begin to mount the vertical towers to the robot, a **5 hole long (2.5″) 3-wide C-channel is screwed onto the inner chassis plate.**
- Step 2
  - To add additional support to the c-channel and to prevent bending of the piece, a method of vertical support called "boxing" is done.
    - To do this run a screw with spacers through the middle of the c-channel to **prevent them from bending outwards**
- Step 3
  - Two 15 hole long (7.5″) 2 wide c-channels are attached to the mounts.
  - A 12 hole (6″) c-channel is used as a **cross brace between the two towers**



Step 1 and 2: Back view of boxed 3-wide C-channel for tower mounting



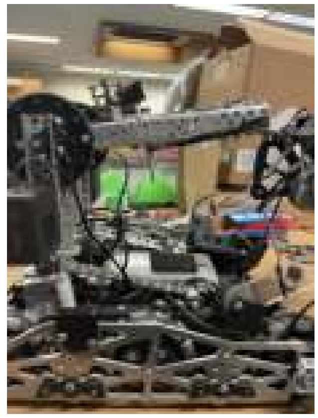
Step 1 and 2: Side view of towers mounted to the inner chassis piece



Steps 3, 4, and 5: Rear view of the support towers

Steps 6, 7, and 8: Top view of hanging bars

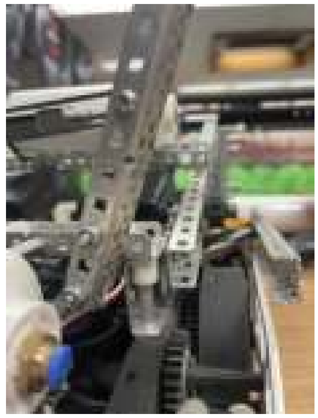
- Step 4
  - Two 12 hole (6″) c-channels are mounted to the cross brace to allow the **axles to be supported through all 4 towers**
- Step 5
  - Two **100 rpm motors** are attached to the vertical towers
- Step 6
  - Two 20 hole (10″) c-channels are mounted to 84-tooth gears and **cross braced with a 7 hole (3.5″) c-channel**
- Step 7
  - A high strength axle is ran through the 4 vertical towers and the two 84-tooth gears to **attach the elevation lift to the vertical support towers.**
- Step 8
  - Screws are ran through the elevation bars to allow the horizontal elevation bar to be **support between the two pieces**
- Step 9
  - A 10 hole (5″) 1×1×1 L-bar is mounted to the chassis and the vertical towers, as seen in figure 5, to **help support the lift structure**



Step 9: Side View of elevation bars

Building the 15″ Elevation Mechanism

- Step 1
  - A 16 hole (8″) C-channel is ran between the two **3-wide c-channel for the intake mounts**
- Step 2
  - A 15 hole (7.5″) C-channel is added to a screw joint on the newly added cross brace.
  - Using screws and spacers, the **screw joint is boxed in both c-channels to prevent bending**
- Step 3
  - A c-channel coupler is attached to the cross-brace to mount the piston to. **A 4 hole (2″) 2 wide-channel is attached to the piston.**
- Step 4
  - A 2.5″ standoff is attached to the end of the elevation arm, to grab the elevation bar. **This screw is also boxed, to prevent the c-channel from bending inwards** as the guide standoff

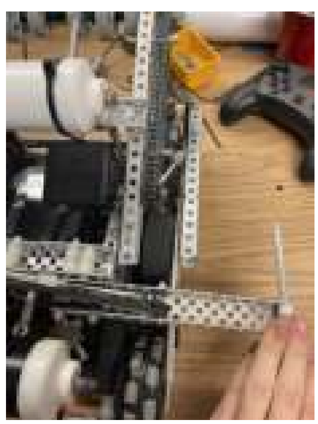


Step 1: Elevation bar attached to cross-brace

Step 2: Top view of the cross-braced

Step 3: Piston attached to the C-Channel coupler

Steps 4 and 5: Piston attached to the arm



Steps 5: Elevation Arm Extended

- Step 5
    - The **piston is attached to the drop down bar**
    - A rubber band is used to hold the arm up
        - This allows the piston to only be a single acting, as it will be rubber banded upwards and pistoned down.

Top view of the elevated robot

Side view of the robot elevated to A tier

Test Solution

24″ Elevation

To ensure that the elevation meets our established goals, we will test how quickly the lift can elevate the robot off of the ground. **The specific testing procedure is outlined below:**

1. Position the robot directly underneath the horizontal elevation bar with the hang fully extended.
2. Ensure that the hang bar is contacting the elevation bar, and spin both motors at full speed.
3. Record the time it takes for the robot to fully elevate into the C tier.
4. Repeat the process four additional times.

	Elevation Speed
Test 1	4.88
Test 2	4.65
Test 3	4.63
Test 4	5.11
Test 5	4.73
Average	4.8

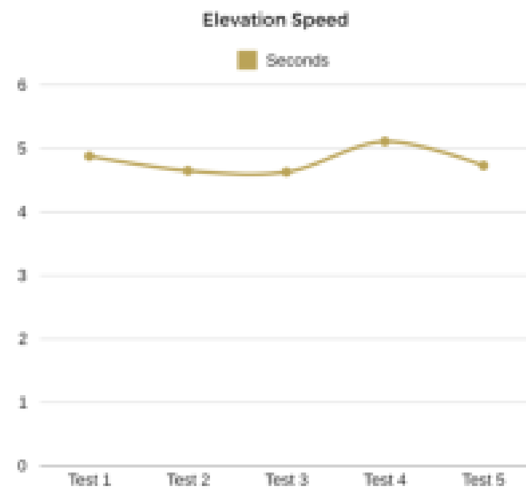


Figure 1: Testing graph

As the robot can elevate in an average of 4.8 seconds, under our benchmark of 5 seconds, **we can conclude that the elevation successfully meets our criteria.**

15″ Elevation

To ensure that the elevation meets our established goals, we will test how quickly the robot can drive onto the bar. **The specific testing procedure is outlined below:**

1. Position the robot on the ground, one foot away from the horizontal barrier barrier.
2. Drive the robot forward at full speed with the drop down bar extended
3. Record the speed at which the robot balances on the barrier.
4. Repeat the process four additional times.

	Elevation Speed
Test 1	2.14
Test 2	1.78
Test 3	1.88
Test 4	2.15
Test 5	1.94
Average	1.98

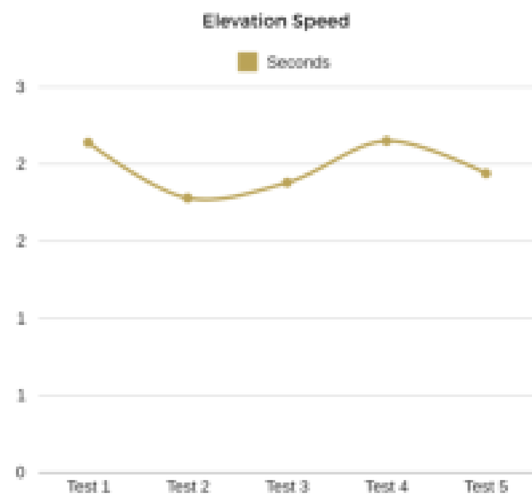


Figure 2: Testing graph

As the average elevation time is well under the benchmark of 3 seconds, **we can conclude that the elevation successfully meets our criteria.**

Comp Software: MCCC Match Autonomous

Date	@January 22, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Mihir Laud Alex Lam
Design Cycle	Design Cycle 2

Objectives

- [ ] Plan match Autonomous for both 15″ & 24″ robots
- [ ] Create flowcharts of autonomous program(s)
- [ ] Implement autonomous code

Planning

The main objective for the match autonomous for both bots is to **provide an advantage within operator control while securing the Autonomous Win Point (AWP).** The AWP will allow the team to obtain a high placement in the qualification rankings.

To complete this, **both robots will be responsible for scoring an Alliance Triball in the goal, as well as touching the hanging bars at the end of the autonomous period.**

Before programming and testing the autonomous on the robots, a route is planned and mapped (as shown below):

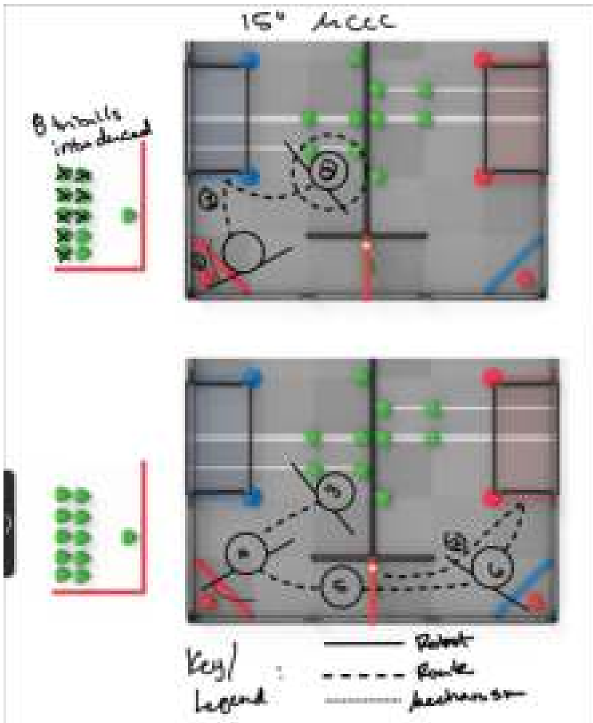


Figure 1a: Match Autonomous Route For 15″ Robot

Step of Autonomous	Procedure
Step 1	Begin by knocking out & grabbing triball in the alliance loading zone
Step 2	Score the triball into the nearby goal
Step 3	Grab triball on the single line closest to the goal and spin to grief triballs to prevent the opponent from scoring them
Step 4	Return to the alliance load zone and introduce triballs (around 8)
Step 5	Push triballs introduced across towards alliance goal
Step 6	Push the triballs into the goal

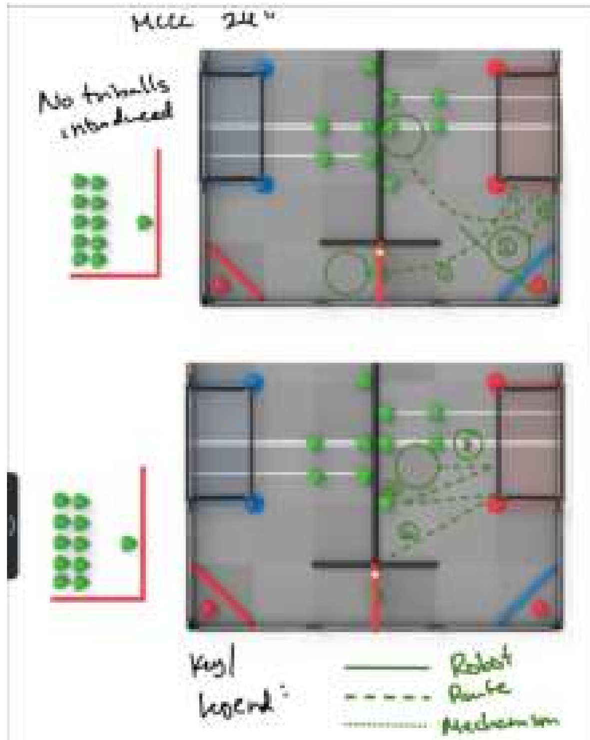


Figure 1b: Match Autonomous Route For 24″ Robot

Step of Autonomous	Procedure
Step 1	Begin by picking up triball under alliance parking pole and scoring into alliance goal
Step 2	Move towards opposing loading zone and remove triball out of zone
Step 3	Score the triball into alliance goal and move towards triballs in the middle of the field
Step 4	Pick up the left triball on the double line and score into alliance goal
Step 5	Pick up triball closest to alliance parking pole and score into alliance goal

Step 7	Move towards alliance parking pole to secure Autonomous Win Point

Step 6	Move towards alliance parking pole to secure Autonomous Win Point

With the mapped plan, our team can score at least 5 triballs within the goal and 10 triballs in the offensive zone. This would therefore net our team at least 45 points during the autonomous period. **Following the mapped plan, a flowchart was created for both robots to convert the route to pseudocode (code procedure):**

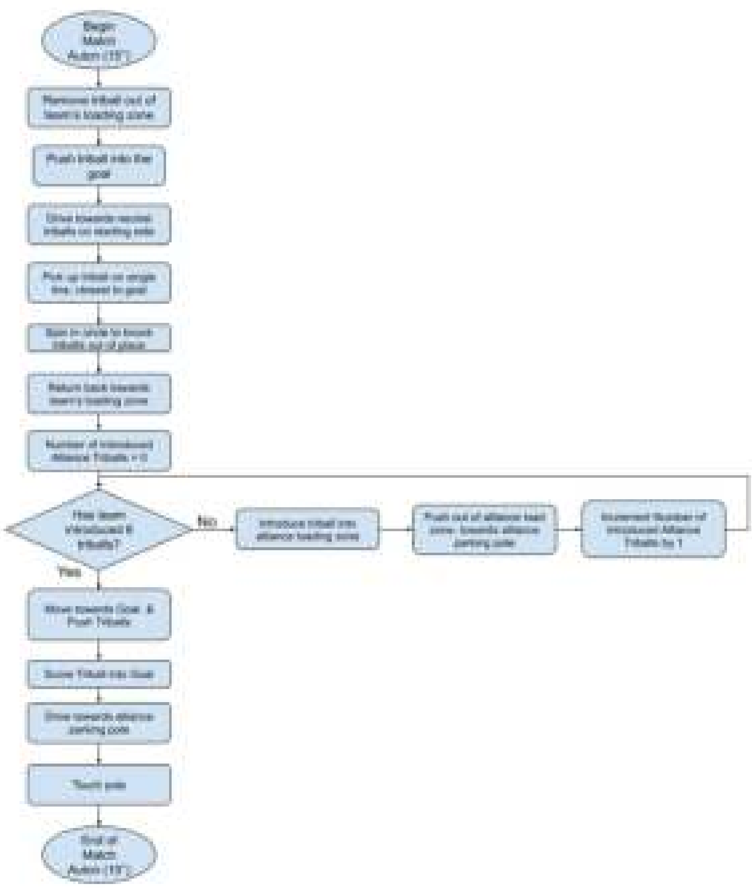


Figure 2a: Flowchart for Match Autonomous Route for 15″ Robot

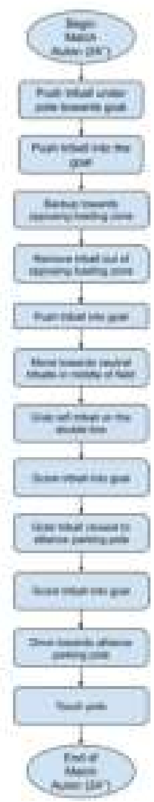


Figure 2b: Flowchart for Autonomous Route for 24″ Robot

Following the flowcharts, implementation of the logic was applied to the robot for competition use. Initialization of the motors was first done before odometry was utilized to program the robot movements of the route via our ARMS library.

Mechanism/System Powered (15′′)	Ports Initialized
Left Drive Motors	2, 3, 5, 6
Right Drive Motors	11, 12, 19, 20
IMU (inertial measurement unit) Sensor	13

Mechanism/System Powered (24′′)	Ports Initialized
Left Drive Motors	1, 3, 4, 21
Right Drive Motors	11, 14, 19, 20
IMU (inertial measurement unit) Sensor	15
Encoder Sensor	5, 6

```
// Negative numbers mean reversed motor
#define LEFT_MOTORS -2, -3, -6, -5 // left drive motors
#define RIGHT_MOTORS 11, 12, 19, 20 // right drive motors
#define GEARSET pros::E_MOTOR_GEAR_600 // RPM of chassis motors

// Ticks per inch
#define TPI 45.7   // Encoder ticks per inch of forward robot movement
#define MIDDLE_TPI 1 // Ticks per inch for the middle wheel

// Tracking wheel distances
#define TRACK_WIDTH 9.75 // The distance between left and right wheels (or tracker wheels)
#define MIDDLE_DISTANCE 0 // Distance from middle wheel to the robot turning center

// Sensors
#define IMU_PORT 13                          // Port 0 for disabled
#define ENCODER_PORTS 0, 0, 0               // Port 0 for disabled,
#define EXPANDER_PORT 0                      // Port 0 for disabled
#define ENCODER_TYPE arms::odom::ENCODER_ADI // The type of encoders

// Movement tuning
#define SLEW_STEP 8             // Smaller number = more slew
#define LINEAR_EXIT_ERROR 0.3   // default exit distance for linear movements
#define ANGULAR_EXIT_ERROR 1    // default exit distance for angular movements
#define SETTLE_THRESH_LINEAR .5 // amount of linear movement for settling
#define SETTLE_THRESH_ANGULAR 1 // amount of angular movement for settling
#define SETTLE_TIME 150         // amount of time to count as settled
#define LINEAR_KP 6 // Linear Proportion constant for PID
#define LINEAR_KI 0 // Linear Integral constant for PID
#define LINEAR_KD 50 // Linear Derivative constant for PID
#define TRACKING_KP 40 // point tracking turning strength
#define TRACKING_KD 200 // point tracking turning strength
#define ANGULAR_KP 1.25 // Angular Proportion constant for PID
#define ANGULAR_KI 0.01 // Angular Integral constant for PID
#define ANGULAR_KD 8 // Angular Derivative constant for PID
#define MIN_ERROR 5 // Minimum distance to target before angular componenet is disabled
#define LEAD_PCT .6 // Go-to-pose lead distance ratio (0-1)

#define MIN_LINEAR_SPEED 5 // lowest linear speed in which robot should go
#define MIN_ANGULAR_SPEED 5 // lowest angular speed in which robot should go
```

Figure 3: Motor & Sensor Initialization + Control Constants for 15″ Robot

```
// Negative numbers mean reversed motor
#define LEFT_MOTORS -4, -21, -1, -3 // left drive motors
#define RIGHT_MOTORS 11, 14, 19, 20 // right drive motors
#define GEARSET pros::E_MOTOR_GEAR_600 // RPM of chassis motors

// Ticks per inch
#define TPI 323.33   // Encoder ticks per inch of forward robot movement
#define MIDDLE_TPI 1 // Ticks per inch for the middle wheel

// Tracking wheel distances
#define TRACK_WIDTH 9.75 // The distance between left and right wheels (or tracker wheels)
#define MIDDLE_DISTANCE 0 // Distance from middle wheel to the robot turning center

// Sensors
#define IMU_PORT 15                          // Port 0 for disabled
#define ENCODER_PORTS 5, 5, 0                // Port 0 for disabled,
#define EXPANDER_PORT 0                      // Port 0 for disabled
#define ENCODER_TYPE arms::odom::ENCODER_ADI // The type of encoders

// Movement tuning
#define SLEW_STEP 8             // Smaller number = more slew
#define LINEAR_EXIT_ERROR 0.2   // default exit distance for linear movements
#define ANGULAR_EXIT_ERROR 1    // default exit distance for angular movements
#define SETTLE_THRESH_LINEAR .5 // amount of linear movement for settling
#define SETTLE_THRESH_ANGULAR 1 // amount of angular movement for settling
#define SETTLE_TIME 250         // amount of time to count as settled
#define LINEAR_KP 4 // Linear Proportion constant for PID
#define LINEAR_KI 0 // Linear Intergral constant for PID
#define LINEAR_KD 12 // Linear Derivative constant for PID
#define TRACKING_KP 60 // point tracking turning strength
#define ANGULAR_KP 2.3 // Angular Proportion constant for PID
#define ANGULAR_KI 0 // Angular Integral constant for PID
#define ANGULAR_KD 20 // Angular Derivative constant for PID
#define MIN_ERROR 5 // Minimum distance to target before angular componenet is disabled
#define LEAD_PCT .6 // Go-to-pose lead distance ratio (0-1)
```

Figure 4: Motor & Sensor Initialization + Control Constants for 24″ Robot

Specific and key coding sections for autonomous triball loading and scoring for the match autonomous can be found below:

```
// match load
for (int i = 0; i < 10; i++) { // loop to match load in autonomous (run till 10 triballs are introduced)
  wings::close_left(); // retract left wings to knock out triball in loading zone
  pros::delay(300); // wait for 300 ms for wing to fully close
  wings::open_left(); // open left wings to prepare for next triball
  pros::delay(800); // wait for 800 ms for wing to open & next triball to be in loading zone
}
wings::close_left(); // retract left wings
pros::delay(800);

// push balls accross
chassis::move({18, 10}, 50, arms::REVERSE); // move forward slightly to line up at alley
chassis::turn(90, 75); // turn around to push & line up for the alley
wings::open_right(); // activate right wings
intake::move(-35); // out take to prevent triball from entering robot
chassis::move({8, 75}, 45, 5, arms::THRU); // push through the alley
chassis::move({8, 95}, 45, 5);  // push through alley
chassis::turn(135, 50); // turn slighly to line with goal
wings::open_left(); // activate left wings
chassis::move(28, 35); // move towards the goal
chassis::turn(180, 50); // line up with the goal

// Push into goal x3
for (int i = 0; i < 3; i++) { // repeat three times to push triballs in
  chassis::move(10, 80, 3, arms::THRU); // drive forward and push
  wings::close_right(); // retract right wings
  pros::delay(200);
  chassis::arcade(-50, 0); // drive backward
  pros::delay(500);
  chassis::arcade(0, 0); // stop movement
  wings::open_right(); // activate right wings
}
```

Figure 5: “Bowling” and Scoring of triballs for 15′′ Robot

```
// score triball from the middle
chassis::turn(180); // turn away from the goal
chassis::move({36, 24}, 60); // move towards the center of the field
chassis::turn(90); // align with triball
intake::move(100); // intake triball
chassis::move(24, 80); // move towards triball
intake::move(0); // stop intake as triball in robot
chassis::turn(0); // align with goal
pros::delay(200);
chassis::move(18); // move towards goal
intake::move(-100); // out take triball into goal
pros::delay(500);
intake::move(0); // stop out taking
chassis::move(20); // push triball into goal
chassis::move(-24, arms::REVERSE); // back up after triball is scored into goal
```

Figure 6: Intaking & Scoring triball from field for 24′′ Robot

Comp Software: MCCC Skills Autonomous

Date	@January 23, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Mihir Laud Alex Lam
Design Cycle	Design Cycle 2

Objectives

- [ ] Plan Skills Autonomous for both 15″ & 24″ robots
- [ ] Create flowcharts of autonomous program(s)
- [ ] Implement autonomous code

Planning

The main objective for the skills autonomous for both bots is to utilize the "bowling" strategy to be able to maximize the points scored in autonomous. To do this, both robots will park at the loading zone and group the triballs near each other. **This will maximize the number of triballs that will get pushed across with the wings, as the grouped triballs provide the wing with a larger surface area to contact and push.** After pushing the triballs across, the 24″ robot will latch on and lift itself on the horizontal parking bar, securing a "C" tier hang. Before programming and testing the autonomous on the robots, a route is planned and mapped.

Below is the mapped plan for the autonomous routes both robots will take:

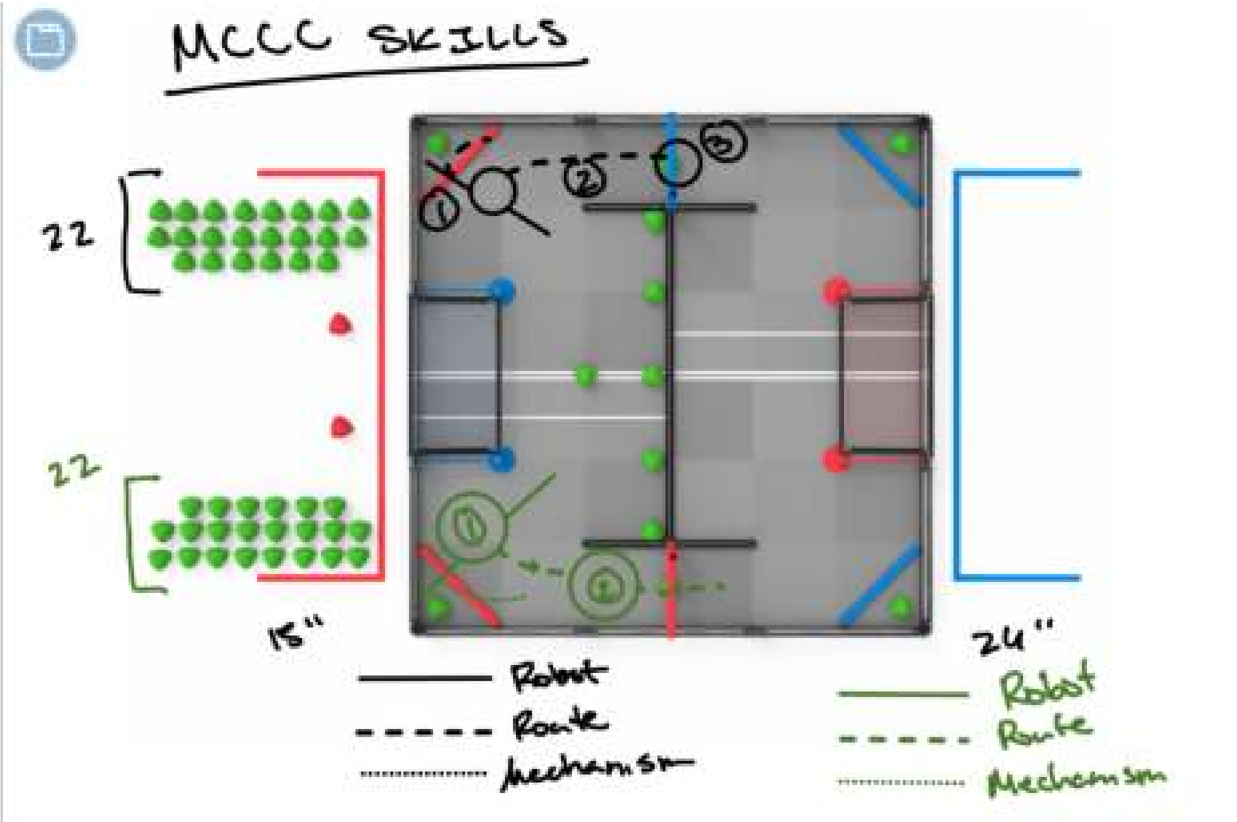


Figure 1: Skills Autonomous Route for Both Robots (15″ & 24″)

Following the mapped plan, a flowchart was created for both robots to convert the route to pseudocode (code procedure):

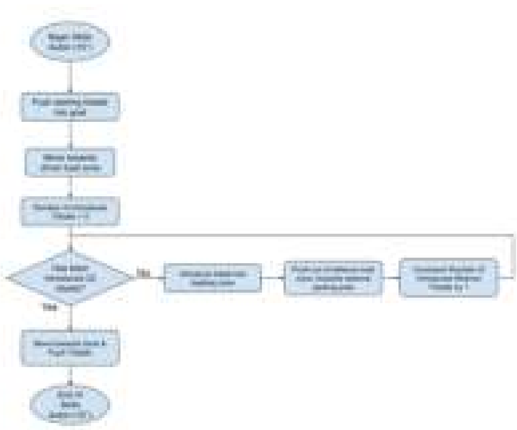


Figure 2a: Flowchart for Skills Autonomous Route for 15″ Robot

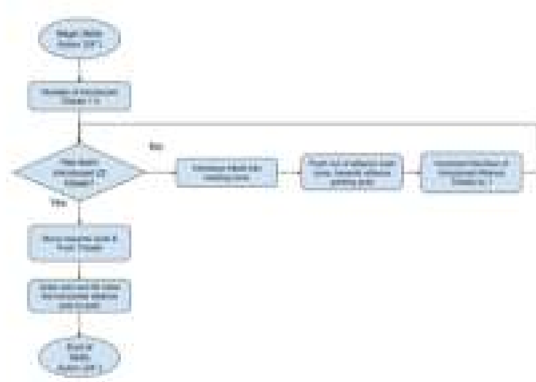


Figure 2b: Flowchart for Skills Autonomous Route for 24″ Robot

Following the flowcharts, implementation of the logic was applied to the robot for competition use. **Initialization of the motors was first done before odometry was utilized to program the robot movements of the route via our ARMS library.**

Mechanism/System Powered (15'')	Ports Initialized
Left Drive Motors	2, 3, 5, 6
Right Drive Motors	11, 12, 19, 20
IMU (inertial measurement unit) Sensor	13

Mechanism/System Powered (24'')	Ports Initialized
Left Drive Motors	1, 3, 4, 21
Right Drive Motors	11, 14, 19, 20
IMU (inertial measurement unit) Sensor	15
Encoder Sensor	5, 6

```
// Negative numbers mean reversed motor
#define LEFT_MOTORS -2, -3, -6, -5 // left drive motors
#define RIGHT_MOTORS 11, 12, 19, 20 // right drive motors
#define GEARSET pros::E_MOTOR_GEAR_600 // RPM of chassis motors

// Ticks per inch
#define TPI 45.7   // Encoder ticks per inch of forward robot movement
#define MIDDLE_TPI 1 // Ticks per inch for the middle wheel

// Tracking wheel distances
#define TRACK_WIDTH 9.75 // The distance between left and right wheels (or tracker wheels)
#define MIDDLE_DISTANCE 0 // Distance from middle wheel to the robot turning center

// Sensors
#define IMU_PORT 13                         // Port 0 for disabled
#define ENCODER_PORTS 0, 0, 0              // Port 0 for disabled,
#define EXPANDER_PORT 0                     // Port 0 for disabled
#define ENCODER_TYPE arms::odom::ENCODER_ADI // The type of encoders

// Movement tuning
#define SLEW_STEP 8             // Smaller number = more slew
#define LINEAR_EXIT_ERROR 0.3   // default exit distance for linear movements
#define ANGULAR_EXIT_ERROR 1    // default exit distance for angular movements
#define SETTLE_THRESH_LINEAR .5 // amount of linear movement for settling
#define SETTLE_THRESH_ANGULAR 1 // amount of angular movement for settling
#define SETTLE_TIME 150         // amount of time to count as settled
#define LINEAR_KP 6 // Linear Proportion constant for PID
#define LINEAR_KI 0 // Linear Integral constant for PID
#define LINEAR_KD 50 // Linear Derivative constant for PID
#define TRACKING_KP 40 // point tracking turning strength
#define TRACKING_KD 200 // point tracking turning strength
#define ANGULAR_KP 1.25 // Angular Proportion constant for PID
#define ANGULAR_KI 0.01 // Angular Integral constant for PID
#define ANGULAR_KD 8 // Angular Derivative constant for PID
#define MIN_ERROR 5 // Minimum distance to target before angular componenet is disabled
#define LEAD_PCT .6 // Go-to-pose lead distance ratio (0-1)

#define MIN_LINEAR_SPEED 5 // lowest linear speed in which robot should go
#define MIN_ANGULAR_SPEED 5 // lowest angular speed in which robot should go
```

