Engineering Notebook

1599V

Team Number

Vietnamese Banana Farmers

Team Name

Saigon South International School

School

2023 - 2024 Over Under

VEX ROBOTICS

REC FOUNDATION

Engineering Notebook

2023-2024

1599V

Table of Contents

DATE	SUBJECT	PG
	Team Introduction	1
	Game overview + Plan	2
04/08/2023	Preseason Planning Day 1	5
07/08/2023	Preseason Planning Day 2	11
10/08/2023	Drivetrain day 1	19
12/08/2023	Drivetrain day 2	20
15/08/2023	Drivetrain day 3	21
16/08/2023	Day 4 Drivetrain	22
17/08/2023	Driv Program Day 1	24
18/08/2023	Sleds Day 1	26
22/08/2023	Wings Day 1	28
24/08/2023	Drivetrain day 5	30
24/08/2003	Sleds day 2	30
24/08/2023	Wings day 2	31
02/09/2023	Virtual Skills update	32
07/09/2023	Drivetrain day 6	33
09/09/2023	Sleds day 3	34
09/09/2023	Wings day 3	35
11/09/2023	Drivetrain day 7	36
14/09/2023	Sleds day 4	37
15/09/2023	Cata day 2	38
05/10/2023	Cata day 2.5	40
06/10/2023	Cata day 3	41
10/10/2023	Cata day 4	42
14/10/2023	Cata day 5	44
15/10/2023	Intake day 1	46
17/10/2023	Intake day 2	47

Table of Contents - 2

DATE	SUBJECT	PG
23/10/2023	Intake day 3	52
25/10/2023	Intake day 4	54
26/10/2023	Intake day 5	54
30/10/2023	Wings day 4	55
31/10/2023	Cata day 6	56
01/11/2023	Cata day 7	57
03/11/2023	Cata day 8	58
06/11/2023	Cata day 9	60
07/11/2023	Cata day 10	63
07/11/2023	Sleds day 5	64
08/11/2023	Cable management	65
10/11/2023	Intake day 5	66
10/11/2023	Sleds day 6	67
10/11/2023	Programming update	68
10/11/2023	Cata day 11	69
10/11/2023	RGB	69.1
11/11/2023	Scrimmage	70
13/11/2023	Post-Scrimmage reflection	73
14/11/2023	Intake day 6	78
15/11/2023	Intake day 7	79
16/11/2023	Drivetrain day 8	81
16/11/2023	PID explanation	81.1
16/11/2023	Intake day 8	82
17/11/2023	Puncher day 1	83
17/11/2023	Wings day 5	84
18/11/2023	Wings day 6	85
18/11/2023	Wedges day 1	85
19/11/2023	Odometry	85.1

Table of Contents - 3

Date	SUBJECT	~~PG~~ PG
20/11/2023	Intake day 9	86
20/11/2023	Puncher day 2	86
23/11/2023	Autonomous day 1	87
25/11/2023	Puncher day 4	88
27/11/2023	Puncher day 5	90
27/11/2023	Intake day 8	91
28/11/2023	Puncher day 6	91
28/11/2023	Auton day 2	92
01-02/12/2023	Formosa	94
04/12/2023	Post-Formosa reflection	97
04/12/2023	Puncher day 7	98
06/12/2023	PID tuning	99
08/12/2023	Puncher day 8	100
11/12/2023	Puncher day 9	101
11/12/2023	Intake day 10	101
13/12/2023	Drivetrain day 3	102
13/12/2023	Puncher day 10	102
14/12/2023	Intake day 11	104
16/12/2023	Intake day 12	105
16/12/2023	puncher day 11	105
16/12/2023	Blocker day 1	106
16/12/2023	slods day 7	107
16/12/2023	Programming update	107
16/12/2023	Autonomous day 3	109
17/12/2023	Pullbright	110
19/12/2023	Autonomous day 4	115
26/12/2023	Puncher day 12	116
28/12/2023	puncher day 13	119

Table of Contents - 4

Date	SUBJECT	PG.
28/12/2023	More motion algorithms	120
31/12/2023	Puncher day 14	121

Meet the Team
1599V

Wayne

Team leader +
Builder + researcher

Jack W

Lead builder +
Media manager

Bao Nhi

Lead programmer +
Documenter

Alex

Programmer +
CAD designer

Jack L

Driver +
Builder

Derek

Builder

Aaron

Builder

Hung

Programmer +
Builder

2023 - 2024

OVER-UNDER

GAME OVERVIEW + PLAN

THE FIELD

- 2 Alliance Triballs per alliance
- 22 Match Load triballs per alliance
- 12 begin on the field

60 triballs total

SCORING ELEMENT: TRIBALL (Game manual pg. A11)

Slightly rounded triangular pyramidal shape known as Reuleaux triangle

Mass: 110 ± 20g

GAME DESCRIPTION (Game manual pg. 2)

- Two alliances: "red" and "blue" composed of two teams each
- Matches consist of a 15 second Autonomous Period followed by a 1:45 Driver Controlled Period

* OBJECT OF THE GAME: attain a higher score than the opposing Alliance by

- Scoring Triballs in goals
- Elevating at the end of the match

(Game overview + Plan cont.)

SCORING

Autonomous Bonus	8 Points
Each *Triball* Scored in a *Goal*	5 Points
Each *Triball* Scored in an *Offensive Zone*	2 Points
Elevation - Top Tier	20 Points
Elevation - 2nd Tier	15 Points
Elevation - 3rd Tier	10 Points
Elevation - 4th Tier	5 Points

- An Autonomous win point is awarded to any alliance that completes the 3 assigned tasks by the end of the Autonomous Period:
  1. Removed triball from Alliance's Match Load Zone that coincides with their Starting Tiles
  2. Scored at least 1 Alliance Triball in Alliance's own Goal
  3. Ended the autonomous period with at least 1 Robot contacting their own Elevation Bar
- An Autonomous Bonus is awarded to the Alliance that has the most points by the end of the Autonomous Period

OTHER SCORING RULES

A triball is considered scored in a goal if:

a. The triball is not contacting a robot of the same color alliance as the goal

b. At least two (2) corners of the Triball are within the goal (i.e., are under the net and have "broken the plane" of the edge of the PVC pipes that defines goal volume)

A triball is considered scored in an offensive zone if:

a. the Triball is not contacting a robot of the same color alliance as the Offensive Zone

b. the triball is contacting the gray foam tiles within the offensive zone

OTHER IMPORTANT RULES

- Possession is limited to 1 triball
- 2 preloads per alliance (1 per robot)

DOUBLE-ZONING

- when both alliance robots are in the same Offensive zone → opponent allowed to "de-score" - "break the plane" and go into the double-zoning alliance's goal to remove triballs

* using concave portions of a robot to control the movement of triballs is considered possession.

(Game overview + plan cont.)

CONSTRAINTS

- Starting size restrictions : 18" x 18" x 18"
  - Can't expand over 36" in any horizontal dimension
- All functional plastic (non-shattering, ~~0.7~~ 0.070" or thinner) on the robot fits within a 12" x 24" rectangle
- Total power of motors limited to 88W equivalence

  # of 11W _____ x 11 = _____

  # of 5.5W _____ x 5.5 = _____

  Total power = _____ (Sum of above)

- maximum 2 pneumatic air resivors (max 100 psi / resivor)

SUBSYSTEMS / SUBPROBLEMS

- Drivetrain (1)
- Shooting mechanism (2)
- Intake mechanism (3)
- De-Scoring mechanism (2)
- Climbing mechanism (4)

we don't want to prioritize climbing, since the top tier is 20 points (= 4 triballs) and we think we could spend the climbing time trying to score more triballs instead

priority

PRESEASON PLANNING

04/08/2023 Day 1 Present: Alex, Bao Nhi, Jack L, Jack W, Hung

We are still on summer break, so we started planning for our robot on calls before the season starts at school.

(problem / question)

Since the drivetrain is one of our top priorities, we discussed some potential ideas for it.

We considered the following designs: (research) (preliminary)

① Tank drive

PROS:
- Simple to build
- Consistently preforms well

CONS:
- Able to be pushed sideways
  - Solution: bling drive
- Can't strafe ☹

source: Purdue Sigbots BLRS Wiki

- The tank drive was the design we used last year (22-23 spin-up)
  - we used a 4motor (4M), 4 wheel (4W), 200 rpm drivetrain.

  The gear ratio we used:

ratio: 60:60 (1:1)

rpm: 200

Something cool about tank drives is we can customize the number of wheels and number of motors. We can also mess around with a lot of gear ratios.

↳ This "customizability" might be a pro for tank drives!

Some examples:

VEX wheels come in different sizes:
- 4" ← what we used last season
- 3.25"
- 2.75"

Motors & motor placement

4 motors

2 motors

6 motors

② Mecanum drive

PROS:
- High forward traction
- Ability to strafe

CONS:
- Slow (vs tank drive) → we want to be able to go fast
- inefficient strafing
- Space inefficient → we need space for our subsystems

③ X-drive

Pros:
- Most manuverable + efficient
- Precise, agile in all directions
  → accuracy / agility advantage

Cons:
- Wheels take up lots of space
- Bit more complex to build

- X-drives are known for their agility and ability to strafe.

For example, to move right,

To move forwards,

Presenson planning - day 1 - cont.