Figure 3: Motor & Sensor Initialization + Control Constants for 15′′ Robot

```
// Negative numbers mean reversed motor
#define LEFT_MOTORS -4, -21, -1, -3 // left drive motors
#define RIGHT_MOTORS 11, 14, 19, 20 // right drive motors
#define GEARSET pros::E_MOTOR_GEAR_600 // RPM of chassis motors

// Ticks per inch
#define TPI 323.33   // Encoder ticks per inch of forward robot movement
#define MIDDLE_TPI 1 // Ticks per inch for the middle wheel

// Tracking wheel distances
#define TRACK_WIDTH 9.75 // The distance between left and right wheels (or tracker wheels)
#define MIDDLE_DISTANCE 0 // Distance from middle wheel to the robot turning center

// Sensors
#define IMU_PORT 15                          // Port 0 for disabled
#define ENCODER_PORTS 5, 5, 0                // Port 0 for disabled,
#define EXPANDER_PORT 0                      // Port 0 for disabled
#define ENCODER_TYPE arms::odom::ENCODER_ADI // The type of encoders

// Movement tuning
#define SLEW_STEP 8             // Smaller number = more slew
#define LINEAR_EXIT_ERROR 0.2   // default exit distance for linear movements
#define ANGULAR_EXIT_ERROR 1    // default exit distance for angular movements
#define SETTLE_THRESH_LINEAR .5 // amount of linear movement for settling
#define SETTLE_THRESH_ANGULAR 1 // amount of angular movement for settling
#define SETTLE_TIME 250         // amount of time to count as settled
#define LINEAR_KP 4 // Linear Proportion constant for PID
#define LINEAR_KI 0 // Linear Intergral constant for PID
#define LINEAR_KD 12 // Linear Derivative constant for PID
#define TRACKING_KP 60 // point tracking turning strength
#define ANGULAR_KP 2.3 // Angular Proportion constant for PID
#define ANGULAR_KI 0 // Angular Integral constant for PID
#define ANGULAR_KD 20 // Angular Derivative constant for PID
#define MIN_ERROR 5 // Minimum distance to target before angular componenet is disabled
#define LEAD_PCT .6 // Go-to-pose lead distance ratio (0-1)
```

Figure 4: Motor & Sensor Initialization + Control Constants for 24″ Robot

Specific and key coding sections of the “bowling” strategy implemented for the skills autonomous can be found below:

```cpp
//open & close wing to push triball out of alliance load zone
for (int i = 0; i < 23; i++) { // introduce 22 triballs from loading zone
        wings::close_left(); // retract left wings to knock out triball in loading zone
        pros::delay(300); // wait for 300 ms for wing to fully close
        wings::open_left(); // open left wings to prepare for next triball
        pros::delay(800); // wait for 800 ms for wing to open & next triball to be in loading zone
  }

intake::move(-100); // out take to prevent triballs from entering robot
wings::open_right(); // open right wings
chassis::turn(-180, arms::THRU); // rotate robot to have wings in front
chassis::turn(-90); // line up for alley
chassis::move({-27, -80}, 65); // push triballs through the alley
wings::close_left(); // retract left wing
wings::close_right(); // retract right wing
chassis::turn(90); // turn around to face away from triballs
chassis::move(10, 35); // move away from triballs to avoid touching them
intake::move(0); // stop out taking
```

Figure 5: “Bowling” of triballs across field for 15′′ Robot

```cpp
//note: only 18 triballs were introduced to allocate time for robot hang
for (int j = 0; j < 6; j++) { // complete 6 cycles of 3 triballs each (18 total)
    for (int i = 0; i < 3; i++) { // load 3 triballs per cycle
        chassis::turn(-135, 50); // turn to knock triball out of loading zone
        chassis::turn(-45, 50); // return back to original position to prepare for next ball
        pros::delay(300);
    }

    double dy = odom::getPosition().y; // determine rate of change of error from movement

    chassis::move(0.7071 * dy, (dy < 0) ? arms::REVERSE : arms::NONE); // correct the change in error
    chassis::turn(-45); // line up with loading zone
}

wings::toggle(wings::RIGHT); // activate right wings
elevation::raise(); // raise lift for hang
chassis::turn(105); // align with alley
pros::delay(1000);
intake::move(-100); // out take to prevent triballs from entering robot
chassis::move({-27, 50}, 60); // move towards
chassis::turn(90); // rotate to have hang mechanism lined up with parking bar
chassis::move(10, 60); // drive forward to hang
intake::move(0); // stop intake
elevation::lower(); // power lift to climb for hang
```

Figure 6: “Bowling” of triballs across field + parking hang for 24′′ Robot

Design and Build: Wings Iteration 2

Date	@January 23, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Ⓒ Conner Siebert Ⓓ Dominic H
Design Cycle	Design Cycle 2

Goals

- Following our discussion on the need to change from vertical wings to horizontal wings, it was decided the **same wings will be used on both the 15″ and 24″ robots.**
- A key goal of the wings is to be able **pop match load triballs over the match load barrier**, to get the triballs into play
- To be able to tune the length of the wings to be able to stay within the 36″ requirement and to be able to **pass under the elevation bar with 1 wing extended.**

Building the Wings

- Step 1
    - A **5-long c-channel is attached to the front of the chassis**, outside of the chassis sleds.
- Step 2
    - The **c-channel is boxed with a screw running through the metal and spacers**, to prevent the member from bending in the flanges.
- Step 3
    - A **screw joint is attached to the c-**

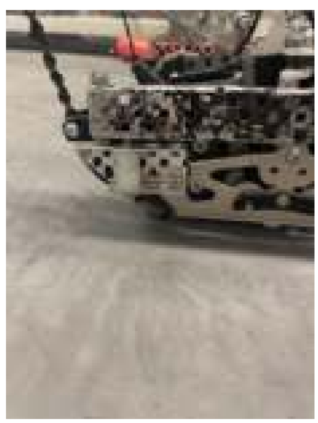


Figure 1: Wing mount on the side of the chassis

Figure 2: Piston mounted to the hang mechanism towers

channel, which passes through the wing c-channels

- Step 4
  - The **Piston is mounted on a 30 degree gusset** attached to the elevation mechanism towers
  - A **3D printed bracket is used to mount the piston** onto the gussets, on a screw joint
- Step 5
  - The **piston is attached to the top of the wing c-channel** using yet again another screw joint.
- Step 6
  - A **rubber band is attached from the chassis to the wings** to retract the wing once the single acting solenoid releases the air from the piston

Figure 3: Piston mounted to the wing

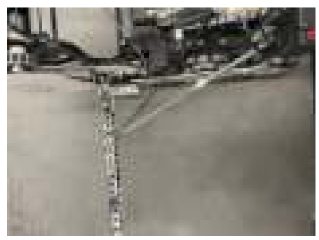


Figure 5: Top view of extended wing

Figure 4: Side view of extended wings

Test Solution

To ensure that the wings successfully deploy quickly, we will test using a pass/fail system to ensure successful deployment. **The specific testing procedure is outlined below:**

1. Ensure the wings are in the default, retracted position.
2. Activate the pneumatics using the brain screen.
3. Check that the wings have deployed to their fullest extent.
4. Close the wings using the pneumatic control on the brain screen.

5. Repeat the process four additional times.

Results

	Successful Deploy (Y/N)
Test 1	Y
Test 2	Y
Test 3	Y
Test 4	Y
Test 5	Y
Average	Yes (5/5)

As the wings are able to deploy quickly and without issue, **we can conclude that they successfully meet our criteria.**

MCCC Tournament Recap and Strategy Adjustment

Date	@January 28, 2024
Category	15" Robot 24" Robot BLRS2 Game Analysis
Authors	Kathleen Lowe Alex Lam
Design Cycle	Design Cycle 3

BLRS2 after winning Excellence at MCCC

On , January 27th, 2024, we attended the MCCC Solar Over Under VEXU Tournament in Monroe, Michigan, which had a total of 8 competing.

Performance

Matches

Overall Summary

Rank	1	Record	W-L-T
WP	12	CCWM	96.17
AP	32	OPR	127.5

SP	188	DPR	31.33

Outcome: Tournament Finalists

Individual Matches

Q1 - WIN	Score
BLRS2	154
KUDOS	35

Q4 - WIN	Score
BLRS2	145
VALPO	0

Q8 - WIN	Score
BLRS2	85
BLRS	73

Across our first 3 matches, we faced several issues with reliability in our deployments during the autonomous period as well as the autonomous runs themselves being untuned. **We were able to tweak these for future matches.**

Q11 - WIN	Score
BLRS2	153
WMU	35

Q15 - WIN	Score
BLRS2	128
LCC2	0

Q20 - WIN	Score
BLRS2	100
MSU	45

The second half of our qualification matches were very successful. **We scored multiple AWPs and had no major issues.**

SF 1-1 - WIN	Score
BLRS2	143
KUDOS	15

F1 - WIN	Score
BLRS2	115
MSU	65

F2 - LOSS	Score
BLRS2	75
MSU	85

F3 - LOSS	Score
BLRS2	85
MSU	88

We had a strong run through elimination matches until F2/3. We ran into several pinning issues by MSU, as well as them violating SG11 multiple times. We learned that we need to clarify how rules will be called by the HR before matches in the future, and also **rely less on our climbs as they may be blocked by other teams.**

Robot Skills Challenge

Rank	**1**	**Total Score**	229
Driver Score	128	Programming Score	101
Driver Attempts	2	Programming Attempts	1

After reaching our goals initially, we retried driver skills to improve our chances. However, our initial scores ended up being the highest of the day.

Subsystems

Drivetrain

On both robots, **our chassis performed more highly than our previous experience at NUKE**. We remedied all of our previous beaching issues, and while we were slower driving over the barrier than expected, we are overall incredibly satisfied with their performance.

Intake

During several matches, including the second match of finals, **the intake on our 15″ robot did not deploy.** This issue, and the time necessary to fix it during the matches, severely impeded our scoring during several matches.

Wings

We used our wings mainly for match loading, which was highly effective. Moving forward, we plan to optimize both the geometry of our wings and the pneumatic system that they are attached to.

Elevation

On both of our robots, our hanging mechanisms performed sufficiently. However, other issues on our robot, mostly match-time injuries, impeded with our barrier parking abilities. We recognize that while our hangs worked well, **our strategic approach to the elevation portion of the game may need to be updated before the world championship.**

Takeaways

- Match Strategy:
  - **Defending elevation is a feasible strategy moving forward.**
  - The durability of our bots may be an issue moving forward, as we had a tendency to flip.
  - Our shuttling, while effective, could be improved or changed.
- Robot Skills Challenge:
  - **Our current strategy is viable if we stay on top of our time management.**
  - Skills went as expected, so we are confident in our abilities.
- Subsystems:
  - We may change the wing geometry in order to improve effectiveness.
  - **We may improve our elevation in order to become a more competitive robot at the world championship.**

    - We will reinforce the pneumatics on our robots to provide more air pressure to both our wing subsystem and our elevation subsystem.
- We should consider a robot able to de-score from the opposing alliance goal
    - We notice in several matches, opponents would double-zone for long periods of time. **With this, moving forward we would like to design robots with a height less than 6" to test the viability of this strategy**
    - Even if the opponents don't double-zone, we could force them to be more cautious with their driving. With the threat of getting their own alliance goal cleared in seconds, opposing drivers will be on nails for the entirety of drive control.

Task and Timeline Update: February

Date	@January 31, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Ⓒ Conner Siebert
Design Cycle	Design Cycle 3

January Recap

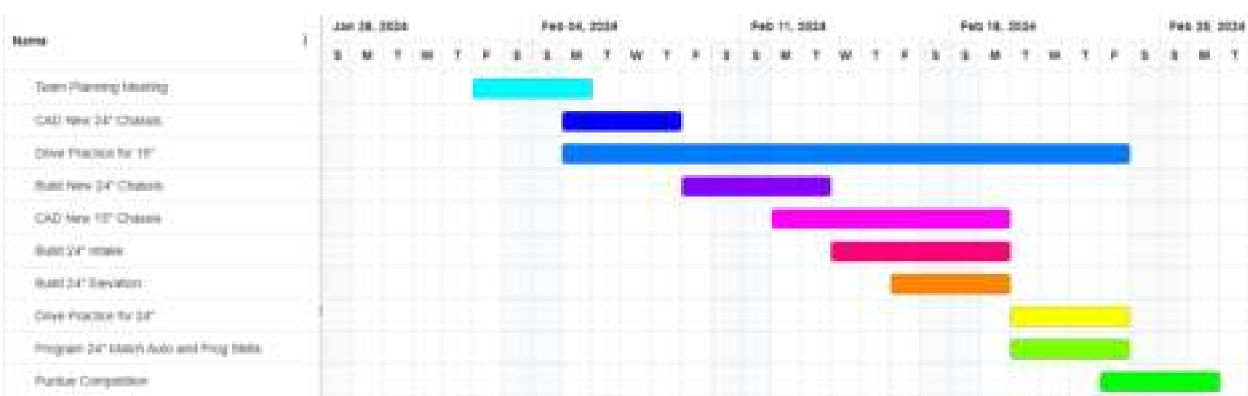
January was a very successful month for the BLRS2 team. **To begin the month, our team ended Christmas break and returned to campus on January 8th.** In the follow three weeks, we accomplished the following:

- Week 1
  - Week 1 was spent at home, with little work completed on the robots.
- Week 2
  - In the week leading up to the NUKETown competition, we were able complete the first iterations of our robots. **At the completion, we found several flaws in our design and strategies, leading to the changes to come in weeks 3 and 4 of the month.**
- Week 3
  - Following the NUKETown competition, our entire team set down for a strategy and planning meeting for the remainder of the month. **This outlined the plan for the next two weeks**, evolving around mechanical and programming changes to the robots.
  - Throughout this week, the **mechanics team was able to successfully implement changes to several subsystems on both robots.** The wings, elevation, and intakes were modified to be in a better working fashion.
- Week 4
  - This week leading up to the MCCC competition was spent on two main goals: driver practice and programming the robots. **Both goals were accomplished, as seen by our high level of performance in the robot skills challenge at MCCC.** With a 229 total, we were able to post the 4th highest skills in Vex U.

February Task and Timeline

Following the MCCC tournament, **our team's goal for for February are focused around preparing for both the Purdue competition and worlds.** All progress made on the robots moving forward should focus on the worlds robots. The main timeline of February are:

- February 5th
  - On this Monday, the team will meet and have a **large discussion on the plans for the post MCCC strategy adjustment plans** and how we will follow the timeline set for February
- February 8th
  - As we begin to redesign the 24″ robot, designing the chassis in CAD is the first priority. **This is the deadline to finish the CAD, so we can begin to build all required subsystems for Purdue**
- February 8th-18th
  - **This 10 day stretch will be the target time to rebuild the 24″ robot.** In this time, we should complete the new chassis, make further decisions on the intake and elevation subsystems, and be ready to drive practice and program the robot.
- February 18th-22nd
  - This week is the final before the Purdue competition. Therefore, we feel **driver practice and programming time should be advocated to prepare for the competition.**
- February 23rd
  - This is the day we host the Purdue Vex U Slam and Jam, the biggest Vex U event in the world outside the world championship. **With over 30 teams in attendance, ensuring we are on the top of our game for this event is a major priority.** It will give a chance to compete on a big stage before the world championship in April.



Design Brief: Iteration 3

Date	@February 3, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Stephen Hohnholt Jacob Zawacki
Design Cycle	Design Cycle 3

BLRS2 Robots: Iteration 3

Problem Statement

These robots are going to need to play and win in VRC Over-Under, meaning that they need to meet all constraints of the game manual and score more points than the opposing alliance. These robots need to be able to score Triballs quickly and efficiently in the alliance goal, in both the autonomous period as well as the driver control period. This includes picking them up off the floor and from the match-load zone. They then need to move them towards the offensive zone, and then finally score them in the alliance zone. They will also have a high-tier elevation for more points.

Design Statement

These robots will be designed to have a chassis, intake, wings, and hanging mechanism. The chassis will allow them to move around the field. The intake and wings will allow the robots to pick up Triballs quickly off the floor and in the match load zone, and then score them. The hanging mechanism is needed to achieve a high tier elevation for more points. In addition, the robots are designed to be able to capitalize on the double-zoning clause of Rule SG8.

Constraints

- Begin within the set size constraints
  - One robot starting within a 24×24×24 inch box
  - One robot starting within a 15×15×15 inch box
- Intake Triballs from the match load zone
- Score Triballs into the alliance goal

- Elevate the robot onto the elevation bars
- Pop balls over the center barrier

Criteria

- Be able to intake or pop Triballs from the match load zone at a rate of 1 Triball per second
    - This will allow us to **quickly score all 10 allowable match loads** during the autonomous period, as well as allow us to quickly intake in driver
- Efficiently shuttle Triballs across the field to our scoring goal
    - To do this, we will need to have efficient drives, quick and secure intakes, and a large amount of drive practice going into tournaments
    - By being able to effectively shuttle, **we will be able to score Triballs without fear of losing them to other teams**
- Achieve elevation at the end of the match
    - While the elevations on our previous design cycle worked well, **we will need to elevate higher to remain competitive at our home competition and the World Championship**
    - We will decide specifics in a future entry this Spring, but the robots will **tentatively aim for at least an H tier elevation**
- Have the ability to descore from the opposing goal
    - Both robots should have the capability of descoring from the opposing goal, allowing us to either detract from their points or **force our opponents to play in an unfamiliar manner**
    - To do this, **both robots will need to remain under 6 inches in total height**

Important Deadlines

- 2/23/2024: Complete 24″ Robot Changes
- 3/16/2024: Complete 15″ Robot Changes
- 4/1/2024: Finalize Elevation Mechanisms
- 2/24/2024: Purdue Competition
    - Robots should be fully built and programmed.

    - Hang may not be included to prevent other teams from seeing our complete strategy.
- 4/28/2024: World championship
    - Robots should be fully built and programmed.

Brainstorming and Decision Matrix: Chassis Redesigns

Date	@February 7, 2024
Category	15" Robot 24" Robot BLRS2
Authors	Conner Siebert Jacob Zawacki
Design Cycle	Design Cycle 3

Identify the Problem

After competing at MCCC on January 27th, our team decided to re-asset the viability of the current chassis. This is due to the following considerations:

- Poor barrier crossing
  - As the current chassis crosses the barrier, **we feel an alternative design could cross the barrier better**
- Unable to reconfigure the speed and gear ratios
  - As the current chassis is full custom, there is often a lack of ability to make the chassis faster.
  - The current strategy heavily relies on a fast shuttle of triballs from the match load zone to the goal, **a strong acceleration and high velocity would be optimal**

We have decided from the post MCCC competition strategy discussion to have this robot's chassis have the following design considerations and criteria:

- Able to easily cross the long barrier
  - As our new offensive strategy revolves around shuttling triballs around from the match load zone to the goals, **the ability to go both under the elevation bar and over the long barrier will allow us to avoid defense**
- Designed for 8 motors, with the ability to add a 9th and 10th
  - As our team is unsure of how many motors will be required for other subsystems on this robot, **designing for either motor configurations is important**
- Ability to potentially PTO an elevation mechanism to the chassis motors
  - As our team is developing potential higher tier elevations, **powering the potential elevation from the chassis should be an option**

Given our new strategy developments following NUKE and MCCC, we believe each robot should have individual characteristics. The chassis for each robot will be designed with the following criteria:

24" Defensive Robot

As the **24" robot aims to focus on playing in the defensive zone of the field, preventing opponents from scoring triballs** into their alliance goals will be important. Furthermore, this robot should maintain the ability to shuttle triballs when necessary, therefore will need to be able to pass under the elevation bar and over the long barrier. The following features are criteria that should be meet:

- Ability to push other robots
  - As we aim to prevent opponents from scoring, having a chassis with enough torque to oppose robots would **allow for more effective defense**
- Strong barrier crossing

- With both robots, crossing the barrier is the largest challenge this year's field poses to the chassis design
- Low center of gravity
  - As this robot is going be pushing opposing robots, **our robot flipping over is a concern.** A lower center of gravity would help prevent this from occurring

15″ Offensive Robot

The 15″ robot's main goal for a majority of matches should be to take match loaded triballs from the match load zone to the alliance goal. To achieve this, the following characteristics would be desired:

- High speed and acceleration
  - As the **robot aims to shuttle triballs straight from the match load zone to the alliance goal**, having a chassis that can quickly move from one side of the field to the other is necessary
- High maneuverability
  - As we suspect this robot will face defense while shuttling, **drifting and quick turns would help us escape such defense**
- Ability to climb into the match load zone
  - As we plan to prototype and develop future strategies to get triballs out of the match load zone, a possible solution is to **drive into the zone, intake a Triball, and reverse out**
  - It will be important to ensure the chassis can't get stuck in the match load zone

Brainstorm Solutions

Solution 1: Keep the Current Chassis

- By keeping the existing chassis, our team would **save time when developing further subsystems**
- Our current chassis is battle tested, proving to be viable in matches at NUKE and MCCC
- The speed cannot greatly increased or decreased

Figure 1: Solution 1: Existing Chassis

Solution 2: Two 3.25″ Wheels with raised Tractions

- The first alterative chassis design removes a center wheel from the current chassis configuration
  - To still clear the barrier, two flex wheels would be added to the center of the chassis.
    - These wheels will spin with the drive wheels
    - They will grab the barrier while crossing, **preventing dead zones and places where the chassis can get stuck on the barrier**

- A limitation of this chassis is the inability to add a center traction wheel
    - As there are only 2 wheels per side on this chassis design, it will be required that all wheels are omni to maintain maneuverability

Figure 2: Solution 2 Chassis Prototype

Solution 3: Four 3.25″ Wheels per side

- By increasing the number of wheels per side of the chassis, **the robot gains additional configurability in where the traction wheel is placed,** in the 2nd or 3rd wheel spot based on the center of gravity
- This chassis could be smoother crossing the long barrier, as the additional wheel would allow for more constant contact across the short barrier
- The increase in wheels would mean an increase in gears, possibly leading to increased friction issues

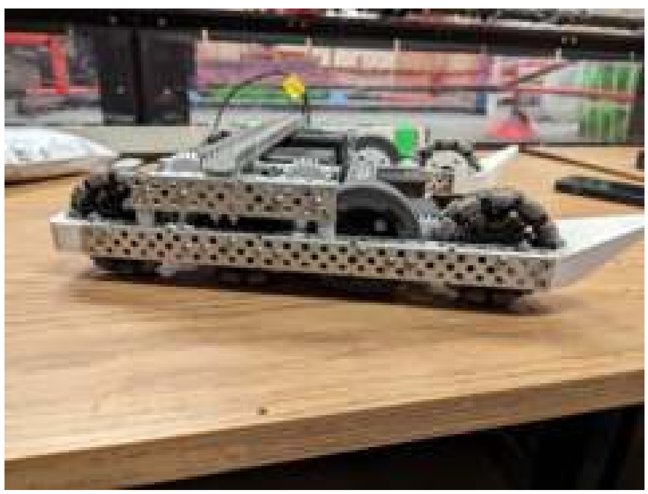


Figure 3: Solution 3 prototype

Select the Best Solution

Pros

Keep the Current Chassis	Two 3.25″ Omni Wheels with raised Tractions per side	Four 3.25″ Wheels per side
Would allow time to be spent on other subsystems	Strong maneuverability with 4 omnis	Ability to configure wheels (omnis and tractions)
Has stood up to competition	Able to drive into the match load zone to intake triballs	Able to drive into the match load zone to intake triballs
Is able to cross the barrier	Keeps the robot lower while crossing the long barrier	Additional traction
	Can have a high acceleration	

Cons

Keep the Current Chassis	Two 3.25″ Omni Wheels with raised Tractions per Side	Four 3.25″ Wheels per side
Wheel configuration can't be changed	Lack of pushing power	Less maneuverability than the two omnis
Speed can't be changed	Can't have a traction wheel	Additional weight with extra wheels
Would get stuck in match load zone while intaking		Slightly worst at crossing the long barrier
Difficulties mounting subsystems to the chassis		

24″ Defensive Robot Chassis Decision Matrix

To decide upon the desired chassis for the 24″ robot, our team considered the following criteria and scored each option.

- Time Required (x/3)
  - As we prepare to design and build robots for the world championship, saving complexity and time for the robot is considered
  - A higher score indicates less time required
- Torque (x/5)
  - As pushing other robots is a key criteria for the this robot, higher torque from the motors would allow for more pushing power
  - A higher score indicates higher torque
- Traction (x/8)
  - The biggest priority for pushing opposing robots is the traction our wheels can get with the field tiles
  - A higher score indicates more traction
- Speed (x/3)
  - While this robot is set to be defensive, still maintaining the ability shuttle triballs quickly is considered
  - A higher score indicates faster speed
- Customization (x/5)
  - As stated in the current chassis concerns, moving forward our team would like the option to reconfigure the drive speed and wheel configuration
  - A higher score indicates a more customizable chassis

	Time Required (x/3)	Torque (x/5)	Traction (x/8)	Speed (x/3)	Customization (x/5)	Total (x/24)
Keep the Current Chassis	3	3	6	2	1	15
Two 3.25″ Omni Wheels with raised Tractions per Side	1	3	2	3	3	12
Four 3.25″ Wheels per side	1	5	8	1	5	20

After considering each of the three solutions, our team has decided to move forward with the four 3.25″ wheels per side. This is because of the following considerations:

- The ability to add traction wheels
    - The wheel configuration for this chassis will be 3 3.25″ omni wheels with 1 traction wheel located in the 2nd wheel slot
    - **This chassis can be configured in the future to place the traction wheel near the center of gravity.** That is because the when turning, the center of rotation should be around the center traction wheel
- Customized gear ratio
    - As we are designing our chassis, the ability to test different speeds and torque ratios will be helpful
- Strong barrier crossing
    - As this robot will have an additional wheel compared to the previous version of the chassis, barrier crossings will be smoother than previously

15″ Offensive Robot Chassis Decision Matrix

To decide upon the desired chassis for the 24″ robot, our team considered the following criteria and scored each option.

- Time Required (x/3)
    - As we prepare to design and build robots for the world championship, saving complexity and time for the robot is considered
    - A higher score indicates less time required to build
- Acceleration (x/5)
    - As leaving the match load zone quickly with triballs is desired to avoid possible defense
    - A higher score indicates faster acceleration
- Traction (x/3)
    - While attempting to score triballs into the alliance goal, traction would allow us to avoid defense, as we could not get pushed from the side
    - A higher score indicates more traction
- Speed (x/8)
    - The main goal of this robot is to be fast, and therefore it is the highest rated criteria
    - A higher score indicates faster speed
- Customization (x/5)

- As stated in the current chassis concerns, moving forward our team would like the option to reconfigure the drive speed and wheel configuration
- A higher score indicates a more customizable chassis

- Maneuverability (x/5)
  - While implementing the shuttle strategy, maneuvering around two main obstacles will be required: the short barriers and opposing robots
  - A higher score indicates a more maneuverable chassis

	Time Required (x/3)	Acceleration (x/5)	Traction (x/3)	Speed (x/8)	Customization (x/5)	Maneuverability (x/5)	Total (x/29)
Keep the Current Chassis	3	2	2	5	1	2	12
Two 3.25″ Omni Wheels with raised Tractions per Side	1	5	1	8	3	5	23
Four 3.25″ Wheels per side	1	3	3	3	5	2	17

After considering all three chassis for the 15″ offensive robot, our team has decided to move forward with the two 3.25″ omni wheels. This is because of the following considerations:

- Best barrier crossing
  - As the two raised traction wheels allow the chassis to stay closer to the ground while crossing the long barrier, **maintaining a lower center of gravity**
- High acceleration and speed
  - As this chassis is lighter than the other two solutions, **we will see better acceleration while maintaining a higher gear ratio**
- Maneuverability
  - As another criteria of the offensive robot, having 2 omni wheels on each side provides the best maneuverability, as **the driver is able to drift around obstacles**

Implementation Plan

- As we would like time to brainstorm, design, and build additional changes, **finishing the chassis redesigns quickly will be a high priority moving forward.**
- Since the chassis redesigns do not require fully custom framing, **we will be able to rapidly iterate on gear ratios on other specifics using 3D-printing.**
- We will aim to complete the 24″ chassis redesign first, so that we always have at least one robot getting drive practice at a time.
- To help keep track of our progress, we will use the following timelines:
    - 24″ Robot
        - **February 7**
            - Have the specifics of the chassis decided upon, including gear ratio and part allocations.
        - **February 9**
            - Complete chassis construction and testing.
        - **February 24**
            - Compete at our home tournament
    - 15″ Robot
        - **Through February 24**
            - Drive Practice
        - **March 15**
            - Complete the chassis redesign
                - This will give us time to **ensure that the 24″ robot is in perfect working order for Purdue competition**, while our 15″ is still serviceable as is.
                - We will be able to **apply things we learned on the 24″ construction** to the 15″, as that will be built well after.