④ H-drive

Pros:
- Can strafe to the sides
- Also pretty simple to build

Cons:
- Takes an extra motor
- Less force vs tank / X-drive
  (source: experiment by Engineering girl on YouTube)

Things we want our robot to be able to do: (restraints / problem)

- move quickly
  - learning from last year, we think our robot for this year needs to be able to drive faster. Our robot last year was a 4" 4w tank drive at 100 RPM.
    This gives us a linear distance of:

$$4\pi\,\text{in} \times 100\,\tfrac{\text{rev}}{\text{min}} = 400\pi\,\tfrac{\text{in}}{\text{min}} \approx 1256\,\tfrac{\text{in}}{\text{min}}$$

    We felt that this was too slow, so we will try to make our robot faster.

- lightweight
  - we felt that our robot from last year sinked too much into the foam.
  - we also think a light robot would be helpful if we ever decide to implement a climbing mechanism.

- be able to go over the barrier → this means we need lots of torque
  - We think a robot that would be able to go over the barrier in the middle is important so our robot can move around the field more effectively, helping us be able to control the field better. It can also help us switch between offense / defense quickly.

- low-profile (go under the goal)
  - we want our robot to be able to go under the goal so we can push triballs out and descore our opponents if they double zone. we watched videos of some scrimmages online and found that double zoning happens quite often.

FINAL DECISION

Using process of elimination, we narrowed down our options:

	SCORE (/10)
· Tank drive	2
· ~~Mecanum drive~~ → too slow	2
· ~~X-drive~~ → Takes up space that we need for subsystems	5
· ~~H-drive~~ → less force - difficult to go over barrier	3

Preseason planning - day 1 - cont.

Therefore, we decided to go with the tank drive design for our drivetrain.

DRIVETRAIN DECISION MATRIX - 04/08/2023

	Weight (/5)	Tank drive		Mecanum drive		X-drive		H-drive	
		Rating (/5)	Weighted Score	Rating (/5)	Weighted Score	Rating (/5)	Weighted Score	Rating (/5)	Weighted Score
Consistency	4	5	20	4	16	5	20	4	16
Maneuverability	1	3	3	5	5	5	5	4	4
Stability (resistance to being pushed)	3	3	9	3	9	3	9	4	12
Space efficiency	4	5	20	3	12	2	8	2	8
Motors / resources	2	5	10	5	10	4	8	2	4
Force / strength (to go over barrier)	5	4	20	3	15	3	15	2	10
Speed	4	3	12	2	8	5	20	3	12
Ease of building	2	5	10	3	6	1	2	4	8
Ease of programming	2	5	10	2	4	1	2	4	8
Total Score			114		85		89		82

Final Decision!

es)

e design we used last year.

linear distance → faster (?)
to go over barrier

u-profile

+ more
(?) torque!

(?) → would require steels,
more complicated!?

a bit too much
version of the

tick with either
w (☹).

e team, we think the

concentrated
at 4 points

more evenly
distributed (?)

[view from below]

Presenson planning - day 1 - cont.

Therefore, we decided to go with the tank drive design for our drivetrain.

Next, we looked at different number of wheels (and sizes).

① 4 wheels (4")

← 4" wheels

This is the design we used last year.

Pros:
- higher linear distance → faster (?)
- easier to go over barrier

Cons:
- not low-profile

② 6 wheels (3.25")

Pros:
- low-profile
- less sinking in foam field (?)

→ + more torque!

Cons:
- not as much ~~speed~~ speed (?) → would require sleds, more complicated!?

③ 8 wheels (3.25")

We felt that 8 wheels was a bit too much and just a more extreme version of the 6W drive (②).

Therefore, we wanted to stick with either the 4W drive (①) or 6W (②).

Since a major concern we had was the robot sinking into the foam, we think the 6W drive would address this issue better than the 4W drive.

4 points of contact

← concentrated at 4 points

← 6 points of contact

more evenly distributed (?)

[view from below]

Preseason planning - day 1 - cont.

Also, being able to go over the barrier easily is an extremely important factor for us. Although the bigger wheel may help us (since it is taller), it also has more traction compared to the 3.25" wheels. We felt that this might affect our overall gameplay.

To go over the barrier, we think a lot of torque would be needed. Therefore, we want to use 6 motors for the drivetrain. We looked on vexforums and saw that 3.25" generally provided more torque and is also a better option for 6M drivetrains.

FINAL DECISION

We decided to go with the 6 × 3.25" wheels tank drive.

GEAR RATIOS!

We looked at different gear ratios we could use for a 6W 3.25" drivetrain.

We considered using ~~X~~ 1 of these 2 motor cartridges:
- 18:1 - green (200 RPM)
- 6:1 - blue (600 RPM)

We collaborated on a google sheets document to look at possibilities:

Cartridge RPM	Input Gear	Output Gear	Output RPM	In/s on 4.125"	In/s on 3.25"	Stall Torque Nm
200	48	36	266.7	57.596	45.379	4.73
200	60	36	333.3	71.995	56.723	3.78
600	36	72	300.0	64.795	51.051	4.20
600	36	84	257.1	55.539	43.758	4.90
600	48	84	342.9	74.052	58.344	3.68
600	36	60	360.0	77.754	61.261	3.50
600	48	72	400.0	86.394	68.068	3.15
600 48:84	343	49.368	58.344	74.052	9.928	10
					gap	
600 36:60	360	51.836	61.261	77.754	10.425	7
						8
600 48:60	480	69.115	81.681			8
600 1:1	600	86.394				

We then narrowed the possibilities down to the RPMs we found suitable (but not too fast so we have enough torque) ~~and also~~ and "fast enough". We also considered which gears were able to fit the configuration we decided on (6W, 3.25")

200	12	24	36	48	60	72	84
12							
24							
36				267	333		
48							
60							
72							
84							

too big gear

600	12	24	36	48	60	72	84
48							
60			360				
72			~~300~~				
too big 84			~~257~~	~~343~~			

Our options now are: ~~36:48~~ 48:36 ~~36:60~~ 60:36
- 18:1 motor: ~~48:36~~, ~~60:36~~
- 6:1 motor: ~~36:60~~ ~~48:60~~ 36:60

Preseason planning - day 1 - cont.

We didn't really like the idea of gearing down from a 200 RPM gear, which would make the input gear too big.

Ex: 200 RPM motor, ~~36:60~~ 60:36 (333 RPM)

60:36 → 5:3 ratio

~~36:60 → 3:5 ratio~~

$\frac{5}{3} \times 200\text{ RPM} = 333\frac{1}{3}\text{ RPM}$

Therefore, our only remaining option was gearing up from a 6:1 motor, 36:60

36:60 → 3:5 ratio

$\frac{3}{5} \times 600\text{ RPM} = 360\text{ RPM}$

This also gives us a linear speed of:

$$3.25\pi \frac{in}{rev} \cdot 360 \frac{rev}{min} \cdot \frac{1\ min}{60\ sec} = 19.5\pi \frac{in}{s} \approx \boxed{61.23 \frac{in}{s}}$$

Which is much faster than our robot from last season, which had a linear speed of:

$$4\pi \frac{in}{rev} \cdot 200 \frac{rev}{min} \cdot \frac{1\ min}{60\ s} = 13.\overline{3}\pi \frac{in}{s} \approx \boxed{41.87 \frac{in}{s}}$$

Also, the torque:

stall torque of 36:1 motor (red) = 2.1 Nm (according to the VEX website)

∴ stall torque of 6:1 motor (blue) = $\frac{2.1\text{ Nm}}{6} = 0.35\text{ Nm}$

$$\underbrace{0.35\text{ Nm}}_{\text{torque/motor}} \times \underbrace{6}_{\#\text{ motors}} \times \underbrace{\frac{60}{36}}_{\text{ratio}} = \boxed{3.5\text{ Nm}} \approx 30.9776\text{ in-lbs}$$

We also looked at the factor of spacing and whether it would fit the stands in the c-channel

it seemed to be a perfect fit!

Final decision

We decided to use the 36:60 gear ratio, which looked something like this

~~the~~ Preseason planning

07/08/2023 Day 2 Present: Alex, Bao Nhi, Jack L, Jack W, Hung

Today we wanted to look at options for a shooting mechanism.

Our goals for the shooter: (problem)

- we noticed that there are a lot of preloads in the game, and we think shooting these preloads can help us gain a significant amount of points. Therefore, the shooter needs to be easily accessable so we can reload it quickly and shoot preloads.

- We want a shooting mechanism that can shoot quickly

- While at worlds last season, we heard many teams discuss about how the triball is slightly larger than the height of the goal. We want to have a shooter that can shoot over the barrier and into the opposing goal (if possible). Therefore, our shooter would also need to be strong.

Potential shooter designs: (Research)

- After watching many reveal videos on YouTube, we noticed that there were 3 main types of shooters:
  1. Catapult (74248Z)
  2. Flywheel (1MARC)
  3. Puncher (Akiva)

- we delved deeper into each of these 3 main options:

① Catapult

For a catapult, we are thinking about using a slip gear and elastics

Teeth engaged Teeth not engaged

when the teeth are not engaged this allows the rubber bands to pull the arm and launch the triball

vs. when the teeth are engaged they push against ~~ther~~ the rubber bands.

This means the more elastics (rubber bands) we use, the more resistance there will be and hence the more torque we would need, to be able to push against it.