Design and Build: 24″ Chassis

Date	@February 10, 2024
Category	24" Robot BLRS2
Authors	Conner Siebert Jacob Zawacki
Design Cycle	Design Cycle 3

Goals

- After deciding to rebuild the chassis for the 24″ robot, our team set to design and build the new solution.
- To design the chassis, **Inventor will be used to determine the motor and wheel placement, along with generating 3D printed gears**
- After the CAD is completed, the chassis will be built in the lab
- With the chassis built, we will test using the following criteria to determine if the chassis is a successful redesign:
    - Barrier Crossing
        - The chassis must be able to cross the field, ramping the center barrier, in under 3 seconds

Design Process

- Step 1
    - To begin designing the chassis, the desired geometry of the wheels was decided that **the wheels should follow the spacing of a vex 2-wide c-channel**
    - The 3.25″ wheels were placed 7 holes (3.5″) from each other center to center, to allow the chassis to **have contact with the barrier while crossing**

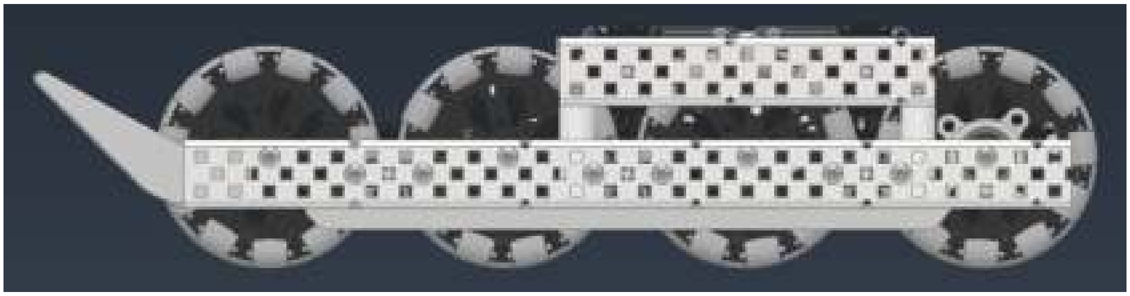


Figure 1: Side View of Chassis

- Step 2
    - The motor placement was decided. Given the chassis was designed for a 10 motor drive, three motors were placed on 24 tooth gears between each wheel. Additionally, two drive motors were added above the rear two motors, to **create a chassis with 5 motors on each side**
- Step 3
    - A 32 tooth gear was designed for the wheels, to create a drive ratio of 450 rpm, as the **gear ratio is 600:450 between the drive gears and wheel gears**
- Step 4
    - **Two c-channel cross braces were placed between each side of the chassis** to hold each side together
- Step 5
    - The following custom 3D prints were created and added to the design:
        - Front sleds
            - To assist the chassis with crossing the center barrier, angled pieces were created. **These sleds allows the robot to lift over the long barrier, as while the chassis will ramp up the pole**
        - Under sleds
            - To assist while crossing the barrier, 3**D printed sleds were added below the drive c-channels.** These pieces protects the drive gears from hitting the barrier, avoiding damage while making the crossing smoother
        - Rear Chassis braces
            - Attached in the same manor as the front sleds, these pieces help create a solid brace

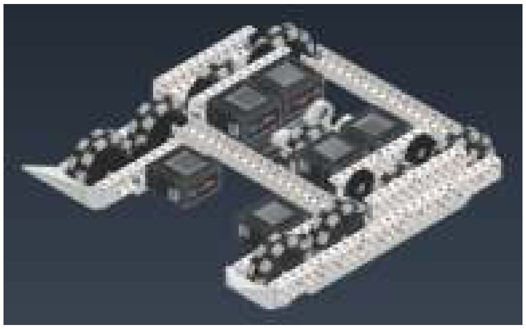


Figure 2: Completed Chassis CAD

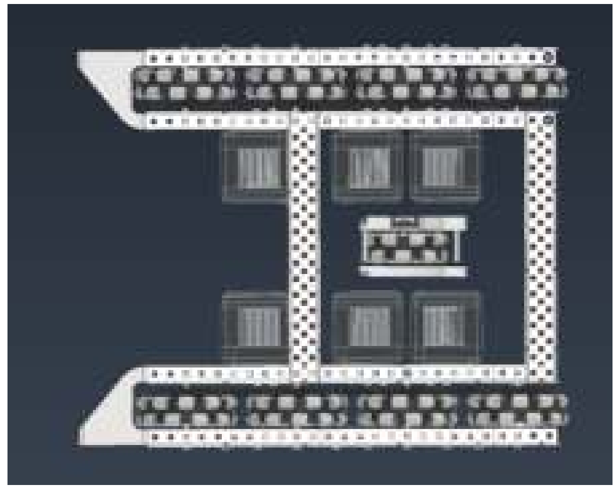


Figure 3: Top View of Completed Chassis CAD

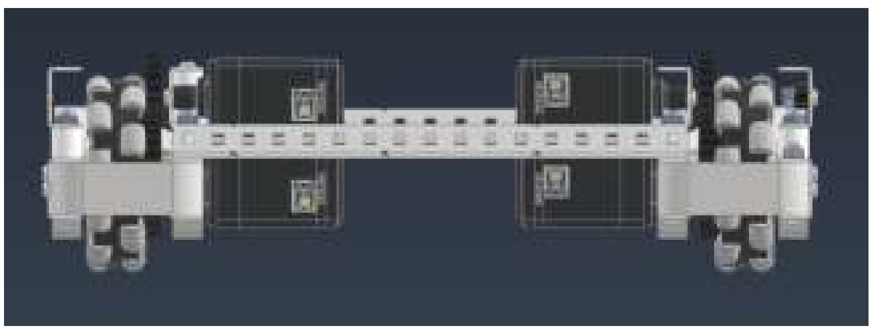


Figure 4: Rear View of chassis

Build the Solution

- Step 1
    - To begin the build process, the **top and bottom drive c-channels were attached with screws and spaced 3/8″ apart.** This process was repeated for the inner and outer support pieces

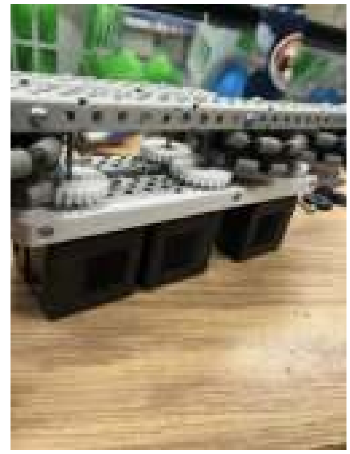
Steps 1 and 2: Inner Chassis Support

Steps 1 and 2: Inner Chassis Support

- Step 2

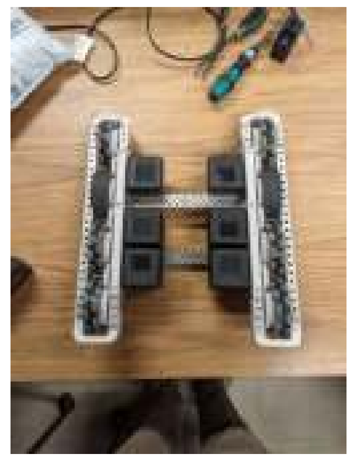
- **Five 11W motors are attached to each side of the chassis.** Inside of each motor a 600 rpm cartridge was installed. This allows the 3600 rpm of the V5 motor to be output to the drive at 600 rpm
- Step 3
  - The outer support pieces were attached to the inner chassis support with the drive motors. This was done by **placing the front sled and rear support 3D prints**
- Step 4
  - The **wheels are attached to the 3D printed gears.** In the motor gears, shaft collars are installed to allow the gear to rotate with the axle
- Step 5
  - The **drive gears are installed onto the chassis through an axle.** The wheels are placed on screw joints, that allow them to freely spin around the screw
- Step 6
  - After each half of the chassis was built, c**ross braces were added** between the inner and outer chassis rails.

Steps 3, 4, and 5: Wheels and gears installed to inner and outer chassis supports



Steps 3, 4, and 5: Wheels and gears installed to inner and outer chassis supports

Step 6: Cross braces attached between the chassis halves



Completed Chassis Bottom View

Completed Chassis

Test Solution

To ensure that the chassis meets our established goals, we will conduct testing to ensure that the chassis meets our criteria of cross the field and barrier in under 3 seconds. **The specific procedure is outlined below:**

1. Position the chassis against the field perimeter, perpendicular to the center barrier.
2. Power the chassis forward at full speed.
3. Measure the time it takes for the drive to contact the opposite side of the field perimeter.
4. Repeat four more times.

Results

	Chassis Speed
Test 1	1.75
Test 2	1.83
Test 3	1.72
Test 4	1.77
Test 5	1.81
Average	1.78 seconds

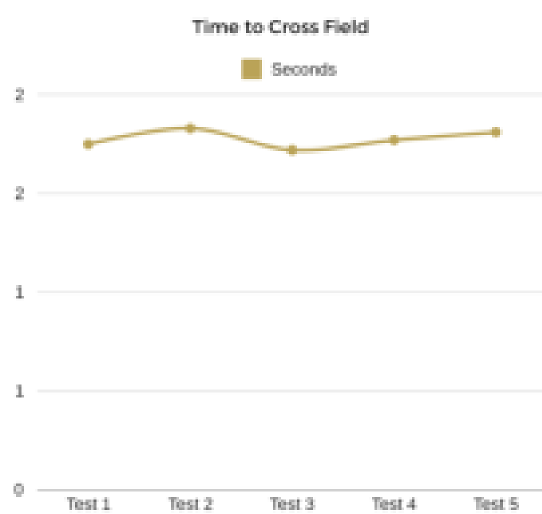


Figure 5: Testing graph

As the average time to cross the barrier is well under the benchmark of 3 seconds, **we can conclude that the chassis successfully meets our criteria.**

Design and Build: 24″ Intake Iteration 3

Date	@February 16, 2024
Category	24" Robot BLRS2
Authors	C Conner Siebert S Stephen Hohnholt
Design Cycle	Design Cycle 3

Goals and Redesign Considerations

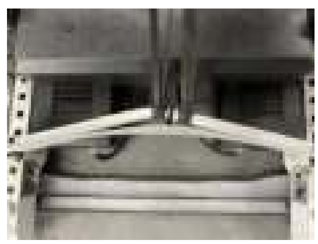
- After the 24″ chassis was completely rebuild to begin design cycle 3, the next step in preparing for the Purdue competition is to **install the existing intake onto the 24″ robot from the previous version.** The following criteria were considered and changed in the design of this intake:
    - To be able to de-score with this robot, **the intake must remain under 6″ tall while not holding a Triball**
    - While crossing the barrier, triballs often drop out of the bottom of the intake. Therefore, **this version should be able to hold the Triball while crossing the long barrier**

Build the Solution

- Step 1
  - The intake was **removed from the old chassis without being disassembled.** To do this, the screw joints on the support arms were removed and the intake was lifted off.
- Step 2
  - A 4-hole (2″) 2-wide c-channel is attached to the chassis cross brace using standoffs. **This will act as a new mounting point for the intake.**
- Step 3
  - **The existing intake is mounted to the c-channel using a screw joint.** This will allow the intake to pivot upwards while intaking a Triball while dropping back down to below 6″ tall when not holding a Triball.
- Step 4
  - To help hold the Triball while crossing the long barrier, a **polycarbonate sheet is added across the bottom of the intake** between the two drive supports.

Steps 2 and 3: Intake attached to new screw joint

Step 4: Polycarbonate shelf to hold triballs



Step 5: added high strength axle cross brace and Triball support

Step 6: Polycarbonate sleds added to the front of the intake

Side view of completed Intake

Top View of completed Intake

- Step 5
    - A high strength axle is attached in the rear of the intake, to **allow for rubber bands to be attached to pull the intake down**.
    - These rubber bands also place tension on the Triball once in the intake, to help **better hold the Triball in the robot.**
- Step 6
    - To allow the intake to pop up while driving into the alliance goal while scoring triballs, two polycarbonate sleds were added. These pieces extend out in-front of the intake, to **protect the flex wheel and axle of the intake.**

Triball in the completed intake

With the intake completed, we can measure that it is **approximately 5.8 inches off of the ground at its highest point, fulfilling the first of our end criteria.** We additionally conducted testing to examine Triball security when crossing the barrier.

Test Solution

To ensure that the intake meets our last criteria, we will test how secure Triballs are in the intake. **The specific testing procedure is outlined below:**

1. Place a Triball within the intake of the robot.
2. Position the robot approximately one foot away from, and facing, the center barrier.
3. Drive forward until the robot has driven over the barrier.
4. Record whether or not the Triball stayed securely within the intake.
5. Repeat the process four additional times.

Results

	Security (Y/N)
Test 1	Y
Test 2	Y
Test 3	Y
Test 4	Y
Test 5	Y
Average	Y (5/5)

Figure 1: Intake Testing

As the average pull force of the chassis is less than our maximum intake time of 0.5 seconds, **we can conclude that the intake successfully meets our criteria.**

PURDUE SIGBOTS

BLRS2 SKILLS ROBOTS

Task and Timeline Update: September

Date	@September 4, 2023
Category	BLRS2 Skills
Authors	Ⓒ Conner Siebert Ⓜ Max
Design Cycle	Design Cycle 1

The BLRS2 Skills team plans to take September as a month of planning for the season alongside onboarding new members onto the team.

All team members should be involved with the brainstorming and planning phase of the skills robots

- Some of the most important points in the season will come from our **initial analysis and strategy for the robot skills challenge**
    - To include all team members, our team will use the **first week of the season to meet and develop game play strategies**
- After a strategy has been developed, our team will make decisions regarding the design of the chassis that will serve as the base of the robot
    - While on the surface a chassis seems simple, design decisions made here will influence the shape of the rest of the robot
- Brainstorming potential ways to score points in a robot skills challenge is critical to success in the season
    - Discussion of the two methods of scoring in Over Under, scoring triballs and hanging on the elevation bar, will be the **focus of the 2nd half of the month of September.**

Important Date and Deadlines

- **9/10/2023:** Skills Challenge strategy should be decided by this date
  - It is important that a strategy is developed first, as the decisions made and priorities established will heavily influence the development of the subsystems of the robot.
- **9/10/2023-9/30/23:** This time is defined as brainstorming and planning varying subsystems of the robot
  - This is to allow for all team members to collaborate, contribute, and document the brainstorming process
- **9/30/23:** Final brainstorming deadline
  - This deadline is set to allow for October to be a month design work on the skills robots

Figure 1: Skills Team September Timeline

Planning: Robot Skills Challenge Strategies

Date	@September 7, 2023
Category	BLRS2 Skills
Authors	Conner Siebert Nathan Smith

Define the Problem

To begin the development of the skills robots, it is necessary to determine our strategy for the 60 second driver and programming challenges. To meet our goals, the robots should be able to:

- Launch all 44 match loads to the alliance side of the field, and into a position to be scored into the alliance goal
- Move the Triballs beginning on the field into the red alliance zone
- Score a large portion of the 58 Triballs into an alliance goal
- Hang both robots onto a "H" or higher elevation

Potential Strategies

Strategy 1:

- Catcher and Shooter
    - This strategy would optimize moving each Triball from the match load zone to the alliance goal by having **one robot shoot a Triball directly from the match load zone to the alliance goal**
    - While this solution would optimize the launching of each Triball, it would **push the limit of the 60 second timeframe in a skills run,** as one robot would be required to shoot a total of 44 match loads between the two zones
    - A **high level of driving and programming accuracy** would be required to ensure the two robots are lined up while launching Triballs

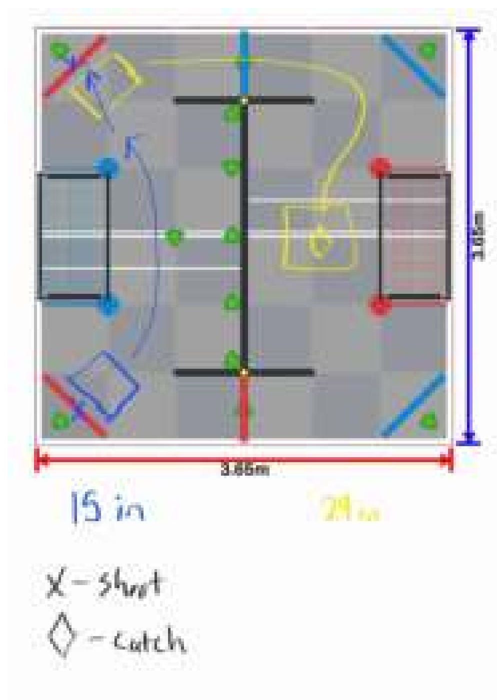


Figure 1: Proposed Strategy 1: Shooter and Catcher

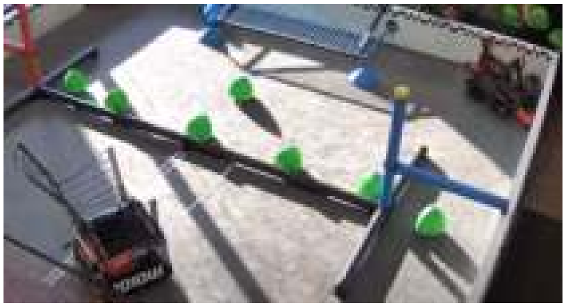


Figure 2: Vex U Team TNTN's shooter and catcher robots

Strategy 2:

- Launchers and Pushers
  - This idea would utilize both robots for the same tasks, with the robots **each occupying one match load zone and launching 22 match loads**
  - After launching all 44 Triballs, these robots would travel to the alliance side of the field, deploy wings, and **push the Triballs into the alliance goal**
  - This strategy **risks missing Triballs that are launched while pushing them into the goal.** This means there will be inconsistencies with this strategy

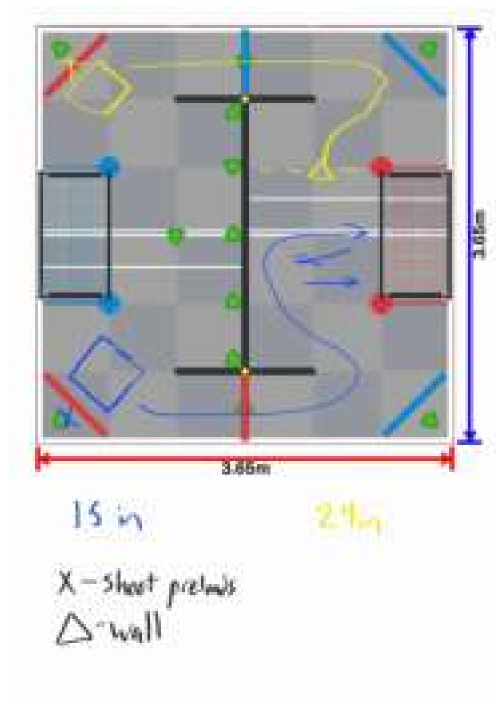


Figure 3: Proposed Strategy 2: Launchers and Pushers

Strategy 3:

- One at a Time
  - To ensure all Triballs are scored into the alliance goal, these robots would **intake a Triball from the match load zone, then transfer it across the field** into the alliance goal
  - Each robot would be **greatly limited by the speed** at which they can traverse the field to score only one Triball at a time
  - These robots **would not require a launcher mechanism**, as only an intake is needed to load Triballs from the match load zone and outtake them into the alliance goal

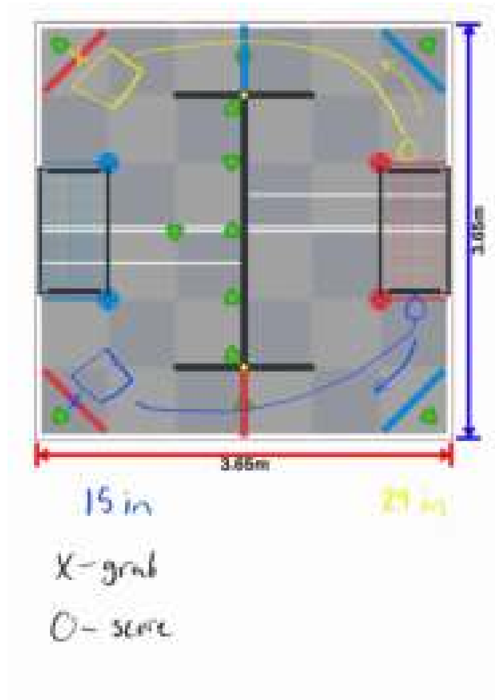


Figure 4: Proposed Strategy 3: One at a Time

Choosing The Best Solution

To compare each of the three skills strategies, our team created a list of pros and cons for each option and then created a decision matrix

Pros

Strategy 1	Strategy 2	Strategy 3
Optimizes match loads into the goal	Fastest match loading times	Scores all triballs directly into alliance goal
Requires minimal driving around the field	Quickly does all 44 match loads	Identical robots, making design simpler
Fast Cycle time from match loader	Identical robots, making design simpler	
	Will be able to score Triballs that start on the blue size while pushing Triballs into the alliance goal	

Cons

Strategy 1	Strategy 2	Strategy 3
Slower at match loading as the shooter must move between the two goals	Risk missing Triballs while pushing into alliance goal	Robot design will need to be light and fast to quickly move triballs
Risk leaving other Triballs on the field if focused on match loading	High traction drive is required to push Triballs into the goal	High complexity while programming
Requires the development of two different robots		Requires how driver skills to optimize route
High complexity while programming		

Decision Matrix

The following criteria were considered while we decided upon to determine the optimal skills plan

- Speed for Loading Match Loads (x/5)
    - With a majority of the Triballs in a robot skills challenge coming from the match load zones, **minimizing the amount of time a robot spends here is key to a successful run**
    - A higher score indicates faster speed
- Accuracy in scoring Triballs into the alliance goal (x/10)
    - With Triballs gaining three points (2 to 5) while being scored into the alliance goal, being accurate is important
    - With scoring each Triball into the goal, **we maximize the increase in our score from our efforts**
    - A higher score indicates more accurate control
- Number of Robots to Design (x/5)
    - With these robots being developed on a tight timeline, **if only 1 robot is needed to be designed our team will be able to optimize resources better**
    - A 5 indicates clone robots, a 1 indicates different robots
- Mechanical Complexity (x/5)
    - The fewer subsystems required on a robot, the better. This is because **additional features on a robot will lead to larger possibilities of mechanical or software failure** in a skills run, as well as add additional time to the development of the robot
    - A higher score indicates a simpler robot
- Time required to score Triballs after removed from the match loading zone (x/5)
    - After intaking a Triball from the match load zone, the **time required to get the ball into the alliance goal should be minimized** to quickly move onto the next Triball
    - If the Triballs are not in a scored position after it’s left the robot, extra would be added to to then move the Triball into the goal
    - A higher score equals a faster speed

	Speed for Loading Match Loads (x/5)	Accuracy in scoring Triballs into a the alliance goal (x/10)	Number of Robots to Design (x/5)	Mechanical Complexity (x/5)	Time required to score triballs after removed from the match loading zone (x/5)	Total (x/30)
Strategy 1	3	10	1	3	5	22
Strategy 2	5	2	5	4	2	18
Strategy 3	1	5	5	5	1	17

After considering each of the three solutions, our team has decided to move forward with solution one, Launcher and Catcher. This is because of the following considerations:

- This option has the highest potential accuracy in scoring Triballs
    - With Triballs going straight from the match load zone to the goal, there is **no chance for Triballs to bounce while landing**
    - In programming skills, there is no random chance for Triballs to be missed while pushing into the goal
- This option has the fastest scoring cycle after a Triball is removed from the match load zone
    - As the Triball will go directly from the launcher into the catcher robot, scoring it into the goal, there is **no slow downs in robot possession of a Triball**
- While this option will involve developing two different robots, **we believe the increased mechanical complexity is worth the additional effort** to maximum our skills routes and scores

Implementation Plan

As the skills challenge strategy has been developed, the following subsystems will be discussed and decided upon with the team for the remainder of September:

- Chassis Type
- Triball launcher mechanism
- Elevation mechanism

This skills challenge strategy will be an important part of the design process, as each subsystem will developed to optimize this strategy. **More precise timelines and planning will be discussed in the next entry.**

Design Brief: 15″ Iteration 1

Date	@September 11, 2023
Category	15" Robot BLRS2 Skills
Authors	Ⓒ Conner Siebert
Design Cycle	Design Cycle 1

BLRS2 Skills Robot: Iteration 1

Problem Statement

In a 60 second skills run, **this robot should be able to successfully launch all 44 match load Triballs from the match load zone into the 24″ robot and to hang in the "H" zone.** These robots are being designed to only compete in the robot skills challenge, as two other robots are being developed to compete in the head to head matches.

Constraints

- Begin in a 15×15×15 box
- Be able to intake Triballs from the match load zone
- Launch Triballs into the 24″ robot
- Achieve an elevation of "H" or higher on the hang bar

Criteria

- Be lightweight, with a goal of **10 lbs or less**
    - This will allow for quick acceleration and easier hanging
- Be able to launch all 44 match load Triballs, **22 from each zone, within 45 seconds**
    - This allows for for ample time to elevate in the endgame period
- Be able to **elevate within 15 seconds**
    - Combined with launching Triballs, this will maximize the score we can achieve

Timeline For Iteration 1

- September
  - **September will be a time for brainstorming and planning** for the skills robot, as we believe in starting the design with a clear vision and goals
  - With new members joining the team, it is also critical to the success of the plan to wait and include them in the brainstorming phase
- October
  - After completing the brainstorming and planning, **October will be a month of design work**, and also to address any issues with the results from the brainstorming phase
  - Nearing the end of the month, our design should be approaching a finalized state and be ready to begin building
- November
  - Concluding 2 months of brainstorming, planning, and designing, **November will be a month to build, drive, and program the robot** in preparation for our first opportunity to complete a robot skills challenge in December

Figure 1: BLRS2 Skills Team Timeline for Iteration 1

Design Brief: 24″ Iteration 1

Date	@September 14, 2023
Category	24" Robot BLRS2 Skills
Authors	Ⓒ Conner Siebert
Design Cycle	Design Cycle 1

BLRS2 Skills Robot: Iteration 1

Problem Statement

In a 60 second skills run, our robots should be able to efficiently score all 58 triballs into the goal and to hang in the “H” zone. These robots are being designed to only compete in the robot skills challenge, as two other robots are being developed to compete in the head to head matches.

Constraints

- Begin in a 24″x24″x24″ box
- Complement fielding of the 15″ robot
- Achieve an elevation of “h” or higher on the hang bar

Criteria

- Be light weight with a **goal of 10 lbs or less**
  - This will allow for quick acceleration and easier hanging
- Catch and place all balls shot by the 15″ robot into the goal
  - With the 15″ shooting all 44 match loads, this robot will need to (with the same/better consistency) **place all of those Triballs into the goal to be scored**
- Complement all needs of the 15″ robot on the field
  - This robot’s primary objective is to score the Triballs shot by the 15″ robot from match loads, but **will also need to assist with the scoring of any other Triballs in an optimal route.**

Timeline For Iteration 1

- September
  - September **plans to be a time for brainstorming and plan for the skills robot**, as we believe in starting the design with a clear vision and goals for the skills robot
  - With new freshman joining the team for the season, it is also critical to the success of the plan to include them in the brainstorming phase
- October
  - After completing the brainstorming and planning, **October aims to a month of design work**, and also flush any issues with the results from the brainstorming phase
  - Nearing the end of the month, our design should be approaching a finalized state and be ready to begin building
- November
  - Concluding 2 months of brainstorming, planning, and designing, **November will be a month to build, drive, and program the robot** in preparation for our first opportunity to complete a robot skills challenge on December

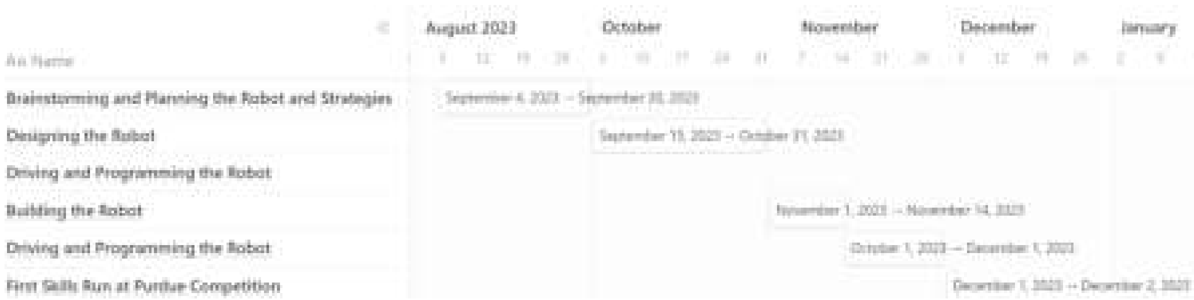


Figure 1: BLRS2 Skills Team Timeline for Iteration 1

Brainstorming and Decision Matrix: Chassis

Date	@September 17, 2023
Category	15" Robot 24" Robot BLRS2 Skills
Authors	N Noah Domogalik J Jacob Zawacki
Design Cycle	Design Cycle 1

Identify the Problem

The first and most important design consideration of the skills robot will be figuring out how to design a chassis which will maximize our skill score. **In order to do so, the robot will need to meet the following criteria:**

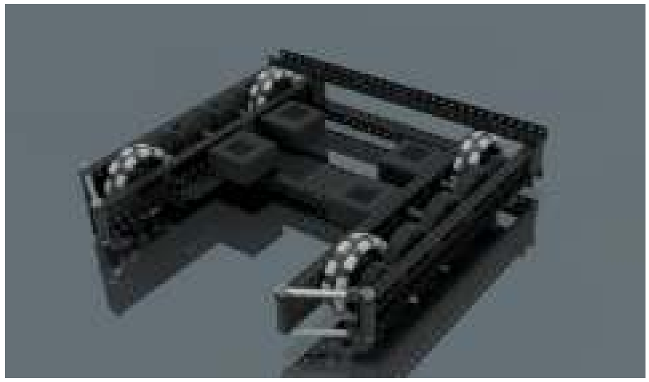
- Create a chassis design which can work in conjunction with our Triball Launcher and Endgame Hanging Mechanism.
- Make the chassis design easy to fix and require little maintenance.
- Creating an effective drive train and wheels which will work well with a 300 rpm gear ratio.
- Be able to move quickly around the field

This entry applies for both the 15″ and 24″ robots.

Brainstorm Solutions

Solution 1:

- 4 Wheel Omni-Drive @ 300 RPM
  - An Omni-Drive would use omnidirectional wheels, (omni-wheels) which are wheels that have rollers which will make the robot have a **fast turn speed.**
  - A 4 Wheel Omni-Drive would consist of the four omnidirectional wheels configured in a **tank orientation for quick forward, backward, and turn movements.**
  - The Omni-Drive could utilize six 11W motors, which would allow this drive type to **move with more power than others.**
  - This gear ratio would be set at 300 RPM.



Tank Drive Example - Team 99999V, BLRS Wiki

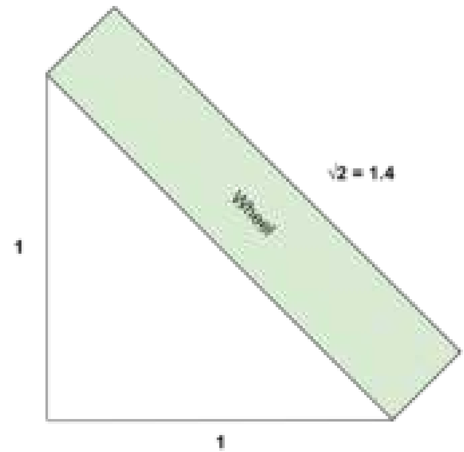
Solution 2:

- 4 Wheel Mecanum Drive @ 300 RPM
  - A Mecanum drive would use Mecanum wheels, which are wheels that allow better traction to the ground, along with the **ability to move horizontally in strafing movements.**
    - The larger surface area of the wheels, alongside the large rubber links, account for the increased traction.
  - The Mecanum drive could utilize either four or eight 11W motors, this design would **move at a faster pace, but would still be slower due to the enhanced traction.**
  - The gear ratio would be set to 300 RPM.