- we can use different gear ratios to adjust torque

Preseason planning - day 2 - cont.

Some other things to consider for catapults (optimization)

- Rotational speed
  - faster arm swing → faster the object will move through the air
  - noticed most teams using 33 rpm catapults
- Length of arm
  - object farther away to center → launches further
- Launch angle
  - Fine tune the trajectory of the triball

Some catapult designs we saw online (for reference):

↑606X

21417A

Pros:
- easy to tune force by changing # of rubber bands
- force stays consistent
- shoots quickly (doesn't need a spin-up time like flywheels)
- kee low-profile
- can hold the triball's odd shape (reuleaux tetrahedron)

Cons:
- tension of rubber bands → large stress on robot
- requires lots of force to draw catapult back
- might cause robot to shake around when shooting
- rubber bands need to be replaced often outside of matches

Preseason planning - day 2 - cont.

② Flywheel

We also used a flywheel last season, but it looked more like this:

- An issue we had with our flywheel last season was that it took longer to "spin-up" (reach its max RPM). Therefore, we had to wait a while until we could shoot the disk.

- compared to teams with catapults, we felt that our flywheel was slow and couldn't shoot the game object quickly enough.

Pros
- Simple / easy to build
- can shoot continuously (sort of)
- Easy to tune by changing RPM

Cons:
- rubber bands / sprocket design - rubber bands might snap
- "spin up" time
- RPM drops when shooting
- Takes up a lot of space (not low profile)

~~③~~ some photos of flywheel designs:

169A Turning Point

1104M Nothing But Net

Preseason planning - day 2 - cont.

⑤ Linear Puncher

We noticed that a linear puncher was a lot similar to our indexer design last year, except with a slip gear and rubber bands to pull it backwards quicker.

From what we saw, it works similar to a catapult, except linear:

① The gear teeth pushes the rack gear backwards and counteracts the ~~force~~ pull of the rubber band. This is considered the "loading" phase.

② When the slip part of the gear (teeth shaven off) gets to the rack gear, since there is nothing holding it back, the rack gear gets pulled forwards by the rubber bands.

③ The gear continues spinning and repeats back to step ① again when the teeth of the gear reach the rack gear.

Some examples of punchers:

~~5746 - Turning Point~~
5735Y on YouTube

5746 - Turning Point

Modifications: (some we can consider)

- ratchet system that would allow puncher to stay ~~t~~ in place when pulled back
- angle adjuster to modify trajectory of the shot
  - can be integrated with other sensors like vision sensor to create automatic / assisted aim for the driver

(source: Purdue BLRS wiki)

Preseason planning - day 2 - cont.

Pros
- Relatively compact compared to other shooting mechanisms
- Easy to build
- Loses no energy to spin
- less violent firing action than a catapult

Cons
- need a shaved gear
- slower fire-rate than a flywheel
- lots of wear and tear on back stop
- rubber bands might need to be replaced

Puncher design for over under by joshhess7358 on YouTube

FINAL DECISION

we decided to go with the catapult since it was relatively easy/simple and also met all our requirements. We also liked the customizability of catapults.

we can save punchers as a future option to look into, but for now, we plan on implementing a catapult for our shooting mechanism.

We also tried to plan our motor distribution since we only were allowed 88W of motors.

- What we decided on (this is general, not final!)

SUBSYSTEM	MOTORS	TOTAL
Drivetrain	6 x 11W	66W
Catapult	1 x 11W	11W
Intake	1 x 11W	11W
		88W ✓

we wanted a fast drivetrain but still enough torque to go over the barrier

- We discussed using PTO or a transmission (a mechanism that allows a single motor to be "shared" used in multiple subsystems ~~through~~ by gears and pistons).
  - We ultimately decided that PTO was too complex for us at the time (we might try using it in the future). We decided to stick with the planned motor distribution for now.

Shooter Decision Matrix - 07/08/2023

	Weight (/5)	Catapult		Flywheel		Linear Puncher	
		Rating (/5)	Weighted Score	Rating (/5)	Weighted Score	Rating (/5)	Weighted Score
tunable / customizable	3	4	12	3	9	2	6
consistency	5	5	25	3	15	3	15
reload / "spin up" time	4	4	16	2	8	4	16
size (low profile?)	5	5	25	3	15	2	10
wear and tear	3	4	12	4	12	1	3
replacements (rubber bands?)	1	3	3	5	5	3	3
Total Score			93		64		53

strategies and what things
ete the tasks.

prevent our opponents from

DEFENSIVE

ng ~~an~~ opponent's matchloads
ker?)

triballs out of goal (De-scoring)
?)

tch loads, so we want a fast
ese matchloads is to have the
bar and running the catapult

the goal after they have been

emely useful since they can
balls into goal) and defensely
we might want to also

e other side from shooting. We
if we have time, since we
wings serve more function)

s, but after some discussion, we
we could be using the climbing
iballs (4×5pts = 20pts) is equal

nd we don't plan on building

offensively and defensively, since
will be like. we also didn't
artner for scoring

to be able to play well:

15.1

side using our catapult
(hopefully our alliance partner can push them into the goal from the other side)
De-score if our opponents double-zone

2) Collect & move triballs into the goal
Push triballs our alliance partner sends over into the goal (if they have a shooting mech)

Preseason planning - day 2 - cont.

STRATEGY DISCUSSION!

- we moved on to discussing some potential gameplay strategies and what things we would need on our robot in order to complete the tasks.

- we first thought of the ways to gain points and prevent our opponents from gaining points:

OFFENSIVE	DEFENSIVE
· Shooting matchloads (using catapult)	· Blocking ~~an~~ opponent's matchloads (blocker?)
· Push triballs into goal (wings?)	· Push triballs out of goal (De-scoring) (wings?)
· Endgame (climbing mechanism?)	

- We noticed that a lot of the triballs are match loads, so we want a fast catapult. A potential strategy for shooting these matchloads is to have the back of our robot touching the match load bar and running the catapult continuously while we load it.

- We can use wings to push the triballs into the goal after they have been launched over by the cata.

  - We noticed that wings would be extremely useful since they can be used both offensively (pushing triballs into goal) and defensely (de-scoring our opponent). Therefore, we might want to also prioritize wings.

- We also want a large blocker to prevent the other side from shooting. We decided we will try to add a blocker only if we have time, since we decided to prioritize wings over a blocker (wings serve more function)

- We wanted to be able to climb for 20 points, but after some discussion, we ultimately decided it wasn't worth it since we could be using the climbing time to score more triballs, as scoring 4 triballs (4×5pts = 20pts) is equal to the top tier elevation.

  ⇒ Climbing won't be one of our priorties and we don't plan on building a climbing mechanism

- We want a robot that could play well both offensively and defensively, since we don't know what ~~a~~ our alliance partners will be like. We also didn't want to be too reliant on our alliance partner for scoring

  - The 2 main roles we want our robot to be able to play well:

    1) Shoot triballs quickly to the other side using our catapult (hopefully our alliance partner can push them into the goal from the other side)
       De-score if our opponents double-zone

    2) Collect & move triballs into the goal
       Push triballs our alliance partner sends over into the goal (if they have a shooting mech)

Preseason planning - day 2 - cont.

- After watching some reveal videos on YouTube, we noticed that the most common bot was similar to 229V's, which can shoot well and has wings:

photos from 229V on YouTube

- however, this robot appears to be too tall to go under the goal and de-score.

- since having a low-profile robot is one of ~~our~~ the main goals for our robot design, we then looked for more robots that were specifically low-profile and met our constraints.

- A robot from 21417A (Robokaus on YouTube) especially inspired us. It was low-profile and had all the features we wanted except blocking. We think we can add a blocker on a design like this to make a complete, well-rounded robot.

DAY 1 - 10/08/2023

Present: Alex, Bao Nhi, Jack W, Jack L, Hung, Aaron, Wayne

DRIVETRAIN (CAD) (DESIGN)

- We wanted a very low friction drivetrain so we can use the motors more efficiently (less work for the motors). A solution we discussed was screw joints

screw joints (new)

drivetrain shaft (new)

- An issue is that the longest legal screws are 2.5"
  - This means that if we placed our c-channels like][we would get a 4 wide gap.
    - however, if we get hit from the side like this, our robot would be easily damaged.
  - Instead, if we put our c-channels like]], we get a 3-wide gap.
    - Therefore, to withstand pushes from the side, we need 3 wide.
    - 4" wheels fit into a 3-wide gap with only 0.005" tolerance, which is way too tight. The antistatic 4" wheels fit but we don't have them.
      - This gives us more reason to use 3.25" wheels instead of 4"!
- We also decided to use zipties instead of screws to attach the bearings on the c-channels.

zipties

screws

zipties could help us save weight. Also, they don't get loose like screws do (re-screwing these c-channels would mean taking apart the whole dt) although they are prone to snapping.

Day 1 cont.

· we CADed out the ideas we discussed:

][c-channels:

- 4 wide
- easily damaged when pushed from side
- HS gear usable
  - we don't really see a need for using HS gears though

]] c-channels:

- 3 wide
- more "protected", less vulnerable to damage when being pushed from side
- LS gear, not a lot of width
- more space-saving (?)

Drivetrain design, Iteration 1.0

left side design (right is symmetrical)

It looks like this when CADed out

DAY 2 - 12/08/2023

Present: Alex, Bao Nhi, Jack L, Jack W, Hung

We began building the drivetrain! (BUILD / CREATE)

This was over the weekend so we met up at Bao Nhi's house to build. However, we quickly realized that we didn't bring enough parts:

- not enough 2.5" screws
- we didn't have a drill to drill the LS gear for screw joints

Therefore, we tried to build just a general drivetrain (without screw joints). This design is similar to the one we CAD'ed, except it uses HS gears and ~~axl.~~ shafts instead of screw joints.

very no friction wheels

football with our new drivetrain

DAY 3 - 15/08/2023

Present: Bao Nhi, Alex, Jack L, Jack W, Hung, Wayne

Drivetrain day 3 (build/create)

- Now that we were back from the weekend, we can work on rebuilding our dt according to our plan (the CAD).

- we started by attaching the gears to the wheels

- Then, we continued to put it together. We completed one of the two sides today (RHS)

- The spacing we used:

DAY 4 - 16/08/2023

Present: Bro Niki, Alex, Aaron, Jack L, Jack W, Wayne, Hung

Drivetrain day 4 (build/create)

- We completed the other half (LHS) of our drivetrain!

- We attached the 2 sides together using 25-long c-channels

Front:

Back:

Day 4 cont.

- How we attached the 3rd motor on each side:

- We also looked at the friction at our dt. We noticed that the RHS had less friction than the LHS, which means it's easier to spin and spins for longer.

QR code of our testing:

This could interfere with our robot's ability to move straight forwards.

DAY 5 - ~~18~~ 17/08/2023

Present: Bao Nhi, Alex, Jack L

Program day 1

- we finished wiring the drivetrain and wrote a basic drivetrain program so our driver (Jack L) can start practicing.
- we looked at potential controlling ~~systems~~ settings we could use:

① TANK DRIVE

Pros:
- easy to program
- direct control => more maneuverability

Cons:
- unintuitive, difficult to use

② Arcade drive (1 joystick)

Pros:
- very intuitive, easy to drive

Cons:
- less control / maneuverability since it's all dictated by a singular joystick

③ Arcade drive

Pros:
- intuitive
- in place turning => smoother

Cons:
- less maneuverability vs tank drive

- our driver Jack L said he prefers to drive something intuitive since he has experience playing videogames with joysticks like the arcade drive.
  - Therefore, we were willing to sacrifice some maneuverability for more efficiency (intuitiveness) in our dt controls.
- Between ② and ③, we went with ② since it had more control while also still being intuitive to control.

FINAL DECISION

We decided to go with ②, a 2-joystick arcade drive since it was intuitive to use.

Program PLAN (pseudocode)

- we started by writing out some pseudocode for a basic ~~dt~~ arcade drive.
- We used a linear arcade drive for now, and we are thinking about implementing an scurve ~~dt~~ or cheesy curve later.

Day 54 cont.

program pseudocode:

Left Y = controller axis 3 pos.

Right X = controller axis 1 pos.

Left Val = Left Y + Right X

Right Val = Left Y − Right X

spin left motors forward, velocity Left Val

spin right motors forward, velocity Right Val

Program iteration 1: (create)

```
double throttle = Controller1.Axis3.position() * 12 / 100;
double turn = scurve(Controller1.Axis1.position(), 1.24);

double leftVal = throttle + turn;
double rightVal = throttle - turn;

left_motor_group.spin(directionType::fwd,leftVal, voltageUnits::volt);
right_motor_group.spin(directionType::fwd,rightVal, voltageUnits::volt);
```

Program iteration 1: (testing)

"Scurve" was a curve function we tried to add but didn't work, so we'll just go with linear controls (no curve).

So, we changed the 2nd line to:

double turn = Controller1.Axis1.position() * 12 /100;

DAY 6 - 18/08/2023

Present: Alex, Bao Nhi, Jack L, Jack W, Hung, Wayne

Sleds day 1

(~~research~~) ① identify the problem - constraints

- we saw that sleds was used by many teams to get over the barrier
- we want our sleds to make our robot able to go over the barrier, while:
  - staying within the 18" x 18" starting limit
  - sturdy and durable - doesn't break mid-game and we don't have to replace it after every match

② Research / identify possible solutions

- some sled designs we saw online:

③ Choosing the "best" solution

- we wanted to start with something simple, so we want to go with 4 polycarb pieces attached to the end of our robot (see design below)

④ Plan / Design

side view:

(both sides are symmetrical, this is just showing 1 side)

Day 6 cont.

⑤ Build / create!

- we cut out the design from polycarbonate plastic.
- we used a CO_2 laser cutter but it kept burning
  - our sleds came out looking a bit burnt (it's not a bug it's a feature)
  - the laser cutter didn't cut it through fully so we had to break it by bending and use the drill press to make drill holes

Final shape of our sleds + odom stress test

- We applied 5 lbs of force up and 2 lbs to the side to see how much it would bend.
- It uses the properties of polycarbonate to simulate how much it will bend with that force

DAY 7 - 22/08/2023

Present: Alex, Bao Nhi, Jack L, Jack W, Wayne, Hung, Aaron

Wings day 1

- we wanted to build wings to go along with our low-profile robot. This would allow us to descore our opponents if they double-zone.
  - It can also help us push many triballs into the goal at once.

① Identify the problem / constraints

- we want to build wings that are strong and can push multiple triballs at once.
- we need to be careful and make our wings flat, since moving a triball with a concave ~~surface~~ face counts as possesion (game manual pg. 19). We can only posses 1 triball at a time.
- our wings have to be within the 36" horizontal expansion limit (rule <SG2>)

② Research

- We noticed that many teams use pneumatics / pistons for their wings

- we also saw that many teams had locking mechanisms (to save pneumatics air)
  - we don't plan on implementing a locking mech. on our first prototype since we wanted to keep it simple. We might (and would likely have to) implement a locking mechanism in a later iteration.

③ Plan

Day 7 cont.

⑤ Build / create!

We started by building the LIFS to use as a proof of concept

```
void wings_expand(){
    piston_wings.set(true);
}

void wings_retract(){
    piston_wings.set(false);
}
```

Program iteration 1.1

we want to program the wings to open and shut.

Plan: (pseudocode)

Function for opening wings:
set the pistons as true

When button is ~~pressing~~ pressed
open wings

Function for closing wings:
set the pistons as false

when button is released
close wings

```
//wings
Controller1.ButtonR2.pressed(wings_expand);
Controller1.ButtonR2.released(wings_retract);
```

DAY 8 - 24/08/2023

Present: Alex, Bao Nhi, Jack L, Jack W, Wayne, Hung, Aaron

Testing!! (all the components so far)

Drivetrain day 5 (testing) (evaluation)

- After driving the robot around, the first thing we noticed was the big difference in speed compared to last year. Our robot was much faster, which is exactly what we wanted!
- After driving for ~5 minutes, we checked the motors for overheating. They weren't, which is a good sign. Since a match is only 2 minutes, we were feeling pretty good on our motors' temperature.
- We're feeling pretty good about our drivetrain so far after testing. We will continue frequently ~~the~~ testing our drivetrain and address any issues that might arise.

Sleds day 2 (testing) (evaluation)

- Our sleds sometimes worked and sometimes didn't (like a 50-50)
  - we think this could be due to the weight distribution and skewed center of mass, which is only temporary (since we will be adding other parts like cata and intake)
- While testing, the shock from going over the barrier momentarily disconnected the battery from the brain, causing the brain to white-screen/glitch and stop working.
  - As a temporary solution, we ziptied the battery and the whole cable down onto the robot. This isn't very viable since we would frequently need to change our battery in a tournament

- Our sleds were bent by the end of our testing. This isn't good since one of our goals was to make strong sleds
  - This probably ~~eat~~ is a problem with the shape of the sleds itself, so we might need to completely redesign our sleds.

Day 8 cont.

Wings day 2 (testing) (evaluation)

- our wings worked but they weren't very strong, especially when pushing the triballs.

The wings expanded fine but the triballs were pushing the wings inward

- This is likely due to our lack of a locking mechanism. From this, we learned that a locking mechanism would be vital in our wings designs.

Programming day 2 (testing) (evaluation)

- The code works but the turning is too sensitive. We are thinking about implementing an S-curve or something similar to make it easier for our driver to control turning.

Virtual Skills Update - 02/09/2023

Alex

We got 1st place virtual skills w/ 72 points!