Mecanum Drive Example - CS-STEM

Solution 3:

- 4 Wheel X-Drive @ 300 RPM
  - An X-Drive places the four wheels on opposite corners of the drive, angled at a 45 degree angle. **This allows for side-to-side movement in addition to the four cardinal directions.**
  - With the four wheels set at a 45 degree angle, this enables the drive to move at a speed **1.4 times that of tank drives at the same RPM.** This can be proven using the lengths of a right triangle, as shown below:



- As the speed increases and power supplied to the drive remains constant, we can conclude that the **torque of an X-Drive is lower than that of a similar 300RPM tank drive.**
- The gear ratio would be set to 300 RPM, which would equate to 420 RPM using the math above.

X-Drive Example - BLRS, BLRS Wiki

Prototype + Test Solutions

To help make our decision on which drive type to use, we will be using rough prototypes and previous builds of each mechanism to record data. **In each, we will be recording the speed, maneuverability, and ability to cross the center barrier.**

- Speed: How fast the drive can cross the field
- Maneuverability: How easily the drive can maneuver around a set of obstacles
  - The same course will be used for each prototype

- Ability to Cross the Center Barrier: Time to cross the center barrier.

Testing Procedure

As the **testing procedure is the exact same for each drive type**, please refer to these steps for each three sets of testing data.

1. Place the chassis prototype against the field perimeter.
2. Drive the chassis forward until it contacts the other side of the field perimeter (12 feet). Record the time from start to stop (speed).
3. Align Triballs in the above position on the field. Record the time it takes to navigate in an s-curve through the obstacles (maneuverability).
4. Place the front end of the chassis contacting the center barrier.
5. Record the time it takes to cross the center barrier from the starting position.

4 Wheel Omni Drive @ 300 rpm Testing

Results

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
Speed	3.3s	3.2s	3.3s	3.3s	3.2s	3.26s
Maneuverability	6.8s	7.2s	6.6s	6.7s	6.8s	6.82s
Crossing the Barrier	0.7s	0.7s	0.7s	0.7s	0.7s	0.7s

Interpreting the Data

Speed:

- The drive averaged a time of 3.26 seconds to cross the field.

Maneuverability:

- The drive took an average of 6.82 seconds to complete the obstacle course.
- While the chassis was able to complete it, **there was definitely time loss with inefficient movements.**

Ability to Cross:

- The chassis had **absolutely no problems crossing the barrier,** consistently crossing in 0.7 seconds with little effort.

4 Wheel Mecanum Drive @ 300 rpm Testing

Results

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
Speed	4.8s	4.7s	4.7s	4.9s	4.8s	4.78s
Maneuverability	5.2s	5.2s	5.3s	5.2s	5.6s	5.3s
Crossing the Barrier	1.2s	1.3s	1.2s	1.2s	1.2s	1.22s

Interpreting the Data

Speed:

- The drive took an average of 4.78 seconds to cross the field
    - Notably slower than the tank chassis

Maneuverability:

- The drive was able to complete the obstacle course in an average of 5.3 seconds.
- While this did improve on the time set by the tank drive, **there was time loss in slower movements.**

Ability to Cross:

- While the drive was able to cross in an average of 1.22 seconds, **it took noticeably more effort than the tank drive.**

4 Wheel X-Drive @ 300 rpm Testing

Results

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
Speed	2.9s	2.9s	2.8s	2.8s	2.8s	2.84s
Maneuverability	4.5s	4.4s	4.7s	4.4s	4.3s	4.5s
Crossing the Barrier	n/a	n/a	n/a	n/a	n/a	n/a

Interpreting the Data

Speed:

- The drive took an average of 2.84 seconds to cross the field, the fastest drive to be tested

Maneuverability:

- **The drive had a good balance of speed and maneuverability**, completing the course in an average of 4.5 seconds.

Ability to Cross:

- Unlike the other two chassis prototypes, **the x-drive was unable to cross the barrier at all**, as the wheels did not gain enough purchase to pull itself over the center barrier.

Select the Best Solution

To compare each of the three chassis designs, our team created a list of pros and cons for each option then created a decision matrix

Pros

4 Wheel Omni Drive @ 300 rpm	4 Wheel Mecanum Drive @ 300 rpm	4 Wheel X-Drive @ 300 rpm
Simple Design	Utilizes strafe movements for higher maneuverability	Higher equivalent speed on a 300 RPM gear ratio
Highest power (speed/torque)	Simple Design	Utilizes strafe movements for higher maneuverability
More flexibility with motors (4, 6, 8, or 10 possible)		

Cons

4 Wheel Omni Drive @ 300 rpm	4 Wheel Mecanum Drive @ 300 rpm	4 Wheel X-Drive @ 300 rpm
Lack of maneuverability with movement only in the front or back directions	Slow moving drive in a speed-focused challenge	Lacks substantial pushing power
Difficult to program due to possible drifting from the omni-wheels	Largest wheels, could take up space from other mechanisms on the robot	Could only use 4 or 8 motors
	Could only use 4 or 8 motors	Difficult to move over the middle bar

Decision Matrix

The following criteria were considered while we decided upon the chassis design:

- Compatibility with Other Subsystems (x/5)
    - The chassis design needs **leave space for and easily compliment the scoring mechanisms** on our robot.
    - A more compatible chassis will have a higher score.
- Ability to Effectively Program (x/10)
    - As Programming Skills is a large portion of the Robot Skills Challenge, a chassis we can use effectively in programming is essential.
    - Ideally, **we would be able to use a chassis for complex movements and consistent paths.**
    - A more easily programmable chassis will have a higher score.
- Maneuverability (x/10)
    - An important aspect to consider is how effectively the drive can maneuver around the field. **The drive should be able to cross under the poles and around the goal easily.**
    - The matrix score for this metric is taken directly from previous test data.
    - A more maneuverable chassis will have a higher score.

- Speed (x/10)
  - Due to having a 60 second time limit, it is required for the chassis to move quickly around the field and **score all Triballs within the time limit.**
  - The matrix score for this metric is taken directly from previous test data.
  - A faster chassis will have a higher score.
- Able to move over middle bar (x/5)
  - Maneuverability over the middle bar will give a greater advantage during designing a skills route.
  - The robot should utilize a design that is able to **move over the middle bar as quick as possible.**
  - The matrix score for this metric is taken directly from previous test data.
  - A chassis that can more effectively cross the bar will have a higher score.

	Compatibility with Other Subsystems (x/5)	Ability to Effectively Program (x/10)	Maneuverability (x/10)	Speed (x/10)	Able to Move Over Middle Bar (x/5)	Total (x/40)
4 Wheel Omni Drive @ 300 rpm	5	6	6	8	5	30
4 Wheel Mecanum Drive @ 300 rpm	3	8	7	6	3	27
4 Wheel X-Drive @ 300 rpm	**5**	**10**	**9**	**9**	**1**	**34**

After considering each of the three solutions, our team has decided to move forward with a 4 Wheel X-Drive. This is because of the following considerations:

- Fastest option
  - As the power is distributed with a 1.4x increase to speed, this will be essential in making our robot complete the skills route quickly and effectively.
    - As there are no opposing robots on the skills field, the **lack of pushing power should not be an issue.**
- Large Degree of Maneuverability
  - With the high speed and holonomic movement, the chassis should be very easy to maneuver around the field obstacles.
    - We believe that the **high speed and maneuverability make up for the lack of ability to cross the middle barrier.**
  - This speed and maneuverability should make the drive more effective in the Programming Skills challenge as well.

Implementation Plan

- Our team will utilize **CAD software to design and plan** out how the chassis will be built, which will be recorded in the following entries.
- As a general idea, the 15″ chassis will need to be considerably smaller than the 24″ robot, and should not focus on catching launched Triballs as a result.
  - As such, we will need to design the chassis in a way to support mechanisms for quick match loading and launching of Triballs
  - The chassis will also need to be fast enough to cross over to the scoring side of the field after launching Triballs.
- For the 24″ robot, this robot will largely focus on catching and scoring the Triballs launched by the 15″ robot. As such, a wider, larger chassis will be necessary to support larger mechanisms and catch Triballs.
- **Referring to the Gantt Chart, we will plan to work on the CAD until our planned completion date of October 31st, where we will then begin building the robot soon after.**

Brainstorming and Decision Matrix: Endgame Hanging Mechanism

Date	@September 19, 2023
Category	15" Robot 24" Robot BLRS2 Skills
Authors	C Conner Siebert J Jacob Zawacki A Alex Lam
Design Cycle	Design Cycle 1

Identify the Problem

A major design consideration of the skills robots will be the endgame hanging mechanism. **In order to maximize our skills score, the robots will need to meet the following criteria:**

- Hang in the "H" zone consistently
- Be a lightweight mechanism
- Quickly latch onto the hang bar and achieve an "H" zone hang or higher
- Be easily powered
- Not interfere with other subsystems of the robot

This entry applies for both the 15" and 24" robots.

Brainstorm Solutions

Solution 1:

- Continuous Lift
    - A continuous lift would use **motors to both raise the pole clamp to the top of the elevation bar and then pull the robot up on the lift**
    - A rope would be used to move the lift with the motors, as seen in the sketch of figure 1 below
    - Initial ideas for the powering the lift is to PTO motors from the chassis, as while lifting the robot will not need the ability to drive

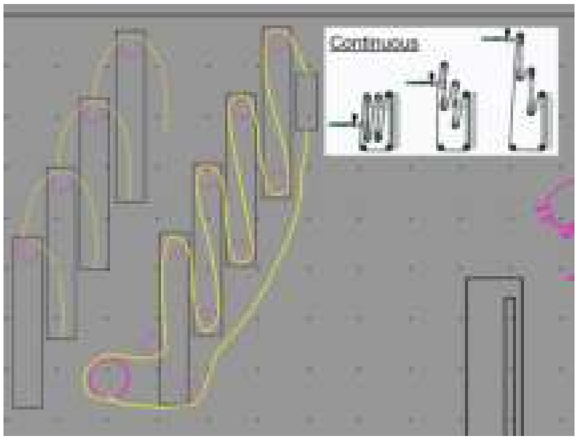


Figure 1: Sketch of Continuous Lift with Reference

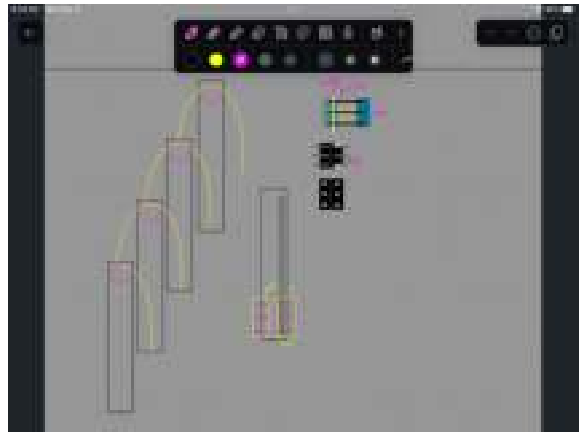


Figure 2: Sketch of Continuous Lift


Figure 3: Cascade Lift Design from 99904A's Tower Takeover Robot

Solution 2:

- Winch
  - This option would **utilize a cascade lift for the pole clamp to reach the top of the elevation bar, then use a rope on a winch to raise the robot to the "h" zone**
  - Like the continuous lift, the winch would be powered by motors
  - As the robot is raised by the winch, the cascade lift would be retracted, allowing for the robot to remain stable and perpendicular to the elevation bar

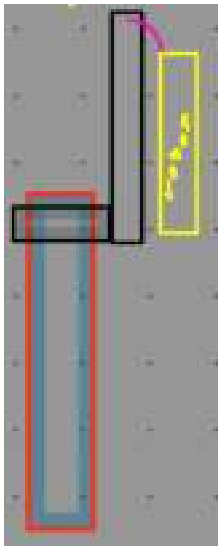


Figure 4: Initial Sketch of Robot Ending Position with a Winch Hanging Mechanism

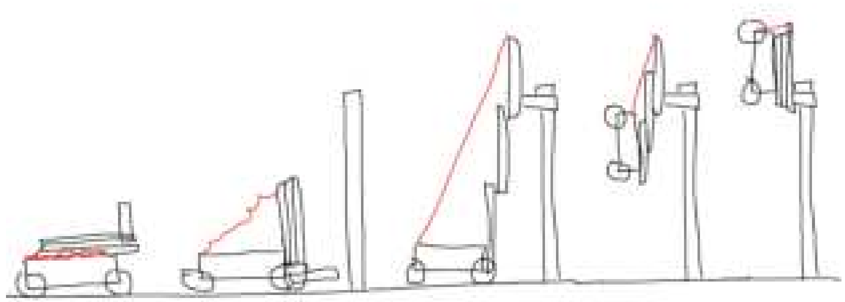


Figure 5: Proposed Sketch of a Solution 2: Winch

Solution 3:

- Double Reverse 4-Bar
  - A double reverse 4-bar is a classic Vex lift, used by thousands of teams in past seasons, as seen in figure 6 below by team 574C during the In The Zone season
  - **Compared to the other two lift options, would be powered by pneumatic pistons, reducing the number of motors and the complexity of the lift mechanism**
  - This lift would have a large footprint on the robot, as a wider design with a long length would be required. This could lead to concerns with implantation onto the robot
  - With the added pneumatics and large structure, the weight of the robot could be increased significantly compared to the other two options

Figure 6: 574C's Double Reverse 4-Bar

Figure 7: Prototyped DR4B Powered by a Pneumatic Piston

Prototype + Test Solutions

To help make our decision on which hanging mechanism to use, we will be using rough prototypes and previous builds of each mechanism to record data. **In each, we will be recording the speed, strength, and ability to hold weight.**

- Speed: How fast the mechanism is able to life a 10 pound weight.
- Strength: How much weight the mechanism is able to lift.
- Ability to Hold: Does the mechanism hold weight after a match.

Continuous Lift Testing

Procedure

1. Attach the Continuous Lift to an object weighing 10 pounds
2. Extend the Lift to its maximum, so that the weight is farthest away from the start of the lift
3. Power the lift at maximum velocity, record the time it takes to contract the lift fully (speed)
4. Turn off the program to eliminate power to the lift. Hold the lift vertical with the weight on the bottom, and record the time it takes to fully extend (ability to hold)
5. Add another 10 pounds and repeat the speed test, not recording the speed. Add more weight and repeat until the lift is no longer able to lift weight (strength).

Results

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
Speed	2.5s	2.4s	2.8s	2.7s	2.3s	2.54s
Strength	10lbs	20lbs	30lbs (fail)	30lbs (fail)	30lbs (fail)	30lbs maximum
Ability to Hold	3.4s	3.6s	3.8s	3.1s	3.2s	3.42s

Interpreting the Data

Speed:

- The lift averaged a time of 2.54 seconds to fully lift the 10lb weight.

Strength:

- In adding increments of 10lbs to the Lift, we found that it could no longer lift 30lbs of weight, meaning the reasonable maximum is 20lbs.
- We tested the 30lbs again multiple times after letting the motor rest to be sure heating was not an issue, and observed no change in the results.

Ability to Hold:

- Without a hold after power is cut, the Lift lowers at a speed averaging 3.42 seconds.
- As this mechanism does not have the ability to hold on its own after a match, **it would need a separate mechanism to prevent lowering.**

Winch Testing

Procedure

1. Attach the Winch Lift to an object weighing 10 pounds
2. Unwind the Winch to its maximum, so that the weight is farthest away from the start of the lift
3. Power the Winch at maximum velocity, record the time it takes to completely winch the chord (speed)
4. Turn off the program to eliminate power to the lift. Hold the lift vertical with the weight on the bottom, and record the time it takes to fully unwind (ability to hold)
5. Add another 10 pounds and repeat the speed test, not recording the speed. Add more weight and repeat until the lift is no longer able to lift weight (strength).

Results

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
Speed	1.6s	1.7s	1.6s	1.8s	2.0s	1.74s
Strength	10lbs	20lbs	30lbs	40lbs (fail)	40lbs (fail)	40lbs maximum
Ability to Hold	Indefinite	Indefinite	Indefinite	Indefinite	Indefinite	Indefinite

Interpreting the Data

Speed:

- The Winch averaged a time of 1.74 seconds to fully wind the 10lb weight.

Strength:

- In adding increments of 10lbs to the Winch, we found that it could no longer wind 40lbs of weight, meaning the reasonable maximum is 30lbs.
- We tested the 40lbs again multiple times after letting the motor rest to be sure heating was not an issue, and observed no change in the results.

Ability to Hold:

- Without a hold after power is cut, the Winch does not unwind, meaning that the Winch could hold the weight indefinitely.
- **While the test was conducted without a clamp mechanism to guarantee the Winch would not unwind, we would likely add one regardless for extra insurance.**

Double Reverse 4-Bar (DR4B) Testing

Procedure

1. Attach the DR4B Lift to an object weighing 10 pounds
2. Extend the Lift to its maximum, so that the weight is farthest away from the weight
3. Power the lift at maximum velocity, record the time it takes to contract the lift fully (speed)
4. Turn off the program to eliminate power to the lift. Hold the lift vertical with the weight on the bottom, and record the time it takes to fully extend (ability to hold)
5. Add another 10 pounds and repeat the speed test, not recording the speed. Add more weight and repeat until the lift is no longer able to lift weight (strength).

Results

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
Speed	2.1s	2.1s	2.2s	2.1s	2.4s	2.18s
Strength	10lbs	20lbs	30lbs	40lbs	50lbs (fail)	50lbs maximum
Ability to Hold	2.2s	2.8s	2.2s	2.6s	2.5s	2.46s

Interpreting the Data

Speed:

- The lift averaged a time of 2.18 seconds to fully lift the 10lb weight.

Strength:

- In adding increments of 10lbs to the Lift, we found that it could no longer lift 50lbs of weight, meaning the reasonable maximum is 40lbs.

Ability to Hold:

- Without a hold after power is cut, the Lift lowers at a speed averaging 2.46 seconds.
- As this mechanism does not have the ability to hold on its own after a match, **it would need a separate mechanism to prevent lowering.**

Select the Best Solution

To compare each of the three hanging mechanisms, our team created a list of pros and cons for each option then created a decision matrix

Pros

Continuous Lift	Winch	Double Reverse 4-Bar
A single mechanism	Once hang bar is grabbed, simple mechanism to raise the robot	Can be powered without motors
Small footprint and space requirements	Lift will provide stability while raising the robot	Stable lift mechansim
Can both deploy and retract on a single powering system (motors)	Fast raise of clamp	Quick to raise lift
Requires no additional pneumatics	Requires no additional pneumatics	

Cons

Continuous Lift	Winch	Double Reverse 4-Bar
Will require more motors or PTO from the drive	Two separate mechanisms	Will require additional pneumatic components
Locking mechanism will be needed for after power is lost to the robot	Will require more motors or PTO from the drive	Large footprint
Will require large amounts of torque from motors to raise the robot	Locking mechanism will be needed for after power is lost to the robot	Will require additional pneumatics, adding weight to the robot
		Will require large pistons to lift robot
		Will require a high PSI to lift robot (near 100 psi, limiting other pneumatic actuations)

Decision Matrix

The following criteria were considered while we decided upon a hanging mechanism

- Size and Footprint (x/10)
    - As the rest of the robot is being developed along side the hanging mechanism, **limiting the size of this mechanism is a limiting factor along the rest of the robot**
    - A higher score will indicate a more convenient footprint
    - As size is a key factor, it's weight is doubled to x/10
- Speed (x/10)
    - Within a 60 second skills run, **minimizing the time spent lifting the robot** will allow for more Triballs to be scored
    - This metric has been tested previously, and the matrix score will reflect test results
    - A higher score will indicate a faster lift
    - As speed is a key factor, it's weight is doubled to x/10

- Mechanical Complexity (x/5)
  - While designing a robot, a **less complex design would have less failure points**
  - A higher score will indicate a less complex design
- Weight Added (x/5)
  - As an overall design consideration of the skills robots is weight limiting, **the lift poses to be a large portion of the total weight**
  - A higher score means that the lift can handle more weight
  - This metric has been tested previously, and the matrix score will reflect test results
- Stability While Lifting and Hanging (x/5)
  - The final mechanism should be stable to **limit the risk of failure while lifting the robot**
  - A higher score indicates a more stable design

	Size and Footprint (x/10)	Speed (x/10)	Mechanical Complexity (x/5)	Weight Added (x/5)	Stability While Lifting and Hanging (x/5)	Total (x/30)
Continuous Lift	8	6	2	3	1	20
Winch	**6**	**8**	**4**	**4**	**4**	**26**
Double Reverse 4-Bar	2	7	5	1	5	20

After considering each of the three solutions, our team has decided to move forward with a winch lift. This is because of the following considerations:

- The winch has a small footprint
  - As the string spool will be the largest portion of the winch, **minimizing the footprint of the lift for the pole clamp** will be large consideration with the design moving forward
- The winch has the smallest added weight
  - Eliminating the double reverse four-bar due to it's added pneumatic components and heavy structure, **the winch has the opportunity to be designed small**
- Stability is key to a strong hang
  - The retraction of the lift for the pole clamp as the robot rises will be a **strong support for the robot**

Implementation Plan

- Before CAD begins on the hanging mechanism, our team plans to test possible winch speeds and diameter of the spool to create the fastest hang possible
  - We aim to complete thorough winch testing within the next **2 weeks**
- After the chassis design is completed, the hanging mechanism will be designed along side the catapult to ensure no interference on the robot between the subsystems

Brainstorming and Decision Matrix: 15″ Triball Intake and Launcher

Date	@September 25, 2023
Category	15" Robot BLRS2 Skills
Authors	C Conner Siebert J Jacob Zawacki
Design Cycle	Design Cycle 1

Define the Problem

In order to move Triballs from the match loading zone to the catcher 24″ robot, **our 15″ will need the ability to:**

- Remove Triballs from the match load zone
- Place the Triballs into a launching mechanism
- Fire Triballs from the stationary robot at the match load zone into the 24″ robot

Brainstorm Solutions

Solution 1:

- Roller intake into a catapult
    - This solution allows for Triballs to be **removed from the match load zone using rubber band rollers**, and placed into a waiting catapult basket
    - This solution requires 2 different subsystems that work in: an intake and a launcher
        - These two aspects would require their own powering method, and will have to be tuned to work together
    - The cycle time will be limited by multiple factors for this solution:
        - The robot driving to get a Triball into the intake
        - The time required for the catapult to return to it's lowered state
        - The speed that the Triball travels from the zone into the catapult

Figure 1: VRC Team 229V Early-Season Robot

Solution 2:

- Roller intake into a flywheel
  - Following the intake configuration as seen in solution 1, the catapult in this solution will be replaced with a flywheel.
  - This allows for a **reduction in cycle time, as the waiting period of the catapult being lowered will be removed**
  - The flywheel would allow for easy tuning of launching distance, as the speed for both the top and bottom flywheels can be adjusted

Figure 2: Prototype of Triball Flywheel to be combined with the intake from figure 1

Solution 3:

- Passive loading Catapult
  - This solution removes the requirement of utilizing a second subsystem of an intake to get a Triball into the launcher mechanism by **directly lowering the catapult basket onto the Triball located in the match load zone**
  - This catapult would follow the following cycle:
    - A Triball is placed by a drive team member into the match load zone
    - The catapult is lowered into the match load zone
    - The basket of the catapult goes over the Triball, passing over it

- The catapult fires, taking the Triball out of the zone with it

Figure 3: Passive Loading Catapult Prototype

Figure 4: Catapult prototype intaking triball from the match load zone

Prototype + Test Solutions

To assist in determining the best launcher type, our team built two prototypes as seen in figures 2, 3 and 4 above. **Two test were conducted, to determine the speed and accuracy of each possible solution to assist in determining the best solution.**

Test 1: Firing Speed

To find the highest rate of speed possible our team conducted a speed test on each launcher type. This was conducted using the following steps:

Testing Procedure

1. The launcher was placed at the match load zone, in the same location it would be positioned on the robot
2. A timer is started as the first Triball is placed into the match load zone or launcher
3. 22 Triballs were introduced into the launcher in the same manner that they would be in a match
    a. For the flywheel, as there was not a prototyped intake, it was assumed a wait time of 0.75 seconds would occur between Triballs being entered into the robot
    b. For the catapult, Triballs were placed in a legal match load position to be picked up by the catapult
4. The timer is stopped as the 22nd Triball is shoot

Test Data

	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Flywheel	16.75	16.67	16.49	16.56	16.73	16.64 seconds
Catapult	14.52	14.67	14.64	14.56	14.72	14.66 seconds

Test 2: Firing Accuracy

To determine how accurate each launch will be in getting Triballs into the 24″ robot's catcher, a simulation of a skills run was completed. **By firing 22 Triballs as was done in test 1, we were able to determine the expected accuracy of the shooter in a skills run**

Testing Procedure

1. The launcher was placed at the match load zone, in the same location it would be positioned on the robot
2. A Triballs is placed into the launcher, aimed towards the alliance goal where the 24″ robot would be positioned
3. As a Triball is launched, it is recorded if it lands within the 12″ diameter circle target placed in front of the alliance goal where the 24″ robot would be positioned
4. This process is repeated until 22 match loads have been fired, completing the test

Test Data

	Test 1 (x/22)	Test 2 (x/22)	Test 3 (x/22)	Test 4 (x/22)	Test 5 (x/22)	Average
Flywheel	18	17	18	16	17	17.2
Catapult	21	20	22	18	19	20

Testing Conclusions

From the two test conducted, **the catapult appears to be a better option.** This is because of the following factors:

- Higher firing Rate
    - From test 1, it is seen the catapult was nearly 2 seconds at firing the 22 match load Triballs
    - Throughout a skills run, **the catapult poses to save 4 seconds when able to be match loaded into directly** compared to a firing mechanism that requires an intake
- Better Accuracy
    - As seen in test 2 results, a catapult on average scores 3 more Triballs into the alliance goal compared to the flywheel
    - From both match load zones, this would be a total of **6 Triball difference from the alliance zone to the alliance goal, a total point differential of 18 points gained**

Select the Best Solution

Pros

Solution 1	Solution 2	Solution 3
Quickly intake into the robot	No rebound time between Triball fires	No chassis movements required
	Quickly intake into the robot	Combines the intake and launching mechanism
		High accuracy with each shoot
		Fastest firing rate between Triballs

Cons

Solution 1	Solution 2	Solution 3
Requires 2 subsystems: intake and catapult	Requires 2 subsystems: intake and flywheel	Requires precise match load placement
Chassis movements required between Triball intaking	Flywheel tuning can be difficult	High torque required to draw back catapult
Requires time between fires while catapult returns to a loading state	Chassis movements required between Triball intaking	Firing rate will be limited by the speed that Triballs can be introduced by human players
High torque required to draw back catapult	Firing rate will be limited by the speed that Triballs can pass through the robot	

Decision Matrix

The following criteria were considered while we decided upon the intake and launcher mechanism(s)

- Speed of intaking (x/5)
  - **The speed in which Triballs are introduced into the robot** are the first lost of time while match loading
  - The ideal intake would work faster than the speed in which we can place the Triballs into the match load zone
  - A higher score indicates a faster intake speed
- Speed of launching (x/5)
  - Once Triballs are moved into the robot, the **speed in which they leave for the 24″ robot will be important**
  - As there is a 1 Triball possession limit, we will not be able to
  - A higher score indicates a faster launch
- Overall Triball Cycle Time (x/10)
  - Overall cycle time of the match loading is the largest consideration of this launcher
  - As 45 seconds of our skills route is advocated to match loading, **optimizing this will be important to meet our 1 Triball a second goal**
  - A higher score indicates a higher cycle time
- Mechanical Complexity (x/5)
  - As these subsystems are designed and used on the 15″ robot, **we will consider both the difficulty of design and also the modes of failure**
  - A higher score indicates a simpler design
- Accuracy of Triball launching (x/5)
  - A secondary consideration is how accurate the launcher will be in getting the Triballs into the 24″ robot catcher
  - **Triballs that are not successfully caught by the 24″ will not be scored into the alliance goal,** leading to a 3 point loss per Triball
  - A higher score indicates a more accurate launch

	Speed of intaking (x/5)	Speed of launching (x/5)	Overall Triball Cycle Time (x/10)	Mechanical Complexity (x/5)	Accuracy of triball launching (x/10)	Total (x/35)
Solution 1	5	2	4	2	10	23
Solution 2	5	4	6	2	5	22
Solution 3	4	5	10	4	10	33

After considering each of the three solutions, our team has decided to move forward with a passive loaded catapult. This is because of the following considerations:

- This solution has the overall fastest cycle time
  - By combining the two subsystems into one motion, a large portion of time is removed from those solutions 1 and 2 from the motion of pulling the Triball from the match load zone
  - With a high gear ratio and torque catapult, **the firing rate will be limited by the speed we can place match loads into the zone**
- Catapults have high accuracy
  - After conducting testing with the flywheel and catapult, **we found a higher accuracy in the firing to be in catapult compared to the flywheel setup**
- One subsystem has less mechanical complexity
  - By only having a catapult, there will be **less possible points of failure in this design compared to a two subsystem design**

Implementation Plan

- To begin implementing the Triball catapult, **our team will begin to design the catapult in CAD** while developing the prototype as seen in figures 3 and 4 further to ensure the design remains viable.
- After design is completed in CAD, it will be built and tested on the 15″ Chassis to prepare for competition

Planning: 15″ Motor and Pneumatic Distribution

Date	@September 30, 2023
Category	15" Robot BLRS2 Skills
Authors	Conner Siebert
Design Cycle	Design Cycle 1

Define the Problem

While planning our first iteration of the skills robots, determining the distribution of motors and pneumatics will provide a base for the design criteria. **The potential solutions will revolve around powering three main subsystems: chassis, Triball catapult, hanging lift, and future subsystems including wings.**

There are three main items to power the robot are:

- 11 Watt Motor
    - The first motor of the V5 system, **11 W motors provides the base for most mechanisms in the Vex Robotics Competition**
    - 11 W motors have three possible outputs: 100 rpm, 200 rpm, and 600 rpm
    - After 8 motors have been connected to the V5 brain, the current of 2.5 amps begins to be limited with each additional motor. **This is often seen as a diminishing marginal return with each additional motor over 8**
- 5.5 Watt Motor
    - New this season, the 5.5 W motor provides a smaller footprint compared to the 11 W motor but only half other power
    - The 5.5 W motor can only output at 200 rpm
    - A 5.5 W motor limits the current in the same way as a 11 W does
- Pneumatic Pistons
    - These linear motion actuators provides a motion outside of the rotational outputs of motors
    - **Additional hardware is required to add pneumatics to a robot, with an air reservoir and solenoids are required, taking space and adding weight to the robot**
    - The use of pneumatics has no affect on the performance of motors
    - To minimize the additional weight, complexity, and space taken by pneumatics, we have decided at the current time to avoid using them on this robot