Rank	Score	Stop Time	Team Number	Team Name	Organization	Event Region	Country / Region
1	70	1	1599V	SSIS Dragons	Saigon South International School (SSIS)	Vietnam	Vietnam
2	67	0	603R	Belmont High School	BELMONT HIGH SCHOOL	New Hampshire/Vermont	United States

The program:

```
drivetrain.set_heading(gps.heading(), DEGREES)
drivetrain.set_drive_velocity(100, PERCENT)
drivetrain.set_turn_velocity(100, PERCENT)
intake_motor.set_velocity(100, PERCENT)
arm_motor.set_velocity(100, PERCENT)

def goto(x2,y2,d=FORWARD,sbt=False,wt=True):
  x1, y1 = gps.x_position(MM), gps.y_position(MM)
  if x1 == x2 and y1 == y2: return
  turnto(x2,y2,d)
  l = math.sqrt((x2-x1)**2 + (y2-y1)**2)
  if sbt == True: l -= 200
  drivetrain.drive_for(d, l, MM,wait=wt)
  x3, y3 = gps.x_position(MM), gps.y_position(MM)
  if abs(x3-x2) > 10 or abs(y3-y2) > 10: goto(x2,y2,d,sbt,wt)

def turnto(x2,y2,d=FORWARD,wt=True):
  x1, y1 = gps.x_position(MM), gps.y_position(MM)
  try:
    a = math.degrees(math.atan((x2-x1)/(y2-y1)))
  except ZeroDivisionError:
    if x2 > x1:
      a = 90
    elif x2 < x1:
      a = -90
    else:
      pass
  if y1 > y2: a -= 180
  if d == REVERSE: a -= 180
  drivetrain.turn_to_heading(a, DEGREES,wait=wt)

def main():
  arm_motor.spin_to_position(1180, DEGREES, wait=False)
  goto(-900,000)
  drivetrain.turn_to_heading(-140, DEGREES)
  wait(0.1,SECONDS)
  intake_motor.spin_for(REVERSE, 400, DEGREES)
  intake_motor.spin(FORWARD)
  goto(-900,-600)
  drivetrain.turn_to_heading(-90, DEGREES)
  intake_motor.spin_for(REVERSE, 360, DEGREES)

  intake_motor.set_velocity(100, PERCENT)
  goto(-400,-600)
  intake_motor.spin_for(FORWARD, 1000, DEGREES, wait=False)
  goto(850,-500)
  drivetrain.turn_to_heading(80, DEGREES)
  intake_motor.spin_for(REVERSE, 360, DEGREES)
  goto(150,-900)
  intake_motor.spin_for(FORWARD, 200, DEGREES, wait=False)
  goto(850,-500)
  drivetrain.turn_to_heading(80, DEGREES)
  intake_motor.spin_for(REVERSE, 360, DEGREES)

  intake_motor.spin(FORWARD)
  goto(1400,-1400)
  arm_motor.spin_to_position(1300, DEGREES)
  goto(1500,-900)
  intake_motor.spin_for(REVERSE, 360, DEGREES)

  intake_motor.spin(FORWARD)
  goto(150,-230)
  wait(0.1, SECONDS)
  goto(850,-100)
  intake_motor.spin_for(REVERSE, 360, DEGREES)
  intake_motor.spin(FORWARD)
  goto(-250,-50)
  goto(500,0,REVERSE)
  goto(850,0)
  drivetrain.turn_to_heading(100, DEGREES)
  intake_motor.spin_for(REVERSE, 360, DEGREES)
  intake_motor.spin(FORWARD)
  goto(700,0)
  goto(120,390)
  wait(0.1,SECONDS)
  goto(600,300,REVERSE)
  goto(850,280)
  intake_motor.spin_for(REVERSE, 360, DEGREES)
  intake_motor.spin(FORWARD)
  goto(700,500)
  goto(150,850)
  wait(0.1,SECONDS)
  goto(850,400)
  drivetrain.turn_to_heading(115, DEGREES)
  intake_motor.spin_for(REVERSE, 360, DEGREES)

  goto(600,600,REVERSE)
  intake_motor.spin(FORWARD)
  goto(1400,1400)
  arm_motor.spin_to_position(1300, DEGREES)
  goto(1500,900)
  intake_motor.spin_for(REVERSE, 370, DEGREES)


  goto(-900,900,REVERSE)
  intake_motor.spin(FORWARD)
  goto(-1400,1400)
  arm_motor.spin_to_position(200, DEGREES, wait=False)
  goto(-1200,1200,REVERSE)
  goto(-300,450,REVERSE)
  drivetrain.turn_to_heading(-90, DEGREES)
  intake_motor.spin_for(REVERSE, 380, DEGREES)

  arm_motor.spin_to_position(1180, DEGREES, wait=False)
  goto(-600,600)
  intake_motor.spin(FORWARD)
  goto(-1400,1400)
  arm_motor.spin_to_position(1300, DEGREES)
  arm_motor.spin_to_position(200, DEGREES, wait=False)
  goto(-1200,1200,REVERSE)
  goto(-300,250,REVERSE)
  drivetrain.turn_to_heading(-90, DEGREES)
  intake_motor.spin_for(REVERSE, 380, DEGREES)

  arm_motor.spin_to_position(1180, DEGREES, wait=False)
  goto(-600,600)
  intake_motor.spin(FORWARD)
  goto(-1400,1400)
  arm_motor.spin_to_position(1300, DEGREES)
  arm_motor.spin_to_position(200, DEGREES, wait=False)
  goto(-1200,1200,REVERSE)
  goto(-300,50,REVERSE)
  drivetrain.turn_to_heading(-90, DEGREES)
  intake_motor.spin_for(REVERSE, 380, DEGREES)

  arm_motor.spin_to_position(1180, DEGREES, wait=False)
  goto(-600,-600)
  intake_motor.spin(FORWARD)
  goto(-1400,-1400)
  arm_motor.spin_to_position(1300, DEGREES)
  goto(1200,-1400)
  goto(1400,-900)
  intake_motor.spin_for(REVERSE, 360, DEGREES)

vr_thread(main)
```

DAY 9 - 07/09/2023

Present: Alex, Bao Nhi, Jack W, Jack L, Waynes, Hung, Aaron

Drivetrain day 6

- we used many collar locks on our drivetrain.
- An issue we had last yr w/ collar locks were that they loosen with vibration.
- For regular screws we can solve this by using nylock nuts. On the other hand, after doing some research on VEX forums, we found that most people agreed the best solution was loctite.
- we ordered some from China and it just arrived!
- Loctite acts similar to glue, but with enough force, it still can be taken apart.
- We spent today's session putting loctite on all the collar locks of our drivetrain.
  - we did this by:

DAY 10 - 09/09/2023

Present: Alex, Bao Nhi, Jack W, Jack L, Hung, Aaron, Wayne

Sleds day 3 (evaluation cont.)

- Our sleds were (sometimes) functional but kept breaking and didn't go over the barrier smoothly:

- **Problem:**
  - our dt sleds weren't really working and very fragile likely because of its shape:

sled hits in middle of barrier

- **Potential solution:**
  - we could instead make higher sleds tangent to the barrier?

easier to go over?

  - This can be done by attaching another C-channel on top of our dt:

versus the old design:

- **Potential solution #2**
  - using free-spinning spacers to make it smoother / more slippy!

Potential solution #1 + #2 → The ultimate sleds?? ^0.0

Day 10 cont.

- ~~Building potential sol. #2~~

Wings Day 3 (evaluation cont.)

- Problem
  - when pushing the triballs, the wings will bend backwards (
    - This is exaggerated after 1 min. of opening/closing when we go lower in pressure.
- Research / choosing best sol.
  - we think a locking mechanism would help with this

  - we really liked 21417A (Robokauz)'s wings locking mechanism, so we are hoping to do something similar:

Catapult Day 1

① Problem / constraints: + research (ish)

②

③
  - we wanted a fast catapult for shooting match loads over.
    - we think 33rpm is reasonable. We saw a lot of teams on youtube use 33rpm and we think this is a good speed.

③ Plan / Design

  - we are thinking about using a 12:36 gear ratio since it's small and compact:

  - When we tried CADing it out, we realized that the stacked motors take up space we needed for the cata
    - we're planning to fix this next session

DAY 11 - 11/09/2023

Present: Alex, Ben Nh, Jack W, Jack G, Aaron, Hung, Wayne

~~Cata~~ drivetrain day 7

· Problem:

· the stacked motors in our dt are taking up space we need for the cata:

· Solution Plan / Design

our old, stacked motors:

X = attach motor

new, proposed solution:

· Building - Images:

DAY 12 - 14/09/2023

Present: Bao Nhi, Jack W, Hung

Sleds day 4

- Today we wanted to try out potential solution #2 from last time (see pg. 34)
  - This was the simpler/quicker of the 2 so we want to try it first.
- Building / Creating:

- Testing: results
  - we noticed that the freespinning spacers also helped reduce friction for going over the barrier when they were at the back:

before

after

~~our~~ This did better for our testing

Free-spinning spacers somewhat help since less friction?

- ~~However~~, we were thinking of adding more:

DAY 13 - 15/09/2023

Present: Alex, Bao Nhi, Jack G, Jack W, Wayne, Aaron

Cata Day 2 - ratchet!

(problem)

- we needed a ratchet on our catapult in order to reduce strain on our motors.
  - A ratchet makes it so a gear can spin one way but not the other.

This allows the motor to spin the cata down while also helping counteract the pull of the rubber bands

- Normally, if the catapult is down, there are rubber bands that try to pull it up and the motor would have to hold it.
  - With a ratchet, on the other hand, the stress would be on the ratchet instead of motor.

- Potential solution #1: (plan)
  - making a screw perpendicular to the gear

Our prototype for potential solution #1 (build)

Results: (testing)

- It kind of works (+)
- But with enough force, it will still spin the other way (+)

- Potential solution #2: (plan)
  - inspired by 2654P
  - when spun anti-clockwise, the gear pushes the screw head up (spinnable)

- ~~clockwise~~ clockwise: gear pulls screw down + to the left
  - BUT the screw is tight and can't go to the right => very tight (can't spin)

Results: (testing)

(photo of prototype on next pg)

- it works but it's loose and shifted / fell apart very easily. If we were to implement this ratchet design, we would need to lock everything in place better.