Potential Solutions

Solution 1: Twelve 11W Motors, One 5.5W Motor

- 11W Motors: 8 Motor Chassis, 2 Motor Catapult, 2 Motor Winch
- 5.5W Motors: 1 Motor Elevation Bar Clamp. **Current per motor = 2.05 A**
- **This solution focuses minimizing the use of pneumatics on the robots**
- The use of only 2 motors on the winch will limit the hanging speed
- Total Weight = (12 11W motors) * (0.342 lb/motor) + (1 5.5 W motor) * (0.2 lb/motor) = 4.304 lbs

Solution 2: Ten 11W Motors, One 5.5W Motor

- 11W Motors: 4 Motor Chassis, 2 Motor Catapult, 4 Motor Winch
- 5.5W Motors: 1 Motor Elevation Bar Clamp
- To decrease weight and mechanical complexity, the chassis is reduced from 8 motors to 4 motors
- **To increase lifting speed, the # of motors on the winch is doubled compared to solution 1**
- Total Motor Weight = (10 11W motors) * (0.342 lb/motor) + (1 5.5 W motor) * (0.2 lb/motor) = 3.62 lbs
- **Current per motor = 2.21 A**

Solution 3: Eight 11W Motors, One 5.5W Motor

- 11W Motors: 4 Motor Chassis with 4 motors ratcheted to both catapult and winch
- 5.5W Motors: 1 Motor Elevation Bar Clamp
- This solution minimizes any current limiting by only having 9 motors on the robot
- **The winch is still able to operate at a high speed, as 4 motors will be operating it**
- **Catapult firing rate will be faster by the use 4 motors compared to 2 previous solutions**
- Total Motor Weight = (8 11W motors) * (0.342 lb/motor) + (1 5.5 W motor) * (0.2 lb/motor) = 2.936 lbs
- **Current per motor = 2.39 A**

Choosing The Best Solution

To compare each of the three motor and pneumatic combinations, our team created a list of pros and cons for each option then created a decision matrix

Pros

Solution 1	Solution 2	Solution 3
Lowest mechanical complexity	High speed winch	Lightest Option (by 1.217 lbs)
	Low mechanical complexity	Lowest current limiting (0.11 A lost per motor)
		High speed winch
		Highest torque for catapult

Cons

Solution 1	Solution 2	Solution 3
High levels of current limiting (0.45 A lost per motor)	Medium current limiting (0.29 A lost per motor)	High mechanical complexity
Heavist Option	Heaver Option	
Low winch speed	Large lost of space in the robot to motors	

Decision Matrix

The following criteria were considered while we decided upon the motor and pneumatic distribution

- Weight (x/10)
  - As weight is a large considering in each stage of the design for the skills robots, **minimizing the impact of additional components is important**
    - For this reason, the weight doubled in the decision matrix considerations
  - A higher score indicates a more weight-efficient design
- Mechanical Complexity (x/10)
  - As the **designability, constructability, and maintainability of the robot is important throughout the design process**, the mechanical complexity of each option will be analyzed
    - The mechanical complexity was also given double the point values in the decision matrix due to it's importance
  - A higher score indicates a less mechanically complex design
- Winch Speed (x/5)
  - As each second of a skills run is optimized, the **speed in which we will be able to climb will be considered**
  - A higher score indicates a faster speed
- Catapult Torque (x/5)
  - In the same consideration as the winch, **maximizing the speed and torque of the catapult is also important**
  - A higher score indicates more torque
- Current Limiting (x/5)
  - As each motor is placed on the robot, the **efficiently of the motor should be maximized to justify it's addition to the robot**
  - A higher score indicates a more power-efficient system

	Weight (x/10)	Mechanical Complexity (x/10)	Winch Speed and Torque (x/5)	Catapult Speed and Torque (x/5)	Current Limiting (x/5)	Total (x/35)
Solution 1	2	3	2	2	1	10
Solution 2	6	8	5	2	3	24
Solution 3	10	5	5	5	5	30

After considering all three options, solution 3 was the clear best choice to move forward with. This is because of the following factors:

- Less Motors = Less Weight
  - As this option had 4 and 2 motors less than solutions 1 and 2 respectively, **the over one pound difference in weight is over 10% of our target weight**
- Winch and Catapult are maximized
  - With four motors from the chassis powering both of these subsystems, they will be able to operate with high torque and speed
- **All motors are operating at full potential**
  - With only nine motors total, there is minimal current limiting occurring on the robot while the other two solutions saw greater losses

Implementation Plan

As each design stage of the robot occurs in the coming months, these decisions will be carried out. It is our goal to stick to this configuration throughout the first iteration of the skills robots.

A more thorough timeline update will be included in our next entry.

Sources

Motor Weights and Current Limiting

https://wiki.purduesigbots.com/vex-electronics/vex-electronics/motors

Pneumatic Piston Weight

https://www.smcpneumatics.com/NCJ2D10-200.html

Task and Timeline Update: October

Date	@October 1, 2023
Category	15" Robot 24" Robot BLRS2 Skills
Authors	Ⓒ Conner Siebert Ⓜ Max
Design Cycle	Design Cycle 1

September Recap

September proved to be a successful month in accomplishing the following:

- Skills challenge strategy development
  - Our team was able to successfully develop our overall strategy early in the month thanks to our team general meetings
- Complete the brainstorming and planning phases of our design
  - After the completion of our match strategy, **our team broke the robot down into three main components: chassis, triball launcher, and hanging mechanism**
  - This allowed for each of the three subsystems to be thought out and decided upon to move forward
- New team member onboarding
  - A team goal outside of the robot development for the month of September was to create a team environment that allowed new members to successfully contribute to the team from the very beginning
  - **Through our new team structure of weekly general meetings, new members were quick to learn about our club and have their inputs on the design and implementation considered**

October Task and Timeline

Following the conclusion of the October 1st team general meeting, we developed a timeline for the design and implantation of the skills robots with the following considerations:

- The robot should be mostly to completely designed with CAD before machining and building begins
    - **This will allow us to check that all the subsystems work together before beginning to build**, as building without a full CAD often leads to troubles with interactions between subsystems
- The initial CAD should be completed by October 31st
    - To allow time for machining, building, driver practice, and programming that all will need to occur in November
- A large emphases is still being placed upon the skills robots being able to compete on December 2nd
    - This goal is to **post a robot skills challenge score before the December 31st World Qualification Spots are allotted** biased off of the top 5 globally ranks skills score

VEX U World Championship Qualifying Criteria

- On December 31st, 2023, the top 5 unqualified VEX U teams based on their World Skills scores will receive an invitation to the 2024 World Championship.

Figure 1: Insert from the Vex World Qualifying Criteria

Important Date and Deadlines

- **10/10/2023:** Chassis CAD completed
  - Initial CAD check-in point to make sure things are on task and to look for future issues
- **10/31/2023:** Complete Robot CAD is completed and building will begin
  - This will allow for time for any issues that arise during building to flushed out in CAD
- **11/12/2023:** Robot #1 is completely built and ready for driver practice and programming
  - The completion of robot #1 will prove our design is function and ready to compete
- **11/19/2023:** Robot #2 is completely built and ready for driver practice and programming
  - After robot #2 is complete, our team will be a good position to prepare for the December 2nd skills runs
- **12/2/2023:** Purdue VRC Blended Comp
  - Ready to post a skills score!

Figure 2: BLRS2 October Updated Proposed Timeline for the Skills Robot Development

Prototyping and Testing: Winch Hang Mechanism

Date	@October 2, 2023
Category	BLRS2 Skills
Authors	Ⓐ Alex Lam Ⓓ Dominic H
Design Cycle	Design Cycle 1

Objective

To maximize the skills score obtainable for the team, the designed robot must be able to “hang” or climb past the “H” level tier. **Testing was done to determine whether the designed winch method is a viable solution.**

Procedure

Step 1: Begin by loading a container or bag with materials/items utilized as a weight.

Step 2: Weigh the container or bag

Step 3: Set up the prototyped winch mechanism with the desired length and type of string that will be added to the robot for the climbing of the bar

Step 4: On the other end of the string, attach the container or bag with the weight

Step 5: Measure the height at which the string will travel from the starting position (where the bag is stationary on a flat surface), to where the string is at the end of the winch mechanism (or where the string begins to wrap around the pulley)

Step 6: When previous steps 1-5 are complete, begin and repeat the tests three times, where the times are measured and documented down

Step 7: Repeat steps 1-6 for all desired testing variations (i.e. weight, RPM of motor, difference in spool radius, etc.)

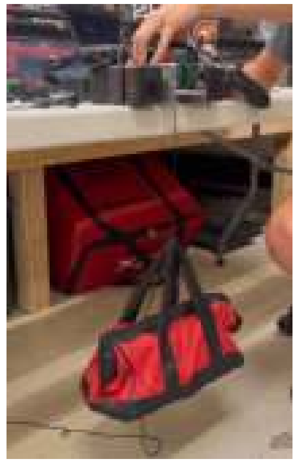


Figure 1: Testing rig

Data/Testing

Table 1: Testing Utilizing Different Weights (4 Motors, 200 RPM, 1:1 Gear Ratio)

Weight Used (lbs)	Height Traveled	Time Measured #1	Time Measured #2	Time Measured #3	Average Time (seconds)	Change in Height (in/average time, sec)
2.92	23 inches	4.64 sec	4.72 sec	4.68 sec	4.680	4.915
11.75	15 inches	3.72 sec	3.67 sec	3.84 sec	3.743	4.008
14.90	16.5 inches	4.34 sec	4.31 sec	4.25 sec	4.300	3.837

Table 2: Difference in Spool Diameter (4 Motors, 200 RPM)

Diameter (inches)	Height Traveled	Time Measured #1	Time Measured #2	Time Measured #3	Average Time (seconds)	Change in Height (in/average time, sec)
0.375	16 inches	3.13 sec	2.95 sec	3.04 sec	3.040	5.263
0.500	16 inches	2.38 sec	2.51 sec	2.47 sec	2.453	6.523

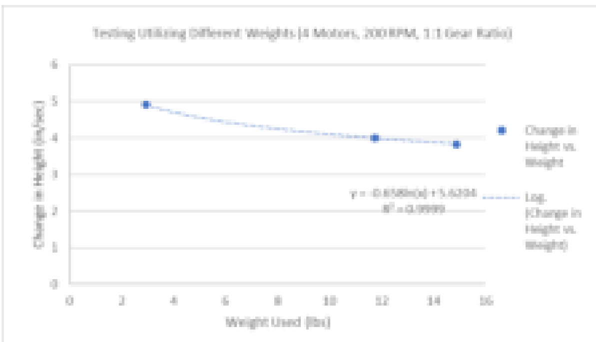
Table 3: Change in RPM (4 Motors, 1:1 Gear Ratio, 0.375-inch Diameter Spool)

RPM	Height Traveled	Time Measured #1	Time Measured #2	Time Measured #3	Average Time (seconds)	Change in Height (in/average time, sec)
200 (initial basis)	16 inches	3.13 sec	2.95 sec	3.04 sec	3.040	5.263
600	16.5 inches	1.74 sec	1.69 sec	1.68 sec	1.703	9.689

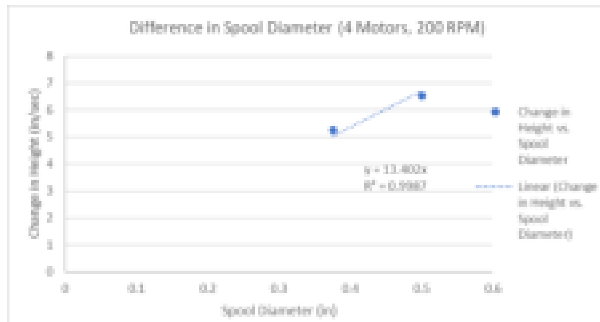
Table 4: Higher RPM Testing with Different Spool Diameter (4 Motor, 600 RPM)

Diameter (inches)	Height Traveled	Time Measured #1	Time Measured #2	Time Measured #3	Average Time (seconds)	Change in Height (in/average time, sec)
0.375	16.5 inches	1.74 sec	1.69 sec	1.68 sec	1.703	9.689
0.500	16.5 inches	1.76 sec	1.80 sec	1.72 sec	1.760	9.375

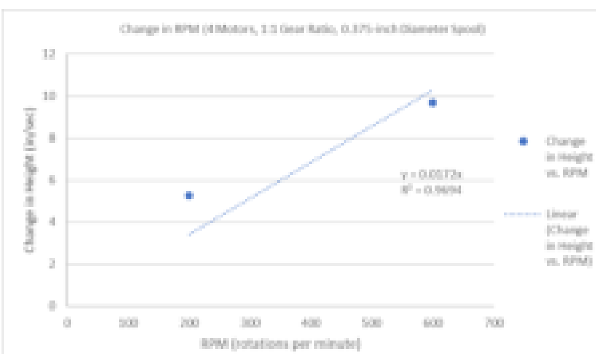
Plots of the subsequent testing and data can be found below:



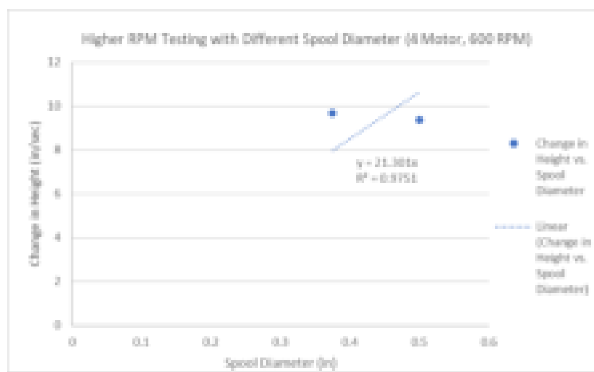
Plot 1



Plot 2



Plot 3



Plot 4

Observations & Testing Outcomes

Table 1 Test - Testing Utilizing Different Weights (4 Motors, 200 RPM, 1:1 Gear Ratio)

- Notes:
  - Testing was done with a current draw of around 1.3 Amps and with an RPM of around 84 - 87.5% due to the current and battery
- **Observed that the mechanism struggled the more weight it needed to lift, as well as the higher it needed to go**
  - This presents an issue, as the final desired height is around 43 inches (29 inches from the pole/climb and 14 inches from the robot itself)

Table 2 Test - Difference in Spool Diameter (4 Motors, 200 RPM)

- Notes:
  - Testing was done with a current draw of around 1.3 Amps and with an RPM of around 84 - 87.5% due to the current and battery
- Observed that **with a change in the spool diameter, the change in height increased by around 23.94%**
  - This is because, with a larger spool diameter, it effectively changes the gear ratio of the system, without the need to change the actual gear itself.

Table 3 Test - Change in RPM (4 Motors, 1:1 Gear Ratio, 0.375-inch Diameter Spool)

- Notes:
  - For the 200 RPM test, a current draw of around 1.3 Amps and with an RPM of around 84 - 87.5% due to the current and battery
  - For the 600 RPM test, a current draw of 1.6 - 1.8 Amps was used with the RPM of around 61 - 75% due to the current and battery
- **Observed that with a higher RPM, the change in height was much higher, while being able to hold the weight.**
  - This test disproved one of the concerns the team had, where at higher RPMs, the mechanism could not hold the weight that was needed to hold.

Table 4 Test - Higher RPM Testing with Different Spool Diameter (4 Motor, 600 RPM)

- Notes:
  - For the 0.375 inches test, a current draw of 1.6 - 1.8 Amps was used with the RPM of around 61 - 75% due to the current and battery
  - For the 0.500-inch test, a current draw of 2.2 - 2.5 Amps was found with the RPM of around 55 - 65% due to the current and battery

Brainstorming and Decision Matrix: 24″ Triball Scoring Mechanism

Date	@October 11, 2023
Category	24" Robot BLRS2 Skills
Authors	Ⓜ Max Ⓚ Kathleen Lowe
Design Cycle	Design Cycle 1

Identify the Problem

After catching Triballs shot by the 15″ robot, the **24″ robot will need to score the balls into the goal**, and that mechanism needs to be designed.

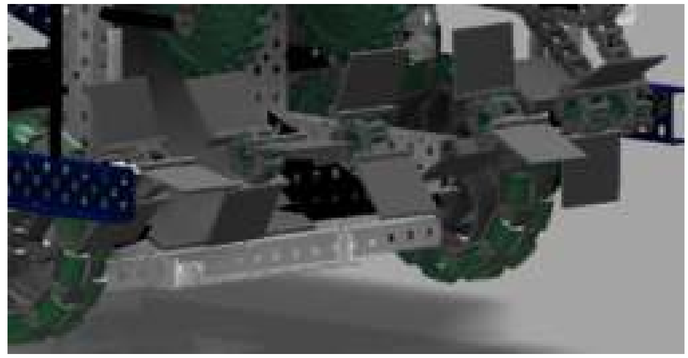
- Consistent scoring with little driver/programming interaction
- High speed to out pace the shooting from the 15″ robot
- Light weight design

Brainstorm Solutions

Solution 1:

- Top Roller
    - Similar to those of designs used by many VRC teams, **one large rubber band roller to feed into the goal from the basket.**
- Insert sketches/initial CAD/inspiration from other teams with reference

Rubber Tube intake from WPI VEX U team GOATS



Flapped Top Roller from Purdue SIGBots Wiki and 929U Turning Point

Solution 2:

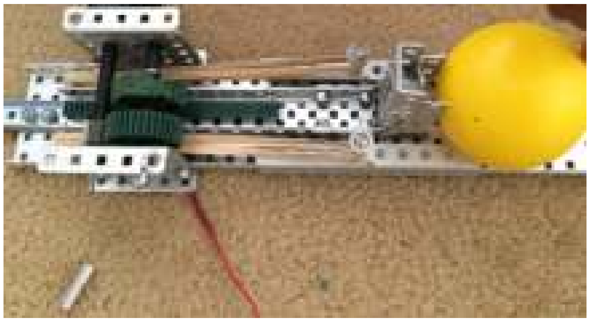
- Side Rollers
  - With similar function to the top roller, using **2 side wheels to force the Triballs into the goal.** Likely, a series of rollers will be used to align the Triballs in the center of the two rollers.

7k Tower Takeover render with flapped side rollers from the Purdue SIGBots Wiki

Flex Wheel side rollers from 94999E and the Purdue SIGBots Wiki

Solution 3:

- Linear Puncher
  - With the 4 motors being connected to our scoring mechanism and winch, a linear puncher with a custom slide can be used to force the Triballs into the goal. **The linear puncher would sit low to the ground and push Triballs parallel to the ground.**



A linear puncher using a slip gear from 574C and the Purdue SIGBots Wiki.

Select the Best Solution

To compare each of the three scoring mechanisms, our team created a list of pros and cons for each option then created a decision matrix

Pros

Top Roller	Side Rollers	Linear Puncher
Simple as only one roller(and shaft) will be needed	Continuous Scoring	Lots of power for packing
Light weight since only one roller is needed	Consistent ball placement(center of drive)	Can disregard tightness of goal/pole height
Continuous scoring	Low profile	Easy to power with central axle
		Wide lateral range of ball placement

Cons

Top Roller	Side Rollers	Linear Puncher
Requires consistent traction	Requires additional centering mechanism	Not Continuous Scoring
Requires consistent compression	Requires consistent traction	Requires additional sensors
Compression will reduce drive traction	Requires consistent compression	Heavy wear on robot

Decision Matrix

The following criteria were considered while we decided upon the ideal scoring mechanisms

- Component Complexity (x/10)
    - With a robot that needs to serve one single function for 90% of it's usage (only in skills. Only placing Triballs) **we need to ensure that there are not too many points of potential failure.**
    - A higher score indicates a less complex design.
- Wear (x/5)
    - With the mechanism needing to score 44+ Triballs per run, and 6 runs at a competition, **the part can not be reliant on mechanisms that will change over time.**
    - A higher score indicates a longer lasting mechanism.
- Additional Components (x/10)
    - This robot needs to be light weight and consistent, **so removing additional components that would add weight** (centering balls, alignment, lifting goals, etc.) is ideal for weight saving and longevity.
    - A higher score indicates less additional components.
- Speed (x/5)
    - Scoring 44+ Triballs is a difficult feat in under a minute, and this **mechanism must score them faster than the 15″ robot can shoot them.**
    - A higher score indicates faster scoring speed.

	Component Complexity (x/10)	Wear (x/5)	Additional Components (x/10)	Speed (x/5)	Total (x/30)
Top Rollers	5	3	6	4	1 8
Side Rollers	3	3	5	3	14
Linear Puncher	7	4	8	3.5	22.5

After considering each of the three solutions, our team has decided to move forward with a **Linear Puncher.** This is because of the following considerations:

- This mechanism will be able to keep the motors utilized for it, and the lifting mech central in the robot without complex gear trains adding friction and taking up space. Additionally, this mechanism has lots of mechanical advantage and doesn't rely on traction **making it powerful and consistent when run for long periods of time during a skills run.**

Implementation Plan

- Draw additional sketches refining the plan for the design
  - Simultaneously figure out the design for the catching mechanism.
- CAD and prototype the solution
  - We will aim to have this done by the end of October, or **as soon as we can allowing for time to perfect the chassis.**
- Implement the solution into the robot
  - As before, we will aim to have this done **as soon as the chassis is complete.**

Design and Building: 24″ Chassis

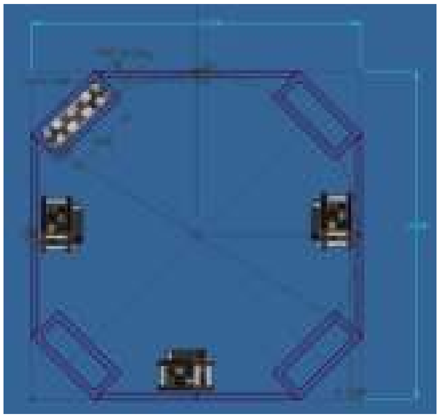
Date	@October 18, 2023
Category	24" Robot BLRS2 Skills
Authors	Ⓒ Conner Siebert Ⓜ Max
Design Cycle	Design Cycle 1

Goals

- As we start to develop the first iteration of the 24″ skills robot, development of **the base will begin with the chassis design.**
- To achieve this, we plan to use Inventor 2023 to design the chassis before beginning to manufacture it.
- The main goals of our chassis is to have the following design considerations:
    - Be lightweight
        - **Being lightweight will allow for a fast chassis that is easier to hang**
        - To test after we build, a successful chassis will be able to cross the field in **under 3 seconds**
    - Be around 18″ in both horizontal directions
        - While the robot can be up to 24″, we believe a 24″ footprint of the chassis will allow for a bot that is still able to easily maneuver the field

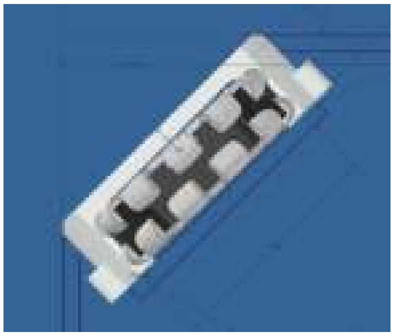
Design Process

- Outlining The Chassis
    - The first thing that needed to happen was outlining the size of our chassis. As previously discussed, **we want to use 3.25″ omni wheels with 600rpm motor cartridges at a roughly 400rpm output speed.**



Initial Chassis Outline

- Creating Wheel Pods
    - After outlining the chassis, we needed to create **4 identical wheel pods to house motors and gears.** To do this, we want the pods in 2 parts where one houses the motors, and the other is simple - **the steps to create each pod can be found in the caption of each picture.**



Top view of wheel pod around wheel

Back view of wheel pod casing around wheel

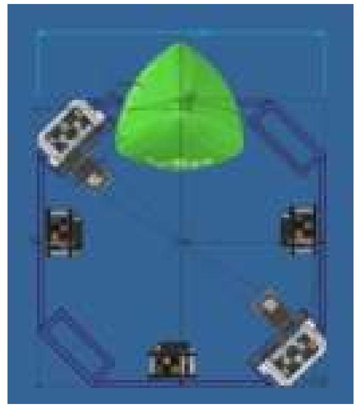
Isometric view of wheel pod

Generated gears at 18 to 28 ratio

Offset gears to increase spacing at front of chassis

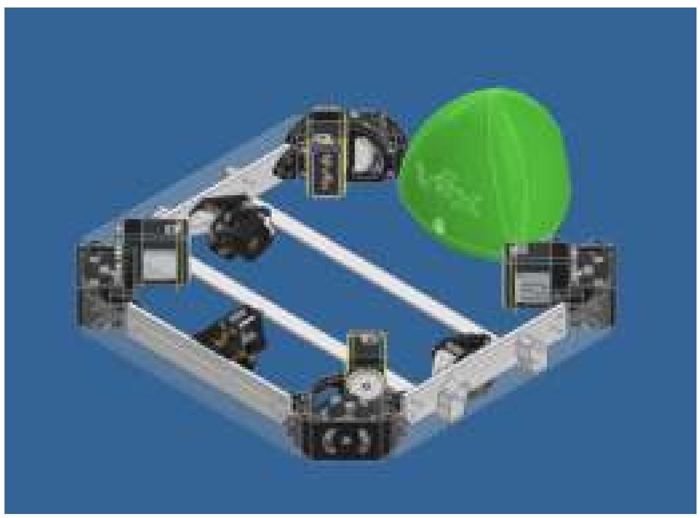
Inside wheel pod piece with motor mount

Front view of complete pod



Top view of drive pods

- Adding Side Bracing
  - With the pods designed, we need to connect them together. **The plan for this is to have two layers of 1/4″ poly carbonate sheets along each side.**



Completed Chassis CAD

- Final Product
  - The design of the side panels was simple, as parametric modeling allowed for easy connection across all 4 identical parts. **The sheets were designed to use heat set inserts to mount screws.** Additional braces were mounted on the left and right side to sandwich the pieces together.

- This chassis is designed to have large open spaces in the middle as well as being very light weight (estimated 4.5lbs (2lbs are motors)). **This chassis should also be simple to construct as many parts are identical.**

Build the Solution

To build the chassis, our team used the following steps:

1. The motor mounts, outside motor pieces, drive gears, and wheel inserts with gears were all 3D printed as shown in figure 1
2. The 1/4″ polycarbonate frame pieces were machined using the following process:
    a. Templates were printed and taped onto the polycarbonate sheet as seen in figure 2
    b. Using a vertical band saw, each piece was cut following the template as shown in figure 3
    c. Using a drill, each hole in the sheets were drilled. A de-burring tool was used to remove any excess polycarbonate left around the edges of the holes
    d. A final polycarbonate frame piece is seen in figure 4
3. The heat sink inserts are placed on a soldering iron to heat them. They are then placed into the holes in the 3D prints, where they melt their way in as seen in figure 5.
4. After the inserts have cooled, they will be bonded to the 3D printed PLA. This will allow screws to be attached and threaded into the 3D prints.
5. A hole is drilled through the center square hole of the vex IQ omnis 2.51″, as they will be free spinning on a screw joint.

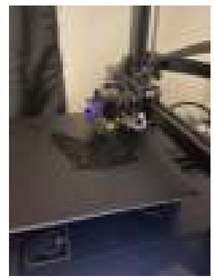


Figure 1: Conner's Ender 3 printing a motor mount to attach to the chassis

Figure 2: Template placed onto the polycarbonate sheet

Figure 3: Polycarbonate sheet being cut by the band saw

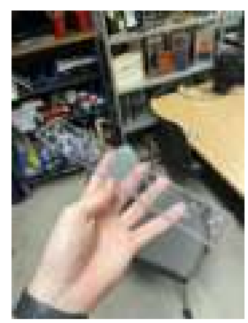


Figure 4: Finished polycarbonate frame piece

6. The 3D printed 28 tooth gear and hub are installed into the 2.51″ Vex IQ omni wheels. The assembly is seen in figure 6.
7. The motor, with a 600 rpm cartridge, is rubber banded onto the motor mount. This practice, commonly know as *hot swapping,* allows the motor to be strongly secured to the 3D print. Since the 4 motor screws are not installed, the motor can be easily removed in case of a failure of the motor.
8. The chassis corner 3D printed chassis pieces are attached to the polycarbonate frame pieces as shown in figure 6.
9. The 4 wheel pods are assembled, as seen in figure 7 using the following parts:
    a. Omni wheel assembly
    b. 19 tooth drive gear
    c. Motor shaft
    d. Inner and outer 3D printed pieces with heat sink inserts
    e. 3/8″ screws
10. The 1/4″ Aluminum flat bar is cut into two 18″ long segments. Two 1/8″ screw holes are drilled into each end of the bar to allow screws to pass through and attach to the bottom of the motor mount. This is seen in figure 8.
11. As seen in figures 9 and 10, the flat bars are attached to the bottom of the chassis, with screws going though the bar into the heat sink inserts.


Figure 5: Soldering iron installed the heat sink inserts

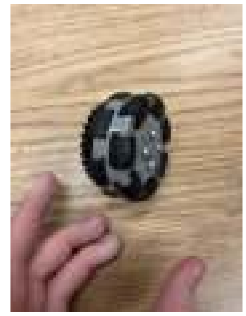


Figure 6: Vex IQ omni with gear hub assembly installed


Figure 7: 3D printed wheel pods being attached to the polycarbonate frame pieces


Figure 8: Assembled Wheel Pods


Figure 9: 1/4″ Aluminum flat bar

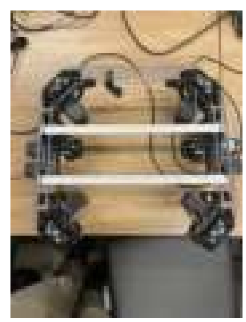


Figure 10: Cross braces attached to the chassis

Test Solution

With the chassis build completed, we can move onto testing for any revisions that may need to be made. As the chassis itself has no other objectives than to move quickly around the field, we will **test the speed at which we are able to move from one end to the other** using the following procedure:

1. Position the chassis with one end against the wall.
2. Power the chassis at full speed forward.
3. Record the time it takes for the chassis to reach the opposite side of the field.
4. Repeat.