Day 13 cont.

Photo of our prototype: (building)

That way, when the rubber bands try to pull on the catapult arm and spin the gear, the ratchet stops the slip gear and whole catapult

- We think that a ratchet is really important for our motors but we decided to prioritize building a working catapult first.
  - We will leave building a ratchet for after building the catapult!
  - However, we will be sure to keep these ideas in mind.

DAY 14 - 05/10/2023

Present: Wayne, Alex, Aaron, Jack W, Jack L (Hung and Bao Nhi at MUN conference)

Catapult day 2.5 (planning cont.)

- we identified the different parts we would need for our catapult:
  1. An arm to hold the triball
  2. A slip gear to free the arm to fling
  3. A ratchet to reduce motor strain when arm being pulled back
  4. Rubber bands to pull cata arm and fling
  5. Stopper to stop the arm at right angle (so it doesn't swing too much)
- Putting these together, we get something like this:

- Building / Prototyping:
  - we threw together a quick prototype

- This design feels really sloppy since there is stuff sticking out everywhere and no collar locks.
- It met most of the things we wanted to do though!
- However, we didn't finish the arm/bucket part yet. We hope to be able to attach those tomorrow!
  - A bit of preplanning for the arm/bucket:

DAY 15 - 06/10/2023

Present: Wayne, Alex, Jack L, Jack W, Aaron (Bao Nhi and Hung at MUN)

Catapult day 3

- building (cont):
  - we finished adding the arm and bucket, now our catapult looks something like this:

(we also added collar locks and cleaned up the issue w/ things sticking out everywhere)

- Testing:

we tested our design with an 11W 36:1 motor. Since the ratio is 12:36, the maximum speed our cata can go at is:

$$\underbrace{100 \text{ rpm}}_{\text{max speed of red motors}} \times \frac{12}{36} = 33.\bar{3} \text{ rpm}$$

Some problems: (evaluation)

- we need a proper and more rigid stopper. Currently, adding too much force onto the stop will cause it to bend.
- Parts of the slip gear "slipping" part still touches arm gear
  - This means we need to shave the slip gear better / smoothen it out
- Portions on the lower end of the design might slowly collapse in on itself if too many elastic bands are added
- High amounts of torsion will be added

DAY 16 - 10/10/2023

Present: Wayne, Alex, Jack L, Jack V, Aaron, Bao Nhi, Hung

Cata day 4 - Starting iteration 2.0

- Evaluation (cont): identifying the problem
  - Our 2nd iteration was planned out in response to things we learned from iteration 1.0:
    1. weak small shafts facing torsion
    2. Rigidity of catapult as a whole
    3. staying low-profile to go under goal

- Plan
  1. This new version is designed to be stronger than our prev. version.
     - To try and address 1, we started using HS shafts instead of regular.

- The frame was also modified too:
  - we made the frame of the catapult stronger
  - BEFORE: weak, regular shafts
  - we added screws with ~~zip~~ spacers and standoffs to reinforce the frame

    - The standoffs also acted as a stopper, which could make it weaker.
      - Shouldn't be much of a concern though since there are other standoff bars that make the frame more rigid

2. Height should be shorter
   - we did this by attaching the motor + ratchet to slip gear's axle instead of displacing it
   - we also wanted to ~~decrease~~ descore our opponents
   - Our modifications reduced ~~that~~ the height from 5.5" → 3.1"

(photos next pg)

Day 16 Cont.

← old: 5.5" height

new: 3.5" h →

· Plan Critique

· Issues with this design:

1. no proper way to connect + secure catapult to robot
2. no proper way to attach rubber bands
3. stop not well designed - won't absorb impact well
4. Frame not as strong as it would be
5. Catapult still too tall, cata very wide

· we felt that since it had so many issues, just adding fixes to each one would not be very efficient

· Therefore, we decided to come up with a new plan completely.

DAY 17 - (14/00/2023)

present: Alex, Aaron, Ben Nk, Jack L, Jack Y, Harry, Wayne

Cata day 5! (iteration 3.0)

· Identify the problem

· Our goals for this iteration:

1. Have a proper way to connect the robot
2. Have a proper ~~stop~~ stopper
3. Have a more rigid frame
4. Be smaller in size

· Plan

· The catapult can be attached to the robot like this:

· This connection was intentionally designed to withstand large impacts by being reinforced with spacers and screws

· WEAKNESS: c-channel at the back will be bent after many uses of the catapult

· This brace is also a stopper for the catapult.

· Thanks to the use of spacers and gussets, we were now feeling like the frame of the catapult in our design is much more rigid.

· We also changed the axle of the arm gear from a HS shaft to a screw joint

· This is because we feel like a screw joint is just as strong and at the same time also provides more security in the frame

↳ This also helps us with the collar locks hitting each other if we used HS shafts.

(images next pg)

Day 17 cont.

- In this iteration (3.0), the size of our catapult went from 3.5" tall to 2.5" :)
  - However, this means a longer robot (1.5').

- we achieved the decrease in height by arranging the gears horizontally instead of vertically

since we can't make the arm go flat when we put the gears like this,

we can use a 45° gusset to change the arm's angle:

- <u>Plan critique</u>
  - Some improvements we think could be made:
    1. Added sensors to provide more accurate feedback while operating
    2. Better + stronger ratchet mech.s to handle more rubber bands
    3. Decrease size, increase ~~size~~ frame strength
    4. A more structurally stable arm
    5. A designated place to string up elastic bands

DAY 18 - 15/10/2023

Present: Alex, Aaron, Bao Nhi, Jack L, Jack W, Hung, Wayne (over text)

Intake day 1 . identify the problem

· Identify the problem

· our intake needs these 2 functions for the first iteration:

1. eat and vomit triballs

2. holds triballs

· once we make a system that can do those things, we can add an additional function of:

1. moving the triball into the cata or a storage area on our bot

· build constraints:

1. it fits our robot and is below 5.6"

· this is so our robot can still fit below the goal and descore

2. it has to use 5 low worth of motors

DAY 19 - 17/10/2023

Present: Alex, Aaron, Bao Nhi, Jack W, Jack L, Henry, Wayne (online research)

Intake day 2 - research

- There are many ways we can build an intake, but we want to research more on:

  1. rollers 2. clamps & claws 3. Hooks

- Since we were on ~~winter~~ fall break, we couldn't meet in person. Therefore, Wayne helped research briefly all 3 types of intakes.

Rollers

Rolling intakes roll in objects from outside the robot to go into the robot. They grip onto the object and roll them from one position to another. There are many types of ways a roller can be assembled, and there are a myriad of materials that it can be made out of.

The rubber flex wheel is a strong option to use in a rolling intake. One aspect of a successful intake is the ability to consistently get a hold onto objects. The flex wheel provides an excellent surface to grip anything that comes into contact with it. Because of this, it becomes one of the better options to use for the intake.

In addition to having good grip, the flex wheel tensile. Relating to the previous point, good intakes need to get a good grip of their objects to control their movements. Due to the flex wheel's malleable quality, it can *mold* into the shape of the object being intake. This leads to an increase in surface area, and in consequence, more grip.

For any system to be good, it needs to have good longevity. In comparison to other options, flex wheels have a long life span. Due to their solid structure and tensile quality, the chances of them breaking during use is lowered. Hence replacing this intake is not required through most of the season.

However, this isn't to say that flex wheels are flawless. One issue that may be faced when using flex wheels is them coming out of their hubs, especially when they are smaller. In order to put the hubs into the flex wheel, you need to stretch them out, creating elastic tension in the center. In the case where the hub slips out, the flex wheel may decide to become loose.

Day 19 cont

One note about using flex wheels is its ability to be used for horizontally oriented intakes. Due to it being flexible, it maintains its structure while held side way, and can bend when enough force is applied on to it. It does not require any form of *hard* support around its circumference, unlike using sprockets and rubber bands.

All these factors make the rubber flex wheel a great option to use for the intake; it is grippy, malleable, and long lasting. It poses a good candidate to be used for a rolling intake

Sprockets and elastic bands (S&Eb) are another way of intaking objects with a rolling mechanism. As depicted in the illustration, the S&Eb roller suspends elastic bands through the teeth of the sprockets to create a roll of *strings*. These rows of elastic bands act similar to a tennis ball picker, when the object gets caught in the gaps and is reeled into the robot by the intake.

An aspect that gives the S&Eb an advantage over the other intake method is its ability to *hook* the triballs. In contrast to other methods, the S&Eb does not leverage surface contact to intake objects, but uses the elastic bands to *grab* the triballs which it effective at doing its job.

Another advantage to using a S&Eb mechanism the reduction of material usage in comparison to other methods. The stretched elastic bands allow it to cover a wider length with only two sprockets and elastic bands.

However, there are more disadvantages that come with using S&Eb in comparison to other methods. There are two main cons that this method is burdened with: the fragility and wear & tear of the elastic bands face and the elastic force induced by the rubber bands.

Constantly having to replace broken elastic bands is a task that is tedious and unsustainable. Having to replace rubber bands throughout practice and competitions will be time consuming, and may lead to us losing time to practice or to us missing our matches due to replacing rubber bands. Moreover, buying rubber bands due to a bad design decision is costly mistake. It unnecessarily wastes money and it causes more trash to be thrown or recycled. Hence this method may not be desirable for us to compete with.

In addition to breaking, the elastic bands will exert a compressive force onto the build, which will lead to additional problems. Due to rolling intakes having hinges, there is an increase of weak points in the system. Engineering the intake to be strong enough will require more effort than needed.

Therefore, S&Eb is a good method to intake triballs, but it comes a costs. It is an unsustainable waste of resources and it is harder to engineer. The S&Eb method will most likely not be chosen as an intake method in the future.

Day 19 cont

Most, if not all vex wheels can be used for a rolling intake. However, I shall only analyze the omni wheel. Unlike the others, the omni wheel does not seem to be fit for supporting an intake mechanism. It is very solid, it does not have as much grip as the others, and it is bulky. On the flip side, there is one gimmick to it that might be helpful: its ability to slide horizontally. In the case where we need to have a bottle neck section in the intake, the omni wheel can be used to make the process smoother.

When considering where the intake could be, it can be located anywhere in the front of the robot. In the illustration above, the blue highlighted section shows where the intake could possibly be. In addition to location, the rolling intake could be positioned vertically or horizontally. More options are also opened up when considering that the intake could be hinged in multiple locations at varying axis.

In the illustration above, it gives some brief ideas on how the intake could be positioned. There are many possibilities and designs that we can make; they can range from being simple to complex. Most of the designs above show a rather simple mechanism for a rolling intake.

Complexity is good, as it allows for multiple problems to be addressed at once. In the case of a rolling intake, we could increase the complexity of the design by adding the ability to extend and contract. This allows it to go under the low goal, which is one of our robot's criteria. In the illustration above, it shows an abstracted solution to achieve the goal of retracting the rolling intake. Although we may not want to achieve this in our robot, it serves to open up new doors for us to understand that we can increase the complexity of our intakes to achieve more than what can be done at surface level.

Day 19 Cont

Clamps & Claws

Clamps and claws latch onto the objects that they are trying to intake. When considering how to design such an intake, it seems to be fairly linear on how to design a claw; designs use two or more arms to pinch an object. However, there are many possible ways to make these claws. Following are some possible ideas and quick sketches on a few ways to make claws

There is not much to comment about such intake. It is a simple concept to grasp, but certainly a challenge to design. Some advantages that claws bring to the table are:

1. Providing an extended reach to the triballs.
2. Being a simple concept, allowing for more levels of complexity to be added.
3. It can come in many different shapes and forms to meet different size and shape constraints

However, the claw has some disadvantages which will require extra effort to overcome. For example, the claw lacks a lot of motion in different axes. This will make further stages of designing the intake harder. Another slight disadvantage to using a clamp is that it requires a motor to control its movements

One cool thing about using a claw is the ability to include compound functionality. For example, the claw can both grab the triball, but also eject the triball when it opens. Examples like this, similar to the complex systems discussed in the rolling intake section, shows that claws can also be made complex and more functional.

Day 10 cont

Hooks

Hooks operate in a similar manner to claws and clamps, but it does not have as many parts to it. In comparison to claws, it is most definitely simpler. I could only imagine two types of hooks, but I am sure that there are many more around. In the illustrations following, it shows the ideas I had for hooks.

It shows two very simple hook designs, one hinged bar hook, and a hook looking hook. Once again, hooks are very simple and carry some advantages. For example:

1. They do not necessarily require a motor to be powered.
2. They are very thin in design.
3. They are super customizable in shape to meet different size and shape constraints.

An issue that hooks face is that they are very static, which makes it require extra mechanisms to achieve the criterias that we have set for the intake. This will take up even more space, which will also make fulfilling the other criterias and constraints tougher.

DAY 20 - 23/10/2023

Present: Alex, Aaron, Dno, Niki, Jack W, Jack L, Wayne, Hung

Intake day 3

Planning - Iteration 1

idea #1

- combine the roller and claw mechanisms to make a really cool looking intake

minor idea/sketch:

simple claw intake

roller

rollers on side

easiest way to fit our requirements

simple motor connected to freespinning Arm

Day 20 cont.

Intake in CAD on the robot!

· Some potential problems we spotted:

1. Too tall for height constraint
2. flex wheels - very slim - intaking triballs ~~the~~ harder, less SA contact
3. c-channels coming into contact w/ triball before flex wheels
4. Flex wheels driven by motor (5.5w 200rpm too slow

5. motor on top coming into contact w/ catch → this system will most likely not work
6. Intake arm's screw joint and stand off screw heads will come into contact which is bad for the stability of the arm
7. There is no control ~~from~~ for keeping the intake from moving away due to it being free spinning
8. Triball is phasing through the catapult

DAY 21 - 25/10/2023

Present: Alex, Bao Nhi, Aaron, Jack L, Jack W, Hung, Wayne, Derek

Intake day 4

· Iteration 2.0

· This design was made to accomondate the height constraint of the robot. However, after some prototyping, we decided to scrap this idea. The intake couldn't roll backwards all the way to contract.

DAY 22 - 26/10/2023

Present: Alex, Bao Nhi, Aaron, Jack L, Jack W, Hung, Wayne, Derek

Intake day 5

· Intake iteration 3.0

· Based off 1st iteration's design

· Simple claw, fits our design constraint

· BUT very clunky and takes up a lot of space in our robot

· Testing results

· poor grip ⇒ we scrapped the design

DAY 22 - 30/10/2023

Present: Alex, Ben Nw, Aaron, Jack L, Jack W, Wayne, Ethan, Derek

Wings day 4

Jack L designed a wing with a thin stopper. when pushed from the wing, it won't close. However, it closes when pulled in by pneumatics.

the piston pushes out and opens the wings

For the lock (orange), we plan on using a long screw

using the collar lock allowed it to spin back and forth.

This helped with opening/closing ~~more smoothly~~ to lock and gave the screw a pivot point.

To connect the piston, we made a makeshift collar lock (since normal collar locks wouldn't fit) by drilling a spacer.

we built a prototype of 1 side to see if it would work.

The prototype worked!

We added this to both sides of our robot.

We now have wings!

DAY 24 - 31/10/2023

Present: Alex, Aaron, Bao Nhi, Wayne, Jack W, Jack C, Hung, Derek

Cata day 6

- since we were planning to do horizontal gearing, we looked online and stumbled across Robokauz's design again:

- we found this idea super interesting while also meeting some of the goals we want in our cata, so we tried to apply a similar concept.

  - CADED out, we get:

- we started by attaching the horizontal bars:

DAY 25 - 01/11/2023

Present: All (see prev. pg)

Cata day 7

- we finished building our cata so we spent today testing!

Testing

☆ # of slip teeth: 5 or 6 ↗ we put each slip gear in and tested

(we were debating about whether it was better to leave 5 or 6)

- 5:
  - slipping before it reaches bottom
- 6: works better than 5t, less slipping

☆ # of rubber bands (outside the robotics lab, not on vex field)

- we conducted this test by putting on the # of rubber bands and trying to ~~launch~~ use motor to launch a triball on a tiled floor, counting how many tiles.

# of rubber bands (per side)	# of tiles (including roll)	arc / notes (first contact w/ ground)	notes
2	10	2.5	low arc barely shooting
3	13	3	
4	20	4	
5	18	5.5	

- Testing results, processed:

- since the 4 rubber bands/side rolled the furthest, we decided to go with 4 per side.
  - However, it should be noted that 5 rubber bands/side has a longer arc. we will keep this in mind although a downside to 5 rubber bands would also be more strain/pressure on the motors.

FINAL DECISION

- we decided to go with a 6 tooth gear (6/12) for our slip gear since we found that it was the perfect amount through testing.
- we decided to use 4 rubber bands per side since we found it launched the triball the furthest (including rolling), which gave us a better chance to launch it into the goal

DAY 26 - 03/11/2023

Present: All

Cata day 8

we felt that 1 36:1 motor (2.1 Nm) wasn't enough for the catapult. we wanted FASTER.

Therefore, an option we considered is:

To make it faster, we could use an 18:1 gearbox for the motors instead

36 : 1 (11W)		18 : 1 (11W)
· 2.1 Nm		· 2.1 Nm
· 100 RPM	$\xrightarrow{\times 2}$	· 200 RPM

This means we would get double the speed for using the green motor (but only half the torque).

↳ $\frac{36}{18} \Rightarrow \frac{18}{36} = \frac{1}{2}$

∴ To make up for this, we can use 2 green motors.

Before we made this change though, we also wanted to improve the placement of the triball:

Problem:

> we kept placing the triball at different positions/angles which led to inconsistent shooting

Potential solution #1 suggested by: Bao Nh:

Adding rubber links at the end so we can align the triball

Pros:

- adjustable (we can try different gaps)
- easy to build

Cons:

- placement is still not precisely consistent
- we might not be able to align it quickly enough while the cata is retracting

Potential solution #2
suggested by: Jack L

we can do this by using the collar locks this way:

Day 26 cont.

Potential solution #2, cont.

Pros

- easy to build
- easy to place

also helps arm be more stable and maintain 90° angle

Cons

- still not 100% accurate
- still not really intuitive how the triball should be placed

Potential solution #3 suggested by: Alex

Pros:

- easier to align
- triball slides into place
- more intuitive on how to place triball

Cons:

- more resources (need to cut polycarb)

Final Decision (testing plan)

We will see which solution is most viable through testing next session, since we were running out of time today.