Trial #	Speed (seconds)
1	2.23
2	2.14
3	2.32
4	2.75
5	2.11
Average	2.31 s

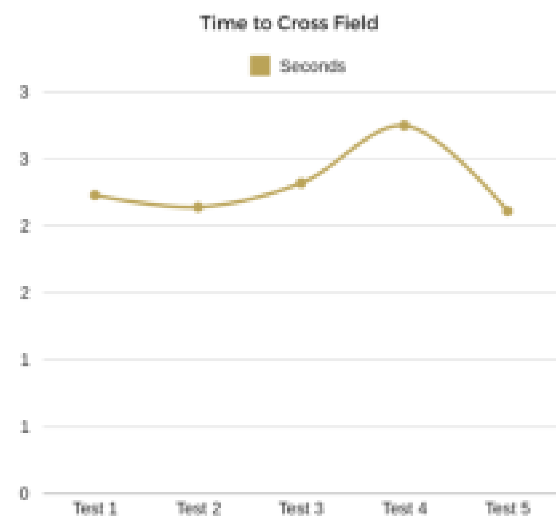


Figure 11: Testing graph

While the chassis does meet our goal of crossing the field in under 3 seconds, we did notice some inconsistency in the speed of the chassis. We believe that some manufacturing defects may be causing additional friction in the wheels, reducing our speed. We will discuss our findings in the next chassis entry.

Planning: 24″ Motor and Pneumatic Distribution

Date	@October 26, 2023
Category	24" Robot BLRS2 Skills
Authors	Ⓒ Conner Siebert Ⓜ Max
Design Cycle	Design Cycle 1

Define the Problem

While planning our first iteration of the skills robots, determining the distribution of motors and pneumatics will provide a base for the design criteria. **The potential solutions will revolve around powering three main subsystems: chassis, Triball catapult, hanging lift, and future subsystems including wings.**

There are three main items to power the robot are:

- 11 Watt Motor
  - The first motor of the V5 system, 11 W motors provides the base for most mechanisms in the Vex Robotics Competition
  - 11 W motors have three possible outputs: 100 rpm, 200 rpm, and 600 rpm
  - After 8 motors have been connected to the V5 brain, the current of 2.5 amps begins to be limited with each additional motor. **Thus, there is a diminishing marginal return with each additional motor over 8**
- 5.5 Watt Motor
  - New this season, the **5.5 W motor provides a smaller footprint compared to the 11 W motor but only half other power**
  - The 5.5 W motor can only output at 200 rpm
  - A 5.5 W motor limits the current in the same way as a 11 W does
- Pneumatic Pistons
  - These linear motion actuators provides a motion outside of the rotational outputs of motors
  - **Additional hardware is required to add pneumatics to a robot, with an air reservoir and solenoids are required, taking space and adding weight to the robot**
  - The use of pneumatics has no affect on the performance of motors

Potential Solutions

Solution 1: Ten 11W Motors, One 5.5W Motor, Two Pistons

- 11W Motors: 6 Motor Chassis, 2 Motor Catapult, 2 Motor Winch
- 5.5W Motors: 1 Motor Elevation Bar Clamp
- Pistons: 2 Wings
- **This solution focuses minimizing the use of pneumatics on the robots**
- The use of only 2 motors on the winch will limit the hanging speed, as this subsystem has the largest torque requirement of the entire robot
- Total Weight = (10 11W motors) * (0.342 lb/motor) + (1 5.5 W motor) * (0.2 lb/motor) + (2 pistons) * (0.131 lb/piston) = 3.862 lbs
- **Current per motor = 2.21 A**

Solution 2: Twelve 11W Motors, One 5.5W Motor, Two Pistons

- 11W Motors: 6 Motor Chassis, 2 Motor Catapult, 4 Motor Winch
- 5.5W Motors: 1 Motor Elevation Bar Clamp
- Pistons: 2 Wings
- This solution continues the focus of minimizing the use of pneumatics on the robots
- To increase our lifting speed, the number of motors on the winch is doubled in this solution
- Total Motor Weight = (12 11W motors) * (0.342 lb/motor) + (1 5.5 W motor) * (0.2 lb/motor) + (2 pistons) * (0.131 lb/piston) = 4.564 lbs
- Current per motor = 2.05 A

Solution 3: 8 11W Motors, One 5.5W Motor, Three Pistons

- 11W Motors: 4 Motor Chassis with 4 motors shared to scoring mechanism and winch
- 5.5W Motors: 1 Motor Elevation Bar Clamp
- Pistons: 2 Wings, 1 PTO Shifter
- The winch is still able to operate at a high speed, as 4 motors from the chassis will be operating it
- **The speed of the scoring mechanism will also be increased by the use of the 4 motors compared to 2 in solutions 1 and 2**
- Total Motor Weight = (6 11W motors) * (0.342 lb/motor) + (1 5.5 W motor) * (0.2 lb/motor) + (3 pistons) * (0.131 lb/piston) = 2.645 lbs
- Current per motor = 2.50 A (max current per motor)

Choosing The Best Solution

To compare each of the three motor and pneumatic combinations, our team created a list of pros and cons for each option then created a decision matrix

Pros

Solution 1	Solution 2	Solution 3
Lowest mechanical complexity	High speed winch	Lightest Option (by 1.217 lbs)
	Low mechanical complexity	No current limiting to motors
		High speed winch
		Highest torque for catapult

Cons

Solution 1	Solution 2	Solution 3
Small levels of current limiting (0.29 A lost per motor)	Very high current limiting (0.45A lost per motor)	High mechanical complexity
Heavy Option	Heaviest Option (motors and pistons are near half of the target weight of 10lbs or less for these robots)	Additional pneumatics are required for PTO Shifter
Low winch speed	Large lost of space in the robot to motors	

Decision Matrix

The following criteria were considered while we decided upon the motor and pneumatic distribution

- Weight (x/10)
  - As weight is a large consideration in each stage of the design for the skills robots, **minimizing the weight of additional components is important**
    - For this reason, the impact of weight is doubled in the decision matrix considerations
  - A higher score indicates a lighter robot
- Mechanical Complexity (x/10)
  - As the **designability, constructability, and maintainability of the robot is important throughout the design process**, the mechanical complexity of each option will be analyzed
    - The mechanical complexity was also given double the point values in the decision matrix due to its importance
  - A higher score indicates a simpler design
- Winch Speed (x/5)
  - As each second of a skills run is optimized, **the speed in which we will be able to climb will be considered**
  - A higher score indicates a faster winch

- Catapult Torque (x/5)
  - Similar to the winch, **maximizing the speed and torque of the catapult is also important**
  - A higher score indicates a more powerful catapult
- Current Limiting (x/5)
  - As each motor is placed on the robot, the **efficiently of the motor should be maximized to justify it's addition to the robot**
  - A higher score indicates a more power-efficient design

	Weight (x/10)	Mechanical Complexity (x/10)	Winch Speed and Torque (x/5)	Catapult Speed and Torque (x/5)	Current Limiting (x/5)	Total (x/35)
Solution 1	5	10	1	1	2	19
Solution 2	3	8	5	1	1	18
Solution 3	**10**	**3**	**5**	**5**	**5**	**28**

After considering all three options, solution 3 was the clear best choice to move forward with. This is because of the following factors:

- Less Motors = Less Weight
  - As this option had 2 and 4 motors less than solutions 1 and 2 respectively, **the over one pound difference in weight is over 10% of our target weight**
- Winch and Catapult are maximized
  - With four motors from the chassis powering both of these subsystems, they will be able to operate with high torque and speed
- **All motors are operating at full potential**
  - With only seven motors total, there is no current limiting occurring on the robot while the other two solutions went over the eight motor limit

Implementation Plan

As each design stage of the robot occurs in the coming months, these decisions will be carried out. It is our goal to stick to this configuration throughout the first iteration of the skills robots.

Sources

Motor Weights and Current Limiting

https://wiki.purduesigbots.com/vex-electronics/vex-electronics/motors

Pneumatic Piston Weight

https://www.smcpneumatics.com/NCJ2D10-200.html

Task and Timeline Update: November

Date	@November 3, 2023
Category	15" Robot 24" Robot BLRS2 Skills
Authors	C Conner Siebert K Kathleen Lowe
Design Cycle	Design Cycle 1

October Recap

While in October our team planned to complete the CAD for both robots, our progress was delayed from the initial timeline.

- The 24″ robot was set back by changes in the chassis design
    - While version 1 included a all 3D printed and polycarbonate chassis, this design was concluded be an insufficient option due to:
        - Hard to manufacture (errors in cutting and drilling tolerances)
        - Lack of rigidity
        - Risk of failure in several locations
- The 15″ robot had a major success in the design of the chassis
    - The manufacturing of the chassis frame for the 15″ was delayed due to increased delivery time of metal.
    - **The chassis will be ready to machine within the first week of November**
- Our team **decided running skills on December 2nd will not be happening.** This is due to the following concerns:
    - The 24″ chassis redesign caused the predicted schedule to fall behind
    - By running skills on December 2nd, **we expose our strategies to other VEX U teams**, weakening our chances at the World Championship
    - By rushing the development both robots, **we would risk not having time to program a complete programming skills run**

November Task and Timeline

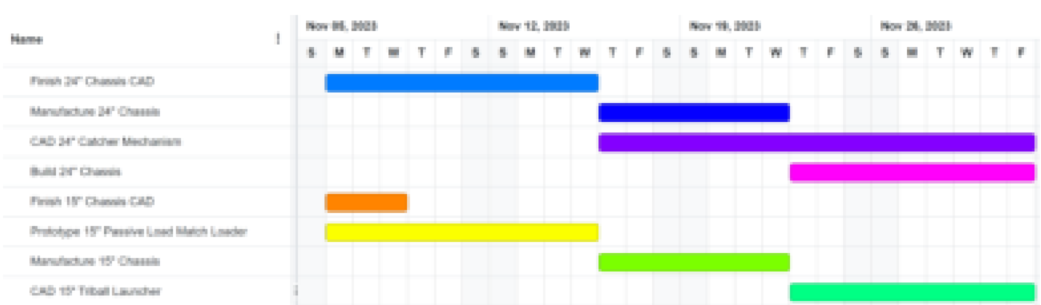
Building off of the delayed developments in October, **our team plans to use November as a month of mechanical progress in design and building of the two robots.** We also plan on losing a week of work time, due to most team members traveling home for Thanksgiving break.

24″ Robot

- With the chassis redesign in process, the **first two weeks of the month will be focused on completing the CAD**
- In the 2nd half of the month, our team plans to complete the manufacturing and building of the chassis.
- With the chassis CAD completed, we also plan to begin work on designing the catching mechanism.
  - **This work will extend into December**, as we feel it is important to flush issues out with the design throughout the process and not make mistakes this early in the season.

15″ Robot

- As found in October, **the chassis CAD only needs a little work in the first days of the semester to complete.**
- After the chassis CAD is completed, a large portion of time will spent developing the passive load launcher for the 15″ robot
  - In order to be accurate in programming and driving skills, we feel **it is important for the catapult to shoot constantly into the launcher**
  - Speed of the loader will also be important, as it will need to shoot all 44 match loads
- Once the chassis CAD is complete, it manufacturing the chassis will also be important



Brainstorming and Decision Matrix: 24″ Chassis Redesign

Date	@November 9, 2023
Category	24" Robot BLRS2 Skills
Authors	Ⓒ Conner Siebert Ⓜ Max
Design Cycle	Design Cycle 2

Identify the Problem

After designing and building the first iteration of the skills 24″ chassis, **several concerns with the design were found. These includes:**

- Inaccuracies in manufacturing
  - As the holes in the polycarbonate had to be drilled by hand, the **screws were often misaligned**
  - **Several of the heat sink inserts got off centered while installing**, causing further issues with the holes in the polycarbonate
- Lack of structural stability
  - Even with the cross braces installed, **the polycarbonate sides still exhibited a tendency to flex while the chassis was driving around the field**
  - The fully 3D printed mounts with heat sink inserts also were causes of possible future concern, **as several weak points could lead to breakage and screws coming loss**

To fix the chassis, our team plans to **decide between a new design or altering the current chassis**

Brainstorm Solutions

Solution 1: Continue with the current chassis

- With the current chassis, **several poorly manufactured parts could be re-machined to improve on the current issues**
  - This option would be the quickest, as no design work would be required
  - The current issues could return, meaning extra care to detail would be required in future building

Figure 1: Completed Current Chassis

Solution 2: Redesign the current chassis

- Several aspects of the current chassis that has issues could be improved upon:
  - Replace polycarbonate with flat bar
  - Increase strength of 3D prints
- This **option would requires larges amount of time**, as redesigning the current chassis would have several time-consuming tasks to complete

Solution 3: Complete re-design

- With the current **15″ robot chassis showing promise in design, manufacturing, and structural stability, re-designing the 24″ robot to match this chassis is an option**
  - By starting over, all of the issues found in the first chassis could be completely erased
  - This option would require the most work, as it would be starting from ground zero again

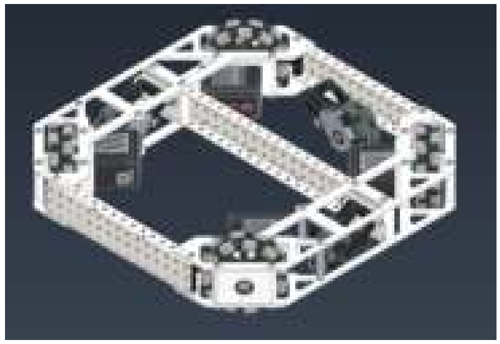


Figure 2: Completed 15″ Chassis CAD

Select the Best Solution

Pros

Solution 1	Solution 2	Solution 3
Fastest Option	Improves the structural stability issue	Allows for the shortfalls of the current chassis to be removed
Easiest Option	Is proven to be a viable solution	Proof of concept is already proven in the 15″ Chassis design
Is proven to be a viable solution	Is proven to be a viable solution	Precision machining is available using the laser cutter
		Structural stability will be greatly improved

Cons

Solution 1	Solution 2	Solution 3
Requires precision in machining	Requires extra time, setting back the timeline	Requires extra time, setting back the timeline
Poses to not fix many of the current issues	Does not improve the issues with the heat sink inserts	Risk having unforeseen issues
Structural Stability will not be improved	Will greatly increase the weight of the chassis	

Decision Matrix

The following criteria were considered while we decided upon the improvements to the chassis

- Time Required (x/5)
  - As the skills bots are currently trending towards being behind schedule, **the additional design time will be a consideration**

  - A higher score indicates less time required
- Structural Stability (x/10)
  - As the chassis is the base of the robot, **being able to securely mount further subsystems to it is required**
  - For the remainder of the season, a chassis that will **withstand competition and require low maintenance is important**
  - A higher score indicates a more structurally sound chassis
- Manufacturability (x/5)
  - Two shortfalls of the current chassis are the **time and precision included with the manufacturing process, which should be minimized in future chassis designs**
  - A higher score indicates an easier to manufacture chassis
- Risk of Future Issues (x/10)
  - **Lowering the future issues with the chassis** is the overall design goal of the redesign
  - A higher score indicates a lower risk design

	Time Required (x/5)	Structural Stability (x/10)	Manufacturability (x/5)	Risk of Future Issues (x/10)	Total (x/30)
Solution 1	4	3	1	2	10
Solution 2	2	7	3	6	18
Solution 3	3	10	5	8	26

After considering each of the three solutions, our team has decided to move forward with solution 3. This is because of the following considerations:

- Removes all issues with the current chassis
  - There is no longer any overcasting issues with the chassis that will need improved upon
  - The new chassis can be 3D printed and machined using a laser cutter, removing any human error in the building process
  - The structural stability issues are removed
  - The skills 15″ chassis has proved that the design is viable, and will only require scaling it up

Implementation Plan

- Work will begin on re-designing the chassis
  - Inspiration will be taken from the successes of the 15″ chassis, while being scaled up to the desired size of the 24″ robot
  - Aim to complete design and implementation **by the end of December**. A rough timeline is included below:
    - **November 16**
      - Observe construction of the 15″ chassis to determine any issues needing to be resolved
    - **November 20**
      - Construct the rebuild 24″ chassis
      - Test the new chassis

Design and Building: 15″ Chassis

Date	@November 16, 2023
Category	15" Robot BLRS2 Skills
Authors	Conner Siebert Kathleen Lowe

Goals

- **Inventor will be used to design all custom parts** before manufacturing and assembly of the 15″ skills robot
- As determined prior, the chassis should use 4 motors in an x-drive configuration to best match the skills strategy
- The chassis design **should result in a lightweight chassis that is easy to mount other subsystems to**

Design Process

- To begin the chassis design, our team decided that **using a combination of aluminum, 3D printed parts, and VEX c-channels would provide the most desirable weight to strength ratio.** The following items were designed with each consideration:
    - Wheel and motor mounts: 3D printed
        - To minimize the friction of each of the 4 drive pods, it was decided a direct drive from the motor along with ball bearings would be used
        - **3D printing the motor mounts allow for a low profile solution**, allowing additional space for other subsystems
        - The ball bearings are able to be easily pushed into the 3D printed mount
    - Machined aluminum
        - To best mount each side of the chassis together, two 1/4″ thick aluminum chassis pieces will hold together 2 drive pods
        - By using custom pieces, the **45-degree angles of each drive pod are easily created without additional brackets**
        - These brackets are seen in figure 1 below, ready to be exported for machining

Figure 1: Completed CAD for Machined Aluminum Chassis Pieces

- VEX C-Channel
  - To cross brace each side of the chassis, standard VEX c-channels will be used.
  - **These cross braces will allow for easily mounting future sub systems to the robot**
  - As we already have these pieces available, there are no additional cost required compared to 3D printed pieces and sheet metal

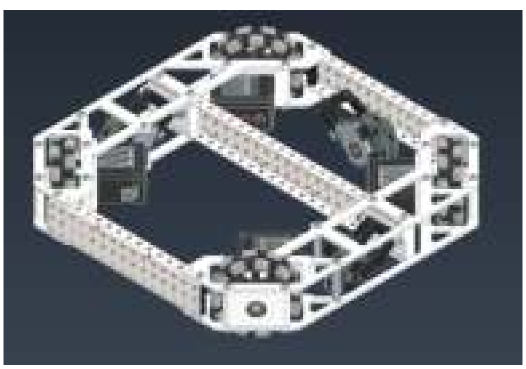


Figure 2: Completed Chassis CAD

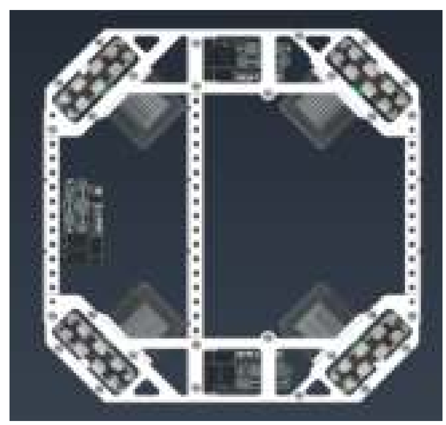


Figure 3: Top View of Completed Chassis CAD

Build the Solution

1. To begin the manufacturing of the chassis, Dominic and Joe went to the mechanical engineering shop to machine side chassis pieces:
    a. The **2′x4′ 1/8″ aluminum sheet is placed into the Bescutter fiber laser,** shown in figure 4.
    b. The designed chassis pieces are **uploaded to a .dxf file in AutoCAD**, which is exported to the Bescutter Laser Fiber as seen in figure 5.
    c. After cutting, the final machined pieces are removed from the machine. They then were **de-burred, removing any extra metal along the cuts.** A cut chassis piece is seen in figure 6.
2. To begin the chassis assembly, the **motor is attached to the 3D printed mount.** This low profile solution is seen in figure 7.
3. Two machined chassis pieces are **assembled with the outside supports for the drive pods.** A ball bearing is placed into the 3D printed wheel support pieces. shown in figure 8.


Figure 4: Bescutter Fiber Laser for Sheet Metals


Figure 5: Chassis Pieces for the skills 15″ and competition robot in the .dxf file


Figure 6: Machined Chassis Piece


Figure 7: Motor attached to custom 3D printed mount


Figure 8: Ball bearing placed in the 3D print


Figure 9: Motors mounted to the chassis frame

4. The **motor pods are attached to each of the two chassis assemblies**, as seen in figure 9. Standoffs are also added between the two frame pieces to **add structural support to the frame.**
5. As figure 10 shows, two 3-wide c-channel pieces are mounted between the two halves of the chassis, **creating a U-shape of the chassis.** The c-channels are boxed, which will prevent the flanges of the metal from bending in. **This will prevent deformation to the structure.**
6. The **3.25" omni wheels are attached to the direct drive 200 rpm motor** in each of the 4 drive pods. A top view of this is seen in figure 11.
7. The assembled encoder pods are **attached to the machined chassis frame pieces on a screw joint** as shown in figure 12.
8. A brain and battery were **attached to the cross brace of the chassis to test.** The completed chassis is shown in figure 13.

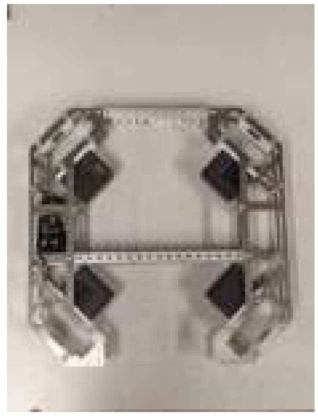


Figure 10: Cross braces attaches the two halves of the chassis frame together

Figure 11: Wheels added do the chassis pods

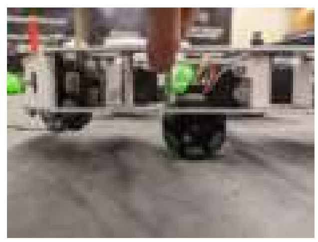


Figure 12: The encoder wheel and mount attached to the chassis frame

Figure 13: Completed chassis with brain attached for testing.

With the chassis completed, we plan to complete in-depth testing after we return from Thanksgiving Break. **This testing will be documented in a future entry.**

Design and Building: 24″ Chassis V2

Date	@November 20, 2023
Category	24" Robot BLRS2 Skills
Authors	M Max J Jacob Zawacki
Design Cycle	Design Cycle 2

Goals

- We will design a new chassis using Autodesk Inventor 2023.
- Using the steps developed in the CAD we will build the chassis.
- **After testing the previous chassis design and taking qualitative data, our goal for this new chassis is to reduce the number of unique parts**, unify the design with the 15″ robot, and increase the rigidity.

Design Process

- Creating Wheel Pods
    - To start out, we chose to use VEX IQ omni wheels since we have them in our shop and the **smaller diameter increases acceleration and torque**. The rubber from these wheels also has higher traction than 3.25″ omni wheels. Choosing to use these wheels will cause us to **adjust the gear ratio to obtain our desired speed to match 600rpm on 3.25″ wheels.**

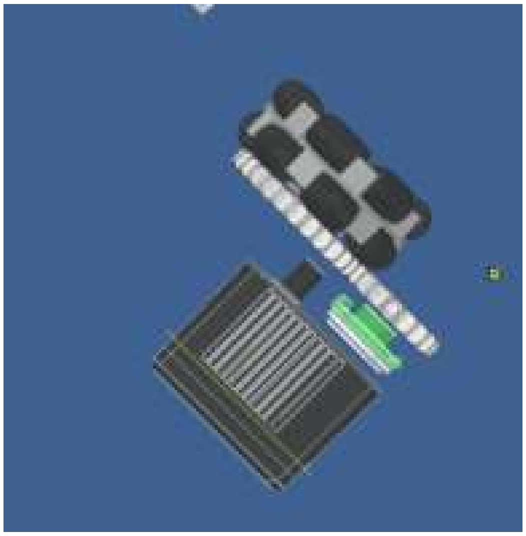


Figure 1:Top view of motor configuration. Key is compactness with motor rotation inwards

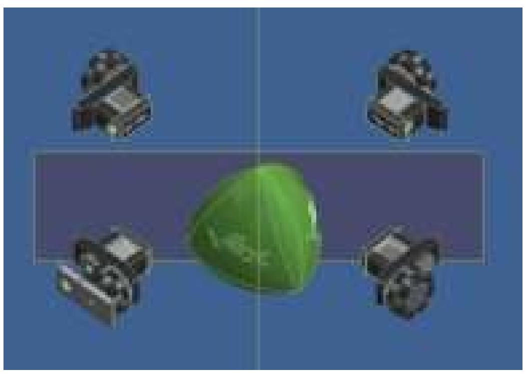


Figure 3: 4 pods with 2 total parts, and 2 gears

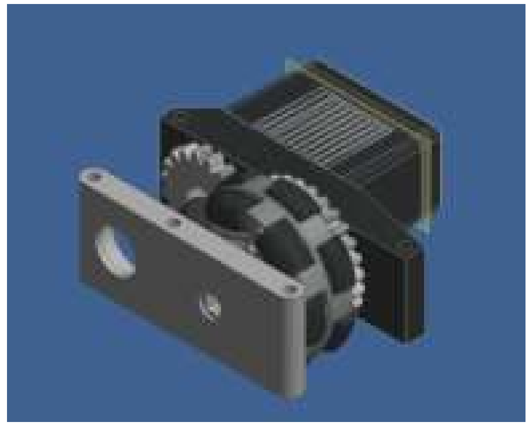


Figure 2: simple blocked mounting to be sandwiched with bracing for easy printing

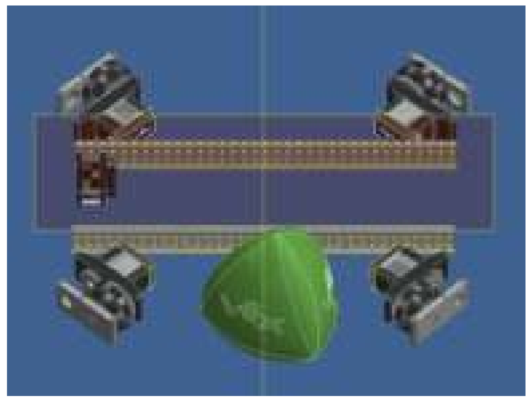


Figure 4: Adding alignment for c-channels internally

- Adding bracing sandwich
    - Starting by making a design similar to the 15″ skills chassis, **we connected the pods with two long aluminum pieces to sandwich the bracing.** We found that this would be too expensive to machine from aluminum, so we split up the four pods and replaced the bracing with a 3x c channel. This will save weight, add strength, and reduce cost drastically. **The 8 identical pieces can all be cut from one aluminum sheet that is 12″x24″.**

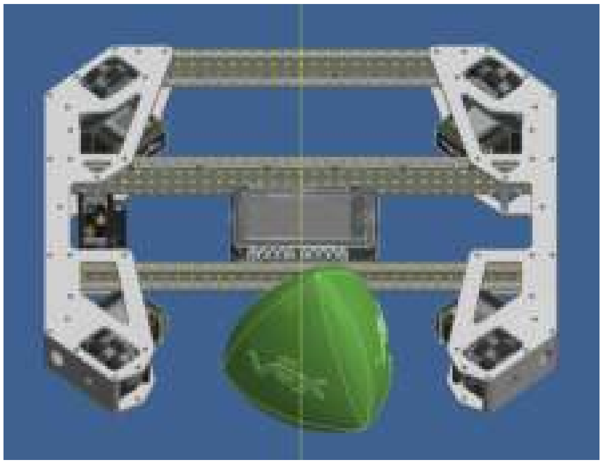


Figure 5: Initial sandwich brace design. This matches the design language of the 15″ chassis but with the larger footprint of 18″ makes it far too expensive to machine using 2 full 12×24″ sheets of aluminum

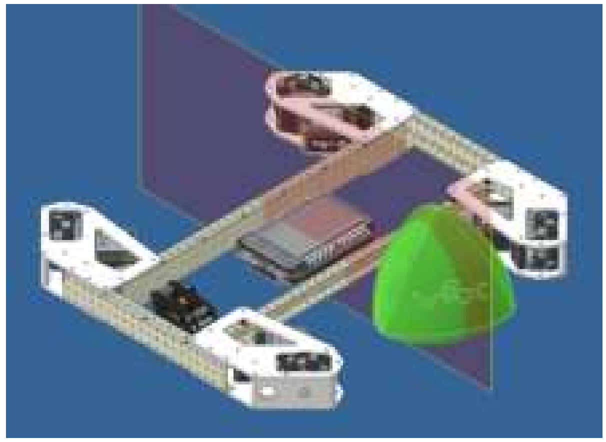


Figure 6: Step 2: We split up the pods and added 3x c-channels to make up for the bracing. Additionally, this may be stronger in the Z loading direction

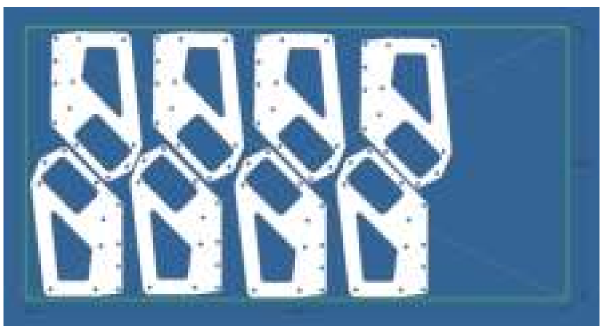


Figure 7: using a 12×24″ sheet of aluminum, we can make all 4 pods.

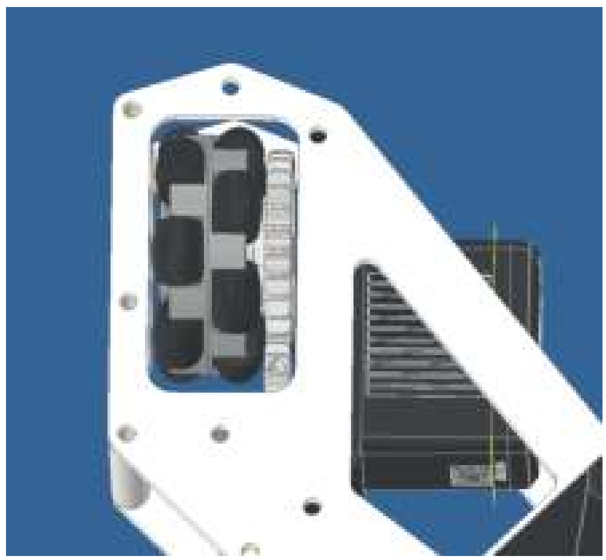


Figure 8: close up view of the wheels with tolerance to remove wheel. Additional holes are for extra stand off bracing.

Build the Solution

To test our initial chassis design, **the pieces set to be machined from aluminum were 3D printed and installed onto the robot.** This was one to lower the cost of the initial test, as **$80 worth of sheet metal would be required to build the full chassis.**

1. Each 2.5″ VEX IQ omni wheel is assembled using the following steps:
   a. The axle hole in the middle is drilled out, **allowing a screw joint to pass through the center of the wheel**
   b. The 3D printed hub, as shown in figure 11, is **attached to the wheel using 4 0.375″ screws**
2. Using a soldering iron to heat the inserts, they are pushed into the holes in the 3D print as they melt the PLA plastic around them. **As the PLA cools, the mount is hardened in place to prevent it from being able to be removed.**
3. The top and bottom 3D printed plates of each drive pod sandwich the wheel and motor mounts. **These are attached by 2″ screws passing through al 3 prints**, as seen in figure 13.
4. In figure 14, the hot swap motor is placed into the drive pod. The wedge piece is then installed behind the motor to **prevent the motor from moving from the pod.**


Figure 9: Printing vertical braces

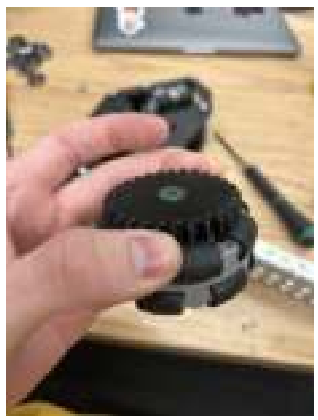


Figure 11: Wheel insert using thread in place screws


Figure 13: Assembled Drive pod top view


Figure 10: Completed printed pieces


Figure 12: Inserting heat set insert with soldering iron

5. Using 3-wide channels, each of the drive pods are connected. **Screws are ran through the top and bottom print pieces, passing through the c-channel creating a sandwich affect.** The braces shown in figure 16.
6. As shown in figure 15, the **brain is mounted between the two cross braces** on a custom 3D printed bracket.

After the chassis was assembled, we determined that the **3D printed top and bottom plates were structurally sound enough to support the chassis.** This allowed the first version of this chassis design to be used for the skills robot, saving the team time and money that would be required to use aluminum instead of PLA plastic.

Figure 15: Mop view of chassis

Figure 14: Motor removed from hot swap mount

Figure 16: Cross-braces of the drive pod

Figure 17: Isometric view of chassis

Figure 18: Side view of chassis

Test Solution

With the chassis build completed, we can move onto testing for any revisions that may need to be made. **We will be using the same procedure as the previous 24″ chassis so that we can measure for improvement**:

1. Position the chassis with one end against the wall.
2. Power the chassis at full speed forward.
3. Record the time it takes for the chassis to reach the opposite side of the field.
4. Repeat.

Trial #	Old Chassis Speed (seconds)	New Chassis Speed (seconds)
1	2.23	2.13
2	2.14	2.01
3	2.32	2.10
4	2.75	2.13
5	2.11	2.07
Average	2.31 s	2.09 s

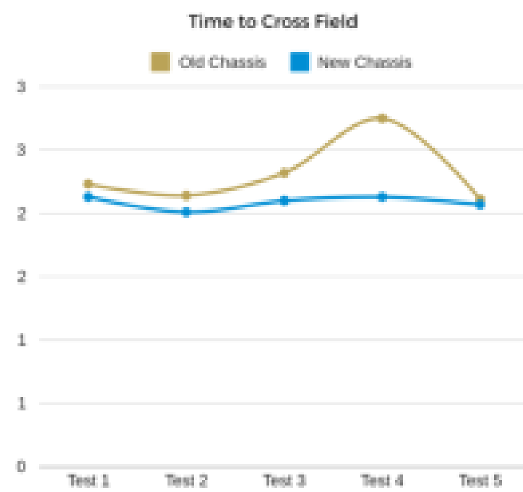


Figure 19: Testing data comparison

As shown in the data, **the new chassis is both faster and more consistent than the old chassis**, proving that the redesign was beneficial to the robot as a whole. In our next entry, we will conduct similar testing with the 15″ chassis.

Testing: 15″ Chassis

Date	@November 28, 2023
Category	15" Robot BLRS2 Skills
Authors	J Jacob Zawacki D Dominic H
Design Cycle	Design Cycle 1

Overall Objectives

In order to ensure that the recently constructed chassis design functions as intended and will perform well in the skills challenge, **we will be testing for various aspects of core functionality. These include:**

1. Speed
    a. The chassis must be able to move quickly.
        i. To test this, **we will measure the time it takes for the robot to cross the field from one side to the other.**
        ii. Ideally, the robot is able to cross the filed in **under 2.5 seconds.**
2. Strength
    a. The chassis must be able to handle weight from additional subsystems, as **well as have the ability to push game objects.**
        i. To test this, we will measure the pull force of the robot in order to indicate drive power.
        ii. Ideally, the robot has a pull force of **over 5 pounds.**

Speed Testing Procedure

Step 1: Place the robot with the rear face contacting the field perimeter.

Step 2: Drive the chassis forward at full speed, starting a timer as the robot begins to move.

Step 3: Stop the timer as the front face of the chassis contacts the various distance milestones we have set. Those are:

- 24 inches (one field tile)
- 72 inches (three field tiles)
- 12 feet (the entire field)

Step 4: Repeat the process for additional measurements.

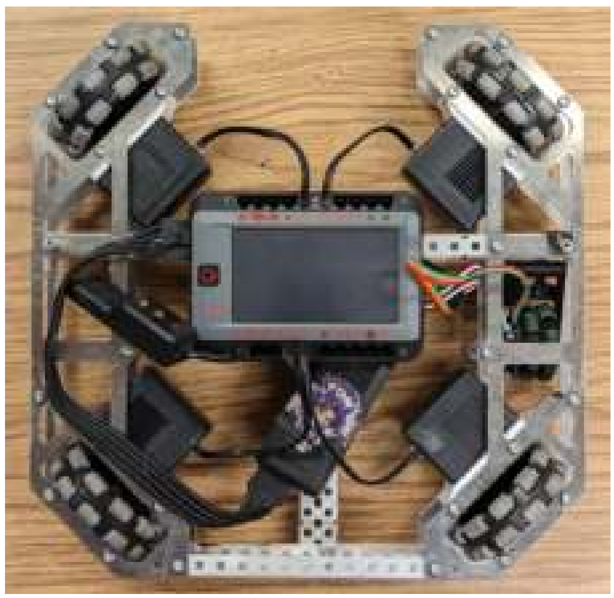


Figure 1: Chassis Speed Testing

Speed Testing Data

Table 1: Cross-Field Speed Testing

Distance Traveled	Time Measured #1	Time Measured #2	Time Measured #3	Average Time (seconds)
12″	0.72 sec	0.83 sec	0.77 sec	0.77 sec
24″	1.82 sec	1.98 sec	1.91 sec	1.90 sec
72′	3.23 sec	3.12 sec	3.18 sec	3.18 sec

Speed Testing Observations

In the speed testing, we noticed the following points of interest:

- With the first three tiles out of six taking over half of the total time to cross the field, we can infer that the **acceleration of this chassis leaves much to be desired.**
- Additionally, the top speed is not as fast as we would like it to be.
- From limited observation, **we believe this to be an issue with traction on the field.** The next strength test should be a good indicator of whether this is the case.

Strength Testing Procedure

Step 1: Attach the robot to the hook of our scale.

Step 2: Hold the scale in place. Drive the robot at full speed away from the scale.

Step 3: Record the weight displayed as the pull force of the chassis.

Step 4: Repeat the process for additional measurements.

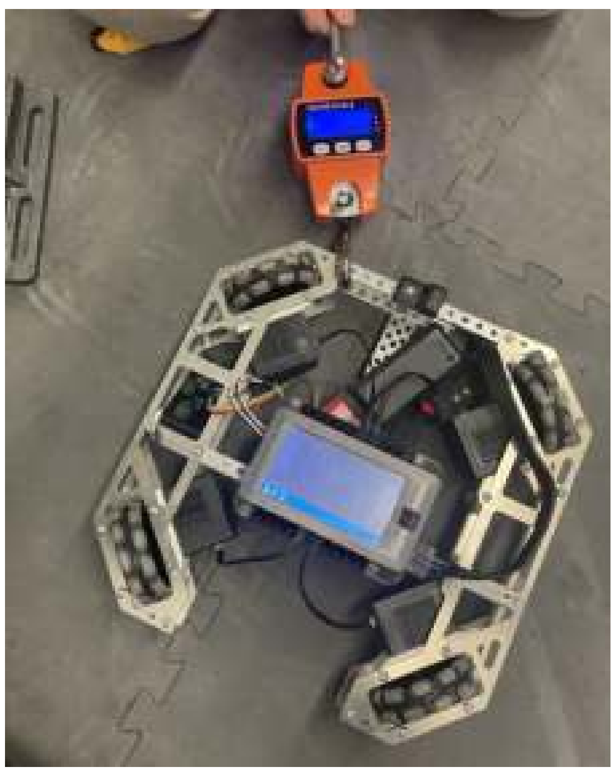


Figure 2: Chassis Strength Testing

Strength Testing Data

Table 2: Strength Testing

Weight Measured #1	Weight Measured #2	Weight Measured #3	Weight Measured #4	Average Weight (pounds)
2.41 lbs	2.52 lbs	2.43 lbs	2.40 lbs	2.44 lbs

Strength Testing Observations

In the strength testing, we noticed the following points of interest:

- The pull force observed by the chassis fell **well under the goal of at least 5 lbs.**
- As we suspected from the speed test, the wheels were slipping on the tile considerably in the speed test, meaning the **robot had little traction was not able to impart the full power into the tests.**

Brainstorming and Selecting Solutions

Identify the Problem and Potential Solutions

Problem: As noted during our testing, the chassis currently **does not have enough traction** to impart the full motor power into movement.

Solution 1: Add more weight to the chassis.

- Increases the ability of the wheels to press into the field tiles, **reducing the likelihood of slippage.**
- Easiest solution to implement, as temporary weight can be added for testing before additional subsystems add the rest.
- **Could increase the likelihood of stalling,** as there would be more friction between the wheels and the ground.

Solution 2: Redesign the wheels to use a more sticky roller material.

- Redesign the omni wheel rollers to **use a stickier plastic material,** so the wheels have more traction to the ground.
- Would involve using a polyurethane blend we use for traction wheels and molding.
- May **reduce the maneuverability of the drive**, as strafe movements may be harder to execute.

Solution 3: Redesign the chassis to use different wheels.

- Switching to a tank chassis with traction wheels would dramatically increase the ability of the chassis to move quickly.
- This would involve a **full subsystem redesign,** which may not be feasible at this point.

Selecting the Solution

Criteria:

1. Ease of Accomplishment: Can we feasibly complete this change quickly (5 for easiest, 1 for most difficult).
2. Planned Effectiveness: How effective would the solution be (5 for most, 1 for least)?
3. Maneuverability: Keeping the maneuverability of the x-drive chassis (5 for most, 1 for least).

Criteria (1-5)	**Solution 1**	**Solution 2**	**Solution 3**
Ease of Accomplishment	5	2	1
Planned Effectiveness	4	3	5
Maneuverability	5	3	4
Total	14	8	10

Due to its ease of implementation and planned effectiveness, we have chosen to **increase the chassis weight** in order to increase overall traction. For testing purposes, we will **repeat the previous tests with different weight amounts** to find optimal weight for additional subsystems.

Revised Objectives

The overall objectives for the revised tests remain the same. These include:

1. Speed

a. The chassis must be able to move quickly.

i. To test this, we will measure the time it takes **for the robot to cross the field from one side to the other.**

ii. Ideally, the robot is **able to cross the field in under 2.5 seconds.**

2. Strength

a. The chassis must be able to handle weight from additional subsystems, as well as have the ability to push game objects.

i. To test this, we will **measure the pull force of the robot in order to indicate drive power.**

ii. Ideally, the robot has a **pull force of over 5 pounds.**

3. Weight

a. In order to determine the optimal weight of future subsystems, **we will be adding different quantities of weight to the chassis when conducting the tests**.

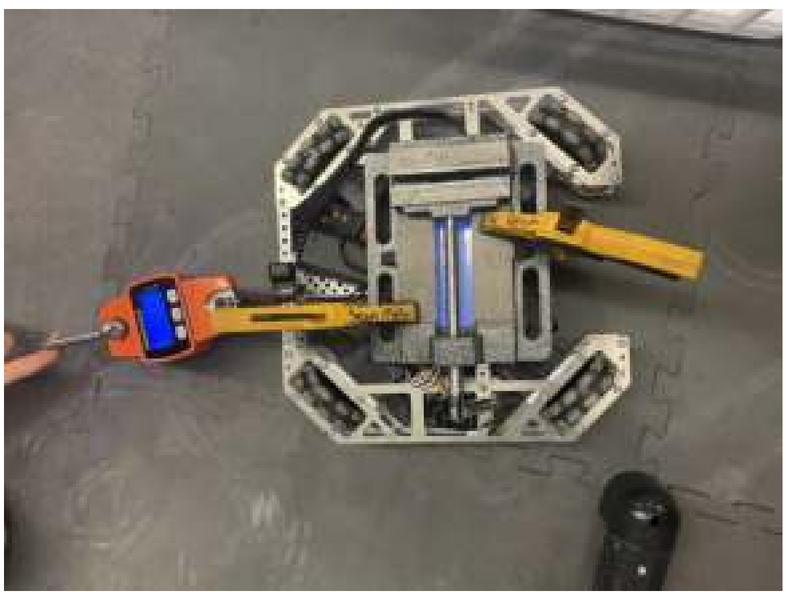


Figure 3: Chassis with additional weight added.

Revised Speed Testing Procedure

Step 1: Place the robot with the rear face contacting the field perimeter.

Step 2: Drive the chassis forward at full speed. Start a timer as the robot begins to move.

Step 3: Stop the timer as the front face of the chassis contacts the opposite end of the field perimeter.

Step 4: Add weight according to the planned weights we have recorded.

Step 5: Repeat the process for additional measurements.

Revised Speed Testing Data

Table 3: Cross-Field Speed Testing

Weight Added	Time Measured #1	Time Measured #2	Time Measured #3	Average Time (seconds)
0 lbs	3.23 sec	3.12 sec	3.18 sec	3.18 sec
2.5 lbs	3.11 sec	3.13 sec	3.07 sec	3.10 sec
5 lbs	2.84 sec	2.91 sec	2.83 sec	2.86 sec
7.5 lbs	2.53 sec	2.44 sec	2.51 sec	2.49 sec
10 lbs	2.97 sec	2.81 sec	2.89 sec	2.89 sec

Revised Speed Testing Observations

In the revised speed testing, we noticed the following points of interest:

- Adding weight to the chassis definitely seemed to help, as indicated by the increase in speed with more weight added.
- The optimal weight added seems to be 7.5 pounds, as further additions to weight decreased the speed due to stalling.
- **With 7.5 pounds of added weight, we meet the initial goal of crossing the field in under 2.5 seconds.**

Revised Strength Testing Procedure

Step 1: Attach the robot to the hook of our scale.

Step 2: Hold the scale in place. Drive the robot at full speed away from the scale.

Step 3: Record the weight displayed as the pull force of the chassis.

Step 4: Add additional weight to the robot, in accordance with the recorded amounts.

Step 5: Repeat the process for additional measurements.

Revised Strength Testing Data

Table 4: Strength Testing

Additional Weight Added	Weight Measured #1	Weight Measured #2	Weight Measured #3	Average Weight (pounds)
0 lbs	2.52 lbs	2.43 lbs	2.40 lbs	2.44 lbs
2.5 lbs	3.41 lbs	3.45 lbs	3.33 lbs	3.39 lbs
5 lbs	5.77 lbs	5.72 lbs	5.80 lbs	5.76 lbs
7.5 lbs	6.11 lbs	6.08 lbs	6.14 lbs	6.11 lbs
10 lbs	5.32 lbs	5.27 lbs	5.28 lbs	5.29 lbs

Revised Strength Testing Observations

In the strength testing, we noticed the following points of interest:

- As reflected in the revised speed test results, adding additional weight helped increase the pull force of the chassis.
- **Additionally, the same weight of 7.5 pounds added is ideal in this testing set,** as pull force decreased with any more weight.
- At this threshold, we are able to exceed the set goal of 6 pounds of pull force.

Conclusions

- In order to maintain optimal speed and strength, **the chassis must have an additional 7.5 pounds of weight added in additional subsystems.**
- With the additional weight, the chassis takes an average of 2.49 seconds to cross the field and has an average pull force of 6.11 pounds.
- With this is mind, we can continue on to completing additional subsystems.

Task and Timeline Update: December

Date	@December 4, 2023
Category	15" Robot 24" Robot BLRS2 Skills
Authors	Conner Siebert Jacob Zawacki
Design Cycle	Design Cycle 1

November Recap

While in October our team planned to complete the CAD for both robots, **our progress was delayed from the initial timeline.** This was caused by several factors, including:

- Max, the 24″ robot lead designer, spent lots of time preparing for the VRC blended competition, where we hosted over 50 MS and HS teams on Purdue's campus
- Dominic, the 15″ robot lead designer, was swamped with a hard semester of classes
- Overall, **manpower was shifted from the skills team to the competition robot team** after the decision to not compete with the skills robots until the World Championships was made

24″ Robot

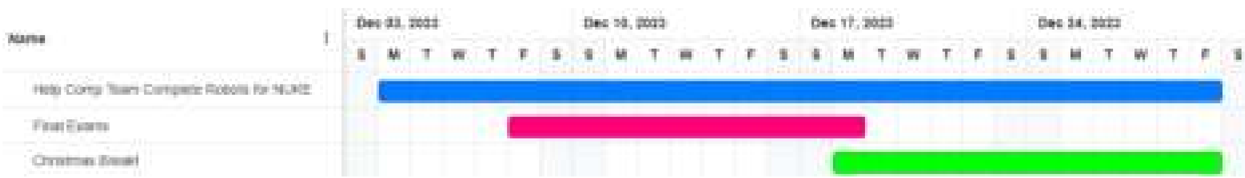
- The chassis redesign was completed by the end of the month, 2 weeks behind schedule
- Several discussions involving the catching mechanism were conducted, to lead our team to **decide on a final mechanism into the goal to be made in December.**

15″ Robot

- The chassis CAD was completed.
- The chassis was machined, built, and tested. **It proved to be viable, once an additional 7.5lbs were added for testing proposes.**

December Task and Timeline

With finals and a 3 week winter break occurring throughout December, the skills team has decided to have a light month. **A majority of the manpower will be shifted towards completing the competition robots.** This will allow the skills team to both continue to contribute to the team and still have time to prepare for final exams and enjoy a well deserved Christmas break.



Task and Timeline Update: January

Date	@January 2, 2024
Category	15" Robot 24" Robot BLRS2 Skills
Authors	Ⓒ Conner Siebert Ⓜ Max

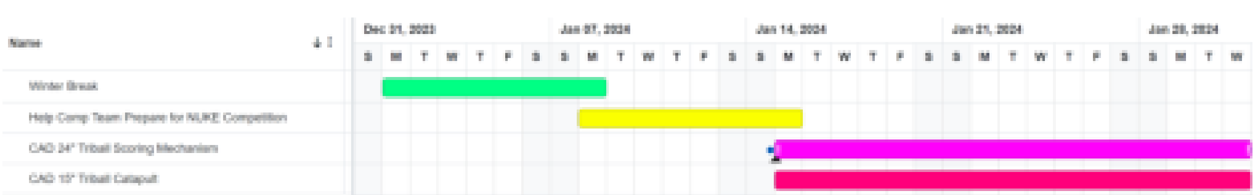
December Recap

As stated in the December task and timeline update, little progress was made from the skills team in December

- Team members were successful in passing final exams
- A much needed break was found at home by members
- The team is excited for a busy January

January Task and Timeline

- As the competition team is attending NUKE on January 14th, the two competition team robots will be completing our teams first skills runs of the season.
    - To assist, **the man power from the skills team will be shifted towards assisting in finishing the comp bots along with programming and driver skills runs.**
- After NUKE, the skills team will resume design work occurring last year
    - The 24″ team will work on designing the Triball catching and scoring mechanisms
    - The 15″ team will work on CAD for the Triball catapult



February Update: Development on Pause

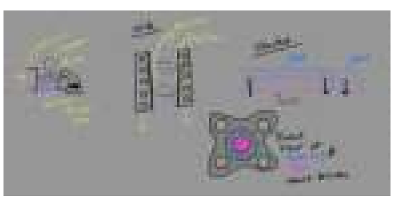
Date	@February 10, 2024
Category	15" Robot 24" Robot BLRS2 Skills
Authors	Ⓙ Jacob ZawackiⓂ Max
Design Cycle	Design Cycle 1

January Recap

For the month of January, our skills robots were put on hold. **Our team competed at 2 competitions, hosted a 60 team VRC Blended competition, and volunteered at 2 other VRC competitions meaning we were all very busy.** Our time during the week was spent making sure that the competition robots would be ready to perform at the standard we expected and needed to qualify for the world championship.

Because of these constraints, very little work was able to be completed on the robots.

Skills 24″ puncher design



F1: Rack and Pinion Sketches

We spent a very small amount of time figuring out what a puncher design could look like on the 24″ robot. **Above are sketches of the bearing setup for a rack and pinion.**

F2: Linear Rail Carriage

We **paused CAD** of linear rail holders for a smooth puncher in favor of our competition bot development.

F3: Linear Rails

F4: Linear Rails

January 30th Game Manual Update

<SG 11> Updates

The updates to <SG11> changed how match loading could be done by adapting the intent. This sparked several Q&A posts about legal vs illegal match loading, **which ended with a change in how we will have to match load for skills.**

A basic summary is that new rulings are intended to ‘make the robot play the game, not humans’. This sparked the GDC to give the following thought experiment as a ruling on how match loads can be legally placed.

- Would the Triball still have been retrieved if it was placed a few seconds before the Robot drove over?
- Would the Robot function the same if the Triballs were being introduced by an uninformed observer or field-side volunteer? (e.g. the only thing they know is "put it down somewhere in this corner")
- If the Match Load Bar were a white tape line instead of a PVC pipe, and no Robot was present, would the Triball have bounced/rolled out of the Match Load Zone? (i.e. an extreme version of point 1-a in SG6)
- If Triballs were cubes, would the Match Load have moved after it was introduced?

Our current skills strategy violates this plan, as we had intended to place the Triballs with 1 tip down into an enclosed ring.

> Because of these changes, **we’ve decided to place a hold on the skills robots.** We may choose to come back to these robot’s developments as more Q&A’s are posted about match loading legality, but until there is a firm ruling made, we don’t want **allocate our resources** towards robots that will likely need to be redesigned as the GDC continues to change the game.

We will revisit these robots in March.

PURDUE SIGBOTS

BLRS2 TEAM AND PROJECT MANAGEMENT

BLRS2 9/24 General Meeting

Date	@September 24, 2023
Category	BLRS2 General Meeting
Authors	Ⓙ Jacob Zawacki Ⓜ Mihir Laud Ⓢ Sean MacDonald Ⓐ Alex Lam
Design Cycle	Design Cycle 1

General Team

Progress Summary:

- Introduced new Notion features to implement for this season
  - Project management and progression with Task Board

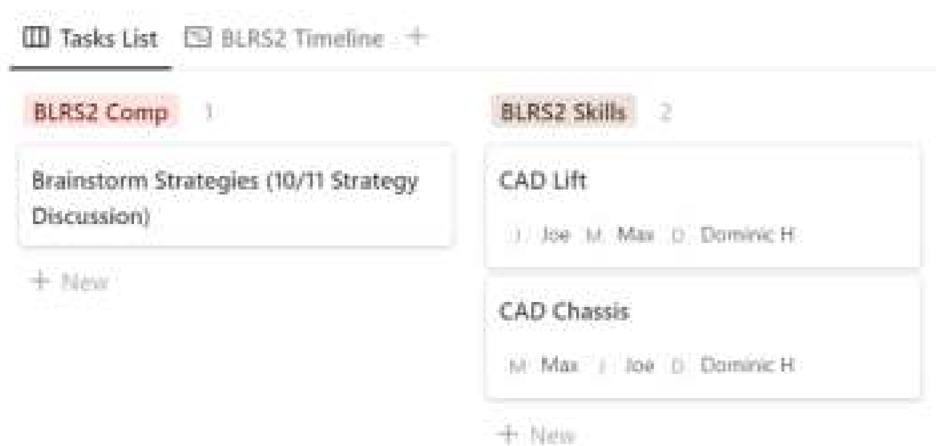


Figure 1: Task board utilized to organize projects within respective teams

- Organize deadlines with Gantt Charts

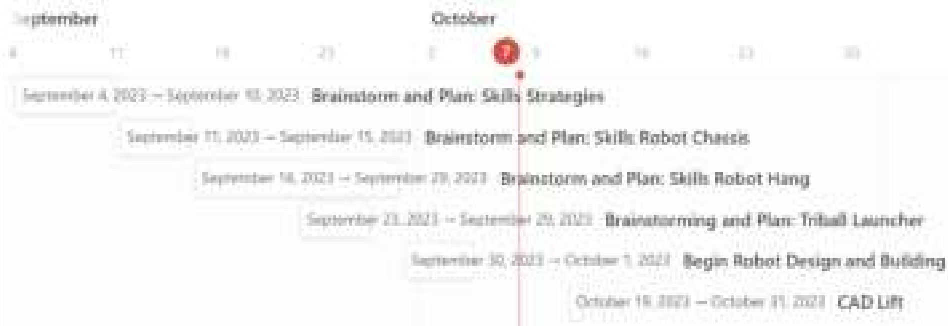


Figure 2: Organize team deadlines/schedule through the use of a Gantt Chart

- Identified skills team as an early season priority
  - Possibility of World Championship qualification through skills
    - Top 5 ranked teams in the world on 12/31 earn a qualification
  - Deadline of setting a skills score at our **12/2 Blended Tournament**

Meeting Accomplishments:

- Introduced team structure
  - 4 robots: 2 for Skills, 2 for Competition
  - Main teams: Skills & Comp Team
    - Each will have an additional independent meeting outside of general meetings
    - Members have the option to work on either team, or both
      - Members must attend independent meetings to stay in communication
  - General weekly meetings
    - Update the overall team on current task progress
    - Outline future plans and goals
    - Open floor discussion and questions from team members
  - Independent team meetings
    - Used to discuss the direction of strategy, robot design, programming, etc.
    - Establish milestones & deadlines
    - Delegate tasks between members of the subgroup
- Established **Sunday at 6 PM EST** as reoccurring general meeting time

Competition Robots

Progress Summary:

- Skills Early-Season Priority
    - Skills subgroup will be developing their robots sooner than Competition subgroup
    - This is due to the 12/2 deadline for setting a skills score, as mentioned
    - This will give competition team more time to develop strategies for match play

Meeting Accomplishments:

- Overall Game Strategy Discussion
    - Catcher/Shooter Strategy
        - Would it be viable to have one robot specialize each in catching and shooting TriBalls?
        - Overall consensus was no, as there is a **high risk of one robot being trapped by the opposing team** causing the other robot to be unable to score
        - Walls could be a very strong counter to this strategy, worth looking into as a subsystem idea
    - “Game Break” Ideation
        - Focus on building strong robots initially
        - As we gain more experience in the competition, developing ideas to differentiate our robots for a competitive edge will help going into Worlds
    - Skills Testing
        - As skills will be starting with development sooner than the competition robots, we can base ideas off of testing for those robots
        - Place more emphasis on competition durability and racing other teams for TriBalls on the field, as opposed to skills
    - Strategy utilized should not rely on winning autonomous
        - Hard to guarantee an autonomous win, planning strategies to win in spite of an autonomous loss will be better in the long run

Weekly Goals and To-Do:

- Brainstorm match play strategies
- Develop basic robot concepts

Name	Task	Target Completion	Priority
All Members	Develop match strategies	10/1/2023	High

Skills Robots

Progress Summary:

- Discussed what the skills strategy for early season will be
    - **Current deadline is December 2nd to post a high skills score at Purdue's Blended VRC event**
    - Looking to make bots as light and fast as possible to maximize score in a short amount of time
    - Two 15" bots with almost identical capabilities
    - Begin with scoring all match loads, then use wing mechanisms to push TriBalls into offensive zone and goal if possible
    - Looking to elevate as high as possible, minimizing weight will help this as well
    - Things to keep in mind:
        - Match loads should be launched over the barrier but not so far that they leave the field or end up in opposing match load zone
        - Programming skills will need to be able to consistently push TriBalls to maximize score
        - There are 22 match loads, so cycle time needs to be kept low to get them all on the field

Meeting Accomplishments:

- Brainstormed potential chassis ideas and sketched out preliminary plans
  - One idea suggested was a power take-off (PTO) drive that would use 8 motors for driving, but could split off power from 6 of those motors to drive a catapult or winch mechanism

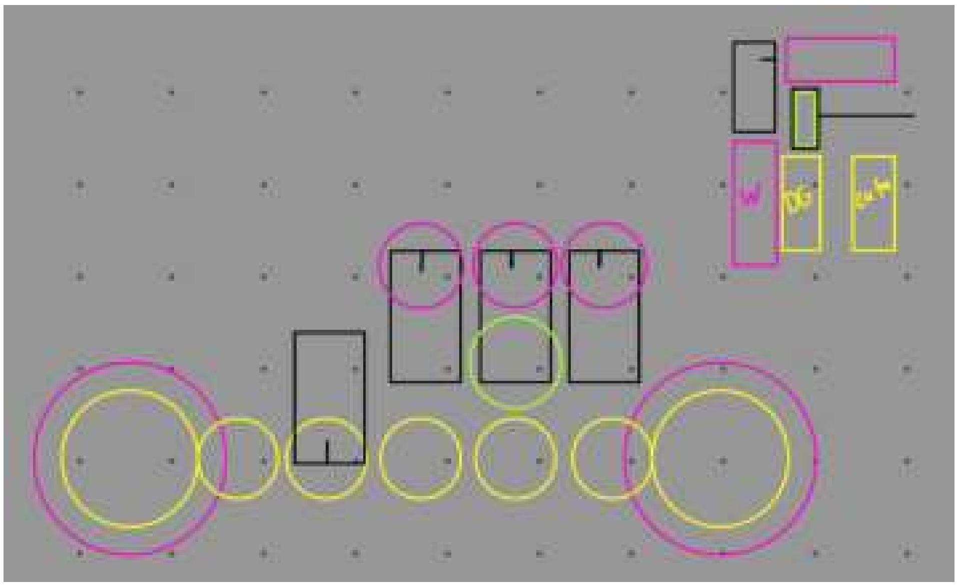


Figure 3: Basic layout of a PTO drive using 8 motors for driving and 6 motors for catapult or winch mechanism

  - Idea stems from the strategy of launching match loads while stationary, so power will not be needed for driving and shooting simultaneously, meaning the motors can be split
  - Pros:
    - Better utilization of motors during match
    - Compact design leaves more room for other mechanisms
  - Cons:
    - Chassis with 8 motors can be very heavy, limiting speed
    - Complexity adds time to the build process and could lead to issues with maintainability down the road

- Brainstormed elevation mechanisms for endgame portion of skills
    - One strategy is using a lift with a winch to pull the robot up to elevation height

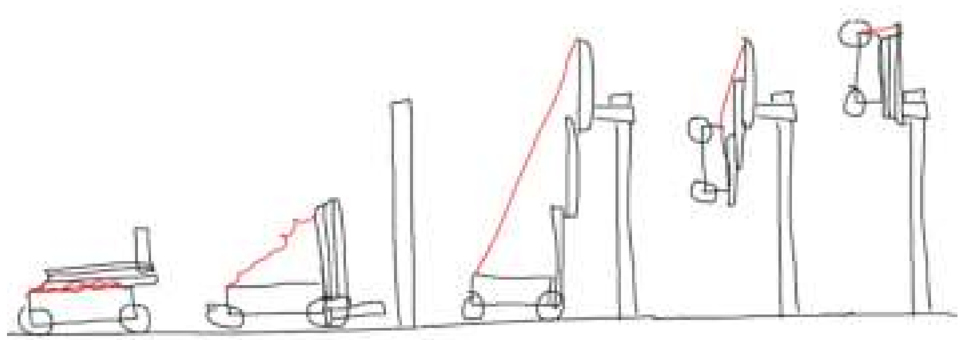


Figure 4: Sketch of how a winched lift might work to elevate the robot. The red line denotes the tether attaching the top of the lift post to the robot chassis

    - This would allow us to pull up the bot as high as possible and score more points in the endgame
    - Potential concerns:
        - Winch could be slow to elevate, and we have a very limited amount of time
        - If we are not lined up straight before elevation, the lift may have unnecessary stresses that could damage the mechanism
        - This strategy requires the bot to be very light

Weekly Goals and To-Do:

- Finalize mechanism decisions and begin to CAD the robot

Name	Task	Target Completion	Priority
All Members	Finalize mechanism design decisions	9/29/2023	High
Max Johnson, Dominic Holifield	Begin robot CAD	10/1/2023	Medium

Strategy

Progress Summary:

- Established general documentation format
  - Use of Notion to coordinate Tasks, Timelines, and Documentation
  - Every member of the club has access to the SIGBots Notion
    - Helps to keep all members in communication and aware of progress
    - Easy integration of digital features
      - Gantt Charts
      - To-Do Lists
      - CAD Models
- Introductory entries have been written
  - Team Organization and Structure
  - Our Engineering Design Process
  - Game Analysis
  - Beginning Skills team analysis

Meeting Accomplishments:

- Established general subgroup roles and responsibilities
  - All members contributing to documentation entries
  - Alex and Jacob specializing in entry review and formatting
  - Kathleen, Krish, and Conner specializing in entry documentation
  - Jacob coordinating team interview

Weekly Goals and To-Do:

- Continue recording brainstorming entries for Skills subgroup

Name	Task	Target Completion	Priority
All members	Continue documenting skills brainstorming	As needed	Medium

BLRS2 10/1 General Meeting

Date	@October 1, 2023
Category	BLRS2 General Meeting
Authors	J Jacob Zawacki M Mihir Laud C Conner Siebert
Design Cycle	Design Cycle 1

Competition Robots

Progress Summary:

- Last week, the competition team started discussing potential robot ideas
  - Need to finalize game strategy before CAD and design can begin
  - There are concerns about what is needed long term, meaning that it is hard to commit to one particular strategy without heavy consideration

Meeting Accomplishments:

- Discussed the possibility of two identical robots
  - Would make testing efficient, as results could be applied across both robots
  - Could make our strategy more versatile, with both robots able to complete every task
  - Include every mechanism on both robots, with only minor variations, so as to maintain flexibility
    - Key subsystems: Drive, Launcher, Intake, Wall, Hanging Mechanism
- Next BLRS2 Team meeting: 10/11/2023
  - Every member will develop and present a strategy to present to the team
  - Ideas will be discussed, considered, and merged until we have at least three overall strategies to discuss and choose from
  - By the end of the meeting, we will have established our overall competition strategy

Weekly Goals and To-Do:

- Brainstorm potential robot ideas and strategies, to make an informed decision at our next meeting on 10/11.

Name	Task	Target Completion	Priority
All Team Members	Brainstorm match strategies and ideas/requirements for the robots	10/11/2023	High

Skills Robots

Progress Summary:

- Decisions have been made about various subsystem specifics, and have been recorded with more detail in specific Brainstorming and Decision Matrix entries
    - Drivetrain
        - Main considerations were motor count (8 vs 6 motor drive) and whether to use a Power-Take-Off (PTO) Mechanism for additional mechanisms
            - 8 motor drive potentially unnecessary for skills
            - 6 motors should be sufficient with 4 motors being PTO'd to the catapult and winch
            - Could add another motor directly to the catapult/winch system
            - **Decided to go with 6 motor drive, PTO 4 motors to catapult and winch**
    - Overall Gear Ratios
        - 600 rpm motor carts geared down to 300 rpm for the drive
        - 200 rpm winch and 50/75 rpm for catapult
    - Catapult
        - Using 4-bar choo-choo linkage in place of slip gear
            - Should have considerably faster match loading potential
        - Sensor should not be located at the top of the mechanism to prevent bouncing and inaccuracy

- Hanging Mechanism
  - Manufacture an incredibly light linear slide mechanism
  - Sits horizontally on bot, gets flipped vertically, and telescopes out due to tension
  - Claw is attached to top of slide and clamps onto elevation post
  - Winch pulls the robot up to the top
  - Potential concern
    - The linear slide will have to retract as the robot is lifted, meaning the tension of the lift will be fought on the way up

Meeting Accomplishments:

- Task and Timeline Update
  - Described an overall recap of September
  - Outlined the month of October using goals, deadlines, and a Gantt Chart

Weekly Goals and To-Do:

- Continue brainstorming for the Hang Mechanism and PTO Design
- Discuss and begin chassis CAD Design

Name	Task	Target Completion	Priority
All members	Brainstorm mechanisms	10/10/2023	Medium
Max and Dominic	Chassis CAD	10/10/2023	Medium

Strategy

Progress Summary:

- Skills robot entries have continued to progress and have been documented as such

Meeting Accomplishments:

- There have been some inconsistencies in our General Meeting entries, which we plan on addressing with an updated layout
    - Priority status added to the to-do charts
    - More clear divisions between general, competition, and skills notes
- Competition Robots will begin brainstorming soon
    - Will need to document as progress occurs
- Skills Robots will be progressing as they have been

Weekly Goals and To-Do:

- Make the necessary adjustments to the General Meeting entries
- Begin the early documentation for the Competition robots
- Continue documenting the Skills robots

Name	Task	Target Completion	Priority
Jacob and Alex	Refine the format for General Meeting entries	10/10/2023	High
Kathleen	Begin Competition Robot entries	10/21/2023	Low
Conner and Noah	Continue Skills Robot entries	10/5/2023	Medium

BLRS2 10/15 General Meeting

Date	@October 15, 2023
Category	BLRS2 General Meeting
Authors	Kathleen Lowe, Jacob Zawacki, Alex Lam
Design Cycle	Design Cycle 1

General Team

Progress Summary:

- Met to discuss the progress of Competition Robots, with some light updates on the Skills Robots CAD progress

Meeting Accomplishments:

- Established a beginning timeline for the Competition Robots
- Discussed goals for the mid-season

Weekly Goals and To-Do:

- Agree on smaller design details to prepare for CAD

Name	Task	Target Completion	Priority
Max, Dominic	Continue Skills CAD	10/31/2023	High
All Competition Members	Decide Competition specifics	10/18/2023	High

Competition Robots

Progress Summary:

- Established central design decision
    - One-sided intake for fielding Triballs
    - One-sided launcher on the opposite end
        - Variable speed for different shooting circumstances

- Mid-tier elevation
- See Planning: Match Strategies for more information!

Meeting Accomplishments:

- Begin Timeline

  - Use this upcoming week to finalize specific robot details
    - Meet Wednesday (10/18) to finalize minor specifics for robot subsystems
  - Begin dividing CAD tasks between members
  - Move into building as CAD progresses
    - ~5 weeks allotted for building
- Goals for NUKE Competition
  - Win Excellence Award
  - Competition robots do not need to be perfectly tuned
  - Have autonomous tuned and prepared to obtain Win Points
  - Put up high skills score at competition
  - If all goes well, does not necessarily need to go to any other competitions
    - Take time leading to World's tune autonomous, drive practice, minor robot adjustments

Weekly Goals and To-Do:

- Decide upon the specifics for the Competition robots, so that CAD and design work can begin

Name	Task	Target Completion	Priority
All Team Members	Flesh out design details	During Wednesday Mechanics Meeting	High
Connor, Brandon	Test Flywheel configurations	Wednesday	Medium
Sean, Joe, Evan, Brandon, Connor	Begin Chassis CAD	TBD	High

Skills Robots

Progress Summary:

- While the CAD is a few days behind schedule, the Chassis CAD has been completed barring any minor modifications

Meeting Accomplishments:

- Our progress goal has been accomplished so far, so work on the remaining subsystems can begin
- We will potentially recruit more members from the Competition team to help spread out the workload

Weekly Goals and To-Do:

- Begin the CAD for the Lift and Catapult subsystems

Name	Task	Target Completion	Priority
Dominic, Max	CAD Lift	10/31/2023	Medium
Dominic, Max	CAD Catapult	10/31/2023	Medium

Strategy

Progress Summary:

- The Competition robot meeting on October 11 has been documented, and needs to be formatted accordingly
- Competition
  - The Design Brief for Design Cycle 1 and the Match Strategies entry have been created
- Skills
  - Testing data has been recorded and documented as CAD progress continues

Meeting Accomplishments:

- Tasks have been allocated for in-progress entries

Weekly Goals and To-Do:

- Continue to keep all entries up to date

Name	Task	Target Completion	Priority
Kathleen	Reformat the 10/11 Meeting Entry	10/21/2023	High
Jacob	Finalize General Meeting Entries	10/19/2023	Medium
Conner, Alex, Noah	Continue Robot Entries	10/21/2023	Medium
Jacob	Create Design Entry Template	10/20/2023	High

BLRS2 10/22 General Meeting

Date	@October 22, 2023
Category	BLRS2 General Meeting
Authors	Ⓒ Conner Siebert Ⓚ Kathleen Lowe
Design Cycle	Design Cycle 1

General Team

Progress Summary:

- Custom cast omnis are being testing
    - Brandon is on V1
- Connor should look into printing a test omni wheel since the club does not have Vex omnis
- There will be no general meeting next week (10/29) due to several members having prior commitments.

Weekly Goals and To-Do:

Name	Task	Target Completion
Brandon, Connor	Finish casting custom omnis and test	10/29/23

Competition Robots

Progress Summary:

- Chassis CAD has begun
    - ~50% completed
    - Outer Plate finished
    - Inner support in progress

Meeting Accomplishments:

- Discuss several aspects of the design
  - Weighted flywheel
    - Begin with vex sprocket
  - Pass through rollers
    - Start with rubber band rollers
    - 11W powering both rollers chained together
    - Faster is better, as speed feed into the flywheel
  - Endgame hang
    - Arm to grab vertical pole, park on short barrier
    - Look into quick, easy pistons if time allows
    - Look into the flex wheel pole climb elevation
  - Wall
    - Joe to begin work, would be nice to have for January 14th competition
    - Piston flip out
- Assigned CAD task for the next week
  - At this point in the timeline, it is critical the team continues great strides on the CAD

Weekly Goals and To-Do:

Name	Task	Target Completion
Brandon, Sean	Finish Inside Chassis Plate	10/29/2023
Joe	CAD Flywheel	10/29/2023
Joe	Begin looking into wall	
All	Brainstorm early season endgame hangs	

Skills Robots

Progress Summary:

- The skills team has had a slow week, due multiple exams
    - The CAD had not been worked on for the last week due to this

Meeting Accomplishments:

- Conner expressed concerns for the timeline of having both robots ready to compete on December 2nd
    - Max and Dominic believe the bot is still on time, and promised to continue to CAD the triball launcher and hanging mechanism

Weekly Goals and To-Do:

Name	Task	Target Completion
Dominic, Max	Continue to CAD tribal launcher and hanging mechanism	10/31/2023

Strategy

Progress Summary:

- Kathleen finished the competition team strategy meeting notes!
- Jacob moved several meeting notes to ready for review

Meeting Accomplishments:

- Conner determined the entries for each member to work on for the next week
    - The notebook is doing very well timeline wise, as each robot is staying up to date and entries are being completed on time.

Weekly Goals and To-Do:

Name	Task	Target Completion
Conner	Determine upcoming entries for each set of robots	10/19/23
Jacob	Complete the flywheel prototyping and testing entry	10/31/23
Alex	Complete the winch prototyping and testing entry	10/31/23
Kathleen, Noah	Assist with future entries to be determined by Conner	TBD

BLRS2 11/5 General Meeting

Date	@November 5, 2023
Category	BLRS2 General Meeting
Authors	K Kathleen Lowe C Conner Siebert
Design Cycle	Design Cycle 1

General Team

Progress Summary:

- Parts and materials ordered
- There will be no general meeting next week (11/12) due to several members having prior commitments.
    - It is important for all members to continue communications throughout the next two weeks through discord and in-person conversations

Meeting Accomplishments:

- Discussed online challenges
- Revisited timeline for CAD and building

Weekly Goals and To-Do:

- Divide up work on online challenges
    - Community, Photography, Poster Design, Girl Powered
- Discuss contributions to the Community challenge
    - Focusing on volunteering and 1 on 1 help

Name	Task	Target Completion
Max	investigate / plan Community online challenge	next week's meeting, 11/12
Everyone	consider which online challenges you'd like to participate in	next week's meeting, 11/12
Kathleen	investigate / plan out Girl Powered online challenge	next week's meeting, 11/12

Competition Robots

Progress Summary:

- Brandon and Sean worked on the chassis CAD
- Conner tuned his 3D printer for tolerances on the chassis piece mounts

Meeting Accomplishments:

- The timeline was discussed
    - Things are going okay, could be slightly faster
    - The goal is get the metal this week so the chassis can be machined
- Sean discussed possibility taking the robots home over Christmas break
    - This would allow for all mechanics work to be completed, as there will only be 1 week on campus following the break before comp team's first competition at NUKE
    - Connor also lives close to Sean, and would be willing to work with him over the break

Weekly Goals and To-Do:

Name	Task	Target Completion
Joe	Machine intake arms, chassis pieces	11/8
Brandon, Sean	Complete chassis CAD	11/8
Conner, Connor	3D print chassis mounts, gears	11/12
Brandon, Conner, Joe, Brandon	Build test flywheel	11/12

Skills Robots

Progress Summary:

- Skills 24″
  - Chassis CAD almost completed
  - Max has been working hard, also working around exams
- Skills 15″
  - Dominic is nearing the completion of the chassis design
    - He’s going to preform finite element analysis to optimize the weight for the strength

Meeting Accomplishments:

- Conner echoed the concerns with the comp team timeline to the skills team
  - With most of the team leaving campus for Thanksgiving break, there will only be three weeks left on campus for the team
  - Pushing the CAD to be finished is a key priority
  - Manufacturing of chassis pieces should be starting soon

Weekly Goals and To-Do:

Name	Task	Target Completion
Max	Finish 24″ Chassis CAD	11/9
Dom	Finish 15″ Chassis CAD	11/9
Dominic, Conner	Test Cata for the robot	11/12
Max	Machine 24″ Chassis Polycarbonate pieces	11/12
Dominic	CAD 15″ Launcher	11/15
Max, Conner, Thomas	Build 24″ Chassis	11/15

Strategy

Progress Summary:

- There was no strategy subteam meeting this week.

Meeting Accomplishments:

- Entries assigned last week are to be continued this week.
- After the mechanics teams begin building in the coming weeks, strategy members should be read to assist with the documentation

BLRS2 11/26 General Meeting

Date	@November 26, 2023
Category	BLRS2 General Meeting
Authors	J Jacob Zawacki C Conner Siebert
Design Cycle	Design Cycle 1

General Team

Progress Summary:

- Not planning to run skills on 12/2
  - More time for chassis improvements
  - Avoid conflict of interest, as Max is our Event Partner
- Skills at Nuke Competition 1/20
  - Potential to not bring the skills robots to Nuke
  - Give more time to develop high-quality robots for Worlds

Meeting Accomplishments:

- Timeline
  - Tentative decision to avoid using the skills robots until Worlds

Weekly Goals and To-Do:

- Maintain progress on skills robot design
- Continue parts for the competition robots

Name	Task	Target Completion
Max, Dominic, Thomas	Skills robot CAD	Early second semester
Joe, Brandon	Machine competition parts	12/17/2023

Competition Robots

Progress Summary:

- Machining parts and aluminum has begun for both competition robots
- CAD on further mechanisms has continued and should be finished within three weeks

Meeting Accomplishments:

- Aiming to have both robots completely built by the end of Christmas break
- Develop a plan for programming in the week we have at school before Nuke (1/3-1/10)
    - Have that ready by the end of the week

Weekly Goals and To-Do:

- Finish machining parts for both robots
- Complete robot construction by the end of Christmas break
- Develop a plan for programming

Name	Task	Target Completion
Programming Plan	Conner, Stephen, Mihir	12/3/2023
Machining Parts	Joe, Brandon	12/17/2023
Complete robot build	Sean, Conner, Stephen	1/6/2024

Skills Robots

Progress Summary:

- Chassis completed on both robots
    - 24”
        - Plastic has proven to not meet the standards of robust build quality
    - 15”
        - Frame is slightly warped
        - Need different wheels for better traction

Meeting Accomplishments:

- 24″
  - Decision made to rebuild the chassis frame using aluminum
    - 15″ aluminum chassis has proven to be very durable
- 15″
  - Continue progress on the catapult and endgame
  - Fix the wheels for more traction

Weekly Goals and To-Do:

- 24″
  - Redesign parts of the chassis to use an aluminum frame
  - Aim to accomplish this within one week
- 15″
  - Fix the wheels by next week

Name	Task	Target Completion
Max, Thomas	Redesign 24″ Chassis	12/3/2023
Dominic	Fix 15″ wheel traction	12/3/2023

BLRS2 12/3 General Meeting

Date	@December 3, 2023
Category	BLRS2 General Meeting
Authors	Ⓒ Conner Siebert Ⓐ Alex Lam
Design Cycle	Design Cycle 1

General Team

Progress Summary:

- We successfully hosted the Blended VRC competition
    - 53 Teams attended
    - A full slate of awards were given, lead by our strategy subteam and Jacob as JA
    - Event was highly praised by coaches, parents, and students for being well organized and on-time
    - Max is an amazing Event Partner
    - Conner and Stephen lead as head refs, running over 100 qualification matches
- Software front update from Mihir
- New 3D printer filament sponsorship
    - Polymaker is a proud sponsor of Purdue SIGBots!
    - They will be donating PLA filament to the club

Meeting Accomplishments:

- Planning for NUKE Competition on 1/14
    - On Saturday the 13th, our team plans to split squad volunteering at both NUKE's high school competition and Cornerstone 2 tournament in Martinsville, IN
    - Those volunteering will then meet in Kentucky Saturday night, and compete on Sunday

Weekly Goals and To-Do:

Name	Task	Target Completion
Jacob	Fill out form to coordinate team for NUKE competition	12/4/2023
All members	Complete the form above	12/10/2023

Competition Robots

Progress Summary:

- CAD is mostly finished
  - Screws needed added
  - Flywheel needs banded
  - Cross bracing
- Inner plates are printed by Taylor
- Things need 3D printed
- Part of the first robot chassis was assembled
  - Waiting on re-machine of chassis plate... ME shop messed up, half of the screw holes were 4mm lower than the rest of the piece was machined

Meeting Accomplishments:

- NUKE timeline updated
  - The first robot will finish being built by the end of the semester (2 weeks from now)
  - Sean, on his return from his internship in Huston, will stop at Purdue at the beginning of Christmas break to pick up a build of materials (BOM) for the second robot
  - Once the spring semester begins, there will be a week to program and drive practice the robots
- Metal for bot 2 needs ordered
  - After the mechanical engineering shop destroyed the remainder of the original sheet, metal for the second competition robot will need ordered

Weekly Goals and To-Do:

Name	Task	Target Completion
Chris (Club mechanics lead)	Order aluminum	12/4/2023
Sean	Pick up parts for 2nd robot	12/15/2023
Brandon, Conner, Stephen	Finish building 1st robot	12/17/2023

Skills Robots

Progress Summary:

- Conner and Max went through a design process decision, decided to re-design and re-build the 24″ chassis
  - The current chassis has human errors in machining
  - The of maintenance the current chassis could lead to future issues
  - The chassis flexes when running to items, such as barriers and the walls
- The 15″ robot has seen no progress this week, due to Dominic having exams and finals
  - To help with this, Stephen has returned to SIGBots after a break to help

Meeting Accomplishments:

- Discussion programming, Mihir and Dominic should be able to program the 24″ and 15″ respectively

Weekly Goals and To-Do:

Name	Task	Target Completion
Max, Chris (Club mechanics lead)	Metal needs ordered to machine the new 24″ chassis	12/5/2023
Dominic, Stephen	Dominic should continue to CAD the catapult	1/6/2024

BLRS2 1/8 General Meeting

Date	@January 8, 2024
Category	BLRS2 General Meeting
Authors	Alex Lam Stephen Hohnholt noah
Design Cycle	Design Cycle 1

General Team

Progress Summary:

- Form sent out and completed by the team for Cornerstone VRC competition volunteering
- Travel set up for Cornerstone VRC competition volunteering & NUKE competition
- Continue preparations for the NUKE competition on Sunday the 14th

Meeting Accomplishments:

- Set meeting for team travel on Wednesday at 6:30 PM
  - Overall club meeting (BLRS & BLRS2) to discuss travel logistics and expectations
- Reinforced team goal for NUKE competition of winning the Excellence Award

Weekly Goals and To-Do:

- Attend club meeting on Wednesday
- Continue progress on competition robots

Name	Task	Target Completion
All members	Club meeting	1/10/2024
Sean, Brandon, Conner	Finalize competition robots	1/12/2024
All members	Compete at NUKE	1/14/2024

Competition Robots

Progress Summary:

- Robot construction has progressed well over break, only minor additions and testing needed before NUKE for mechanics
- Both robots need an autonomous routine for matches and skills programmed

Meeting Accomplishments:

- Mechanical to-do has been decided:
    - The flywheel launcher mechanism still needs to be designed and built, as we did not have access to necessary parts over break
    - Custom traction wheels still need to be molded for the chassis
- Software subteam will begin to design and program autonomous routines

Weekly Goals and To-Do:

- (what progress needs to be made)

Name	Task	Target Completion
Mihir, Dominic, Stephen	Programming Competition Robots	1/13/2024
Brandon	Create/Mold Traction Wheels	1/9/2024
Conner, Sean	Design and Build Flywheel	1/10/2024

Skills Robots

Progress Summary:

- As priority has been to complete the competition robots for the NUKE competition, little progress has been made for the skills robots
- We plan to observe skills strategies shown at NUKE, to make any potential changes to our strategy while still in the Brainstorming phase

Meeting Accomplishments:

- Discussed putting further skills robot progress on hold while we observe competition strategies from other teams

Weekly Goals and To-Do:

- Conduct scouting at NUKE and online for skills challenge strategies

Name	Task	Target Completion
Dominic and Max	Scout skills strategies	1/29/2024

Strategy

Progress Summary:

- The notebook has been kept up to date well this past semester, with little additional work needed before NUKE

Meeting Accomplishments:

- Review the team interview plan in Preparation for NUKE Competition
    - We plan to closely mirror the rubric in our interview, so as to convey information in a way familiar to the judges
- Plan to compile the notebook for NUKE
    - Combine the individual Notion entry PDF's into one combined notebook

Weekly Goals and To-Do:

- Review interview plan with competition team
- Proofread the notebook before NUKE
- Compile the notebook for NUKE

Name	Task	Target Completion
Alex and Kathleen	Proofread the notebook	1/10/2024
Conner and Jacob	Compile the notebook	1/14/2024
Jacob	Review interview plan	1/12/2024

BLRS2 1/15 General Meeting

Date	@January 8, 2024
Category	BLRS2 General Meeting
Authors	Ⓐ Alex Lam Ⓙ Jacob Zawacki Ⓒ Conner Siebert
Design Cycle	Design Cycle 2

General Team

Progress Summary:

- BLRS2 won the Design Award at NUKE and made it to Quarterfinals
    - Unfortunately, the team did not qualify for the World Championship

Meeting Accomplishments:

- BLRS2 will be attending the MCCC Tournament on 1/27 in the hope of qualifying for Worlds
- Various competition and communication strategies were discussed:
    - Competition strategies will be remarked below
    - We will be attempting to improve our communication by sending informal progress summaries whenever any progress has been made
        - This will allow our team to function more as a fluid, cohesive unit, and ensure tasks are being met on time

Weekly Goals and To-Do:

- Attend MCCC tournament
- Continue to improve communication

Name	Task	Target Completion
All members	Attend MCCC	1/27/2024
All members	Improve communication	Forever

Competition Robots

Progress Summary:

- While the robots worked at NUKE, we noticed several strategic failures that we will need to correct going forward:
  - Passthrough ended up not being used or necessary, and will likely not be considered in the future
  - Using fully-custom machined parts, while educational, did not help to create easily tunable robots
  - The large intake to intake from the match load zone was difficult to use and maneuver

Meeting Accomplishments:

- The Brainstorming process for our MCCC robots has begun, and will be updated within the main section of the notebook as progress occurs
- We will base our decisions for the Worlds robots as we gather intel from MCCC

Weekly Goals and To-Do:

- Continue the Brainstorming phase for MCCC robots

Name	Task	Target Completion
All members	Discuss MCCC Robots	1/16/2024

Strategy

Progress Summary:

- The notebook and interview both performed well at NUKE, earning the Design Award

Meeting Accomplishments:

- We will continue to make progress within the notebook and practice our interview, so that we can keep up the good performance
- As Brainstorming occurs for the MCCC robots, we will document this in the notebook

Weekly Goals and To-Do:

- Document MCCC robot brainstorming
- Proofread the notebook before MCCC
- Compile the notebook for MCCC

Name	Task	Target Completion
Alex and Kathleen	Proofread the notebook	1/23/2024
Conner and Jacob	Compile the notebook	1/25/2024
All members	Document MCCC Brainstorming	1/18/2024

BLRS2 1/29 General Meeting

Date	@January 29, 2024
Category	BLRS2 General Meeting
Authors	J Jacob Zawacki N Noah Domogalik A Alex Lam
Design Cycle	Design Cycle 2

General Team

Progress Summary:

- BLRS2 won the Excellence Award, Robot Skills Champion, and Tournament Finalist at MCCC
    - This qualifies us for the World Championship in April!

Meeting Accomplishments:

- We will be attending our home competition on 2/24
    - As we are hosting a VEXU tournament with 30 teams from across North America, we will be competing for practice against some of the best teams in the world
- Our robots likely will not change much between now and our home competition, so that we are able to perform well at both and gather practice for our Worlds robots

Weekly Goals and To-Do:

- Prepare for Purdue within the coming month
- Begin preparations for our Worlds robots

Name	Task	Target Completion
All members	Prepare for Worlds	2/24/2024 (flexible)
All members	Compete at Purdue	2/24/2024

Competition Robots

Progress Summary:

- Both robots performed very well at MCCC
  - The concept of scoring Triballs from the match load zone using our wings proved very effective, as we were quickly able to score large amounts of Triballs

Meeting Accomplishments:

- We plan to keep very similar concepts moving forward for Purdue and Worlds
  - Purdue will likely serve as a “beta test” of our Worlds robots, using some of the subsystems to test before we fully commit

Weekly Goals and To-Do:

- Begin to make changes for Purdue and Worlds
  - Further discussion is required to determine what we would like to change and refine

Name	Task	Target Completion
All members	Discuss further changes for Worlds and Purdue	2/26/2024

Strategy

Progress Summary:

- The notebook and interview continued to perform well at MCCC, winning the Excellence Award!

Meeting Accomplishments:

- We will continue to keep the notebook updated as we move forward with potential subsystem redesigns, in preparation for the World Championship deadline

Weekly Goals and To-Do:

- Continue overall notebook progress

Name	Task	Target Completion
All members	Keep updating the notebook	As mechanics progresses

BLRS2 2/9 General Meeting

Date	@February 4, 2024
Category	BLRS2 General Meeting
Authors	Ⓙ Jacob Zawacki Ⓚ Kathleen Lowe Ⓒ Conner Siebert
Design Cycle	Design Cycle 3

General Team

Progress Summary:

- The competition robots have begun to shape as far as plans for the World Championship
- We have made a decision as to the state of the skills robots

Meeting Accomplishments:

- Continue progress for Purdue on 2/24 and Worlds in April
  - We should be ready to test mechanisms at our competition on the 24th
  - After Purdue competition, we will finalize our robots for the World Championship

Weekly Goals and To-Do:

- Continue progress on the robots for Purdue and Worlds

Name	Task	Target Completion
All members	Have robots ready for Purdue	2/20/2024

Competition Robots

Progress Summary:

- Our strategy for Worlds has been set
  - One robot focusing more on defense with another on offense
    - Both robots still having excellent offensive capability

    - Both robots being able to descore from under the opposing goal
        - Even if we are unable to descore due to double-zoning rules, this will still give the opponents one more thing to worry about
- Chassis redesigns have begun with these strategies in mind

Meeting Accomplishments:

- Further robot progress has been discussed:
    - We have begun the process of brainstorming endgame hang mechanisms
    - We need to determine which, if any, additional subsystems to redesign

Weekly Goals and To-Do:

- Continue to make progress for the Purdue/Worlds robots

Name	Task	Target Completion
Sean, Conner, Joey	Begin Hang Brainstorming	2/21/2024
Stephen, Dominic, Brandon, Joe	Continue chassis redesigns	2/18/2024

Skills Robots

Progress Summary:

- As detailed in the concluding entry in the skills category, we will be discontinuing the skills robots for the foreseeable future
    - Our method of scoring match loads using the wings at MCCC was very fast, and should be more than enough to perform well in the robot skills challenge
    - We will revisit the idea of skills robots after Purdue Competition on 2/24 in the event that a faster method of match loading is discovered

Meeting Accomplishments:

- In the meantime, Max will focus on hosting the Purdue VEXU Competition as the Event Partner, and Dominic will help build, program, and drive a competition robot

Weekly Goals and To-Do:

- Host the Purdue VEXU Tournament
- Remain vigilant for any new skills strategies

Name	Task	Target Completion
All members	Revisit skills robots if necessary	As needed
Max, All members as necessary	Host the Purdue VEXU Tournament	2/24/2024