Our plan is to try out different solutions (from the above) and see which one(s) our team members find "easiest" / most intuitive

Testing the overall catapult

We tested our current catapult so far to see our progress. ~~(without motor, manually)~~

We found out that we can shoot into the goal cross court!

We did it manually (without running the motor)

Day 27 - 06/11/2023

Present: All

Cata Day 9

Adding guides:

We tested out the different potential guides we brainstormed last time:

we found out the ideal placement should be something like this:

Since the back of the triball is facing the far side, potential solution #1 would not be very viable:

however, we think the rubber links could still be useful to prevent the triball from rolling backwards when loading:

Therefore, from our testing, we found that using rubber links like this isn't ideal, especially for the placement we are going for.

we tried adding potential solutions #2 and #3 onto our catapult.

we tried to give both to all team members to see which one was easier to place. we came to a consensus that #3 was easier, since it was 3d, helping us guide the triball more during placement

However, we noticed that the two designs were on opposite sides, so we can incorporate both in our design:

the two ideas incorporated nicely and we decided to go with using both at the same time for maximum consistency!

Day 27 cont.

Therefore, we built the design incorporating both guides:

Adding a motor

We wanted to go back to the idea of adding a 2nd 11W motor to our catapult. We considered some pros / cons:

Pros

- more ~~torque~~, ~~con~~ speed, can shoot faster
  - this is good for shooting matchloads over the barrier quickly

Cons

- takes up an extra motor, this would mean our motor distribution is:

6 × 11 W	drivetrain	=	66 W
2 × 11 W	catapult	=	22 W
		→	88 W ← limit

- this means we reached the limit of motors we are allowed to use, which means we can't use motors for other subsystems like the intake.
  - maybe we could use pneumatics for the intake instead??

* Speed is really important for us since shooting matchloads is one of our main game strategies.

Day 27 Cont.

Therefore, we decided to add a 2nd motor, and change both to using the 18:1 gearbox.

Using the gear ratios, this should give us a speed of:

$$\frac{24}{24} * 200 = 200 \text{ RPM}$$

versus the old speed

$$\frac{24}{24} * 100 \text{ RPM} = 100 \text{ RPM}$$

This would be the speed that the slip gear spins, which is the amount of times that the catapult would shoot per minute.

200 RPM is $\frac{200 \text{ RPM}}{60 s} = 3.3 \frac{\text{RPM}}{s} \Rightarrow 3.3$ times/s

which was TOO FAST! So we could slow it down a bit by changing this gear ratio, which could also allow an increase in torque:

12:36 would give us

$$\frac{12}{36} * 200 \text{ RPM} = 66.6 \text{ RPM} \approx 1 \text{ time/s}$$

This was slower but more torque compared to the 1 36:1 motor, which allowed us to do cross-court shots like the video testing from last time (since we did it manually - the motor couldn't handle the rubber bands)

this could get us the AWP which could help us place in tournaments!

DAY 28 - 07/11/2023

Present: All + Derek ← New team member!

Cata day 10

we practiced shooting matchloads from the catapult, but it started bending:

For a temporary fix, we tried boxing the catapult where we think it was hitting/bending at:

point of contact where the arm was hitting the stopper

repeated force onto the stopper was causing the L-channel to bend

By boxing it, we could add more support on the L-channel

our catapult arm after we boxed the L-channels on the sides:

Day 28 cont.

Sleds day 5

(Problem)

Our current free-spinning spacers felt loose and flimsy, so we wanted to find some better fitting spacers.

After bringing our robot on the field and testing it for size, we found that we should add the spacers 1 stud ~~behind~~ below the back.

We weren't really sure how: (potential solutions + decision)

①

adding a plate wouldn't fit well with our drivetrain (gears in the way) and also felt extremely flimsy.

②

adding spacers and a 2nd c-channel was doable but complicated because there was a motor in the way:

this makes it tough for us to attach anything ~~to the left~~ (on the inside) of our dt.

③

We could use the collar locks' ability to create different angles:

this solution was easy to do and felt sturdier than the first one.

Therefore, we decided to try out the 3rd solution.

After lots of trial and error to find the right spacers, we got:

(Testing)

We tested this over the barrier and found that it went over the first part smoothly.

However, it still got stuck between the wheels:

So, we also wanted to add free-spinning spacers between the wheels.

We found bearing blocks which we think is the right shape for this.

Thus, we zipped some on (we didn't want to take apart the whole dt)

DAY 29 - 08/11/2023

Present: ~~Ah~~ Wayne

Cable Management

Today wayne helped do some cable management on our robot so far.

So clean!

hella fresh!

we also came up with new battery / pneumatic mounting:

very structural integrity 👍

The battery placement is a bit tough and unintuitive (difficult to figure out) but we think it's strong. we have a scrimmage coming up on 11/11 so we will focus on other things for now.

DAY 30 - 10/11/2023

Present: All

The Day before the scrimmage!

We did some driving practice at the start of the session:

Intake Day 5, Iteration 4

For the final design iteration before the scrimmage, we chose a hook intake powered by a cylinder. This was a rash decision made, but it worked the best in comparison to the previous three iterations. The image above shows the quick build that proves that the concept works.

One niche characteristic of this design is that it can be retracted before the match and dropped out when the match starts. It can be tucked into the catapult and will fold out when the catapult fires. However, the prototype had an issue as it came into collision with the battery wire.

To improve on the prototype, two things had had to be done:

1. Decrease the weight to allow for more uses (accommodate for psi drop)
2. Fold over the battery better

To work around this, the intake was designed to use polycarbonate as it was light and it fit the use. In addition to this, another set of hinges were added to the intake to allow for a two step bend. To deploy, the intake used elastic force to come down.

Day 30 cont

The completed intake, on the robot:

how it works:

Sleds day 6

Our sleds aren't working super well, so Alex designed some new ones:

keeping in mind what we discussed 09/09/2023 (sleds day 3, pg. 34), Alex designed and cut out the shark sleds:

Day 30 cont.

Programming Update!

we added programming for the wings and intake through pneumatics:

step ① we declared functions for the events of opening/closing:

For wings:

```
void wings_expand(){
  piston_wings.set(true);
}

void wings_retract(){
  piston_wings.set(false);
}
```

similarly, intake

```
void intake_open(){
  piston_intake.set(true);
}
void intake_close(){
  piston_intake.set(false);
}
```

~~Then, we used if/else statements in the main to de~~

Then, we used events to determine when a button was pressed/released

```
Controller1.buttonR2.pressed(wings_expand);

Controller1.buttonR2.released(wings_retract);
```

```
Controller1.buttonL1.pressed(intake_open);

Controller1.buttonL1.released(intake_close);
```

since we were using wings/intake/cata facing different directions, we thought a button to switch the orientation

- state 1: intake forwards (default)
- state 2: wings forwards

we did this by using a boolean variable "reversey" that is true when the orientation is reversed (state 1) and false when it's the other way (state 2)

```
void reverse_dir(){

  Controller1.Screen.clearScreen();

  reversey = !reversey;

  if (reversey){
    Controller1.Screen.setCursor(1,1);
    Controller1.Screen.print("front: cata/wings");
  }
  else{
    Controller1.Screen.setCursor(1,1);
    Controller1.Screen.print("front: sleds/intake ");
  }

}
```

Day 30 cont.

```
reversey = false;
Controller1.ButtonX.pressed(reverse_dir);
```

```
if (reversey){
  leftVal = -1 * (throttle - turn);
  rightVal = -1 * (throttle + turn);
}
```

Cata Day 11

We continued shooting the cata today but noticed it was still bending.

The bending was considerably less than before (thanks to the boxing). However, we didn't want to risk it so we added some dampening.

It was the day before competition so we didn't have much time to research on this matter.

Therefore, we just wrapped a bit of grip mat around the standoff:

we couldn't make the dampening too thick though, since it ~~was~~ could affect our launch angle.

we will definetly keep dampening in consideration for our next iteration.

Final CAD of our cata (dampening + guides not included):

FLiP!

THERE'S STUFF UNDER!

RGB UPDATE!!! (& explanation)

Entry by: Alex (skills programmer)

Neopixels!

Something we are very proud of in our robot is the RGB underglow, which is something Alex has wanted to do since he started VRC (inspired by 5090X). Even though the coach says "it's a waste of time", at least we lose in style (also it *totally* makes our bot faster). There are two types of RGB strips:

- traditional strips with 4 pads, one for each color (red, green and blue) and one for ground with the input as varying voltages
- neopixels with 3 pads that use a PWM signal which is a rapid change of a digital signal.

The VEX brain can only output a digital signal so it can't drive the 4 pad led strip directly. However, the old 393 cortex motors also needed an analog signal to function. So, it used the 29 Motor controller– which converts a PWM signal into an analog signal. The issue (and the reason we never did this) was we needed 3 different motor controllers for each color, which takes up a lot of space on the robot. Additionally, without additional electronics, the motor controller does not have a current cap. So, if we accidentally flash the lights too bright (33 leds at max brightness), we might fry the brain's adi port, and the motor controller is also illegal this season. The other method is using an addressable WS2812 led strip, where we use a PWM signal to address each individual pixel. Neopixels use a different protocol of PWM than the one VEX supports– we can overcome this by implementing this protocol by using the digital out function. The VEXos kernel makes it so that all user programs are run on the CPU1 thread, which is a low priority thread thathas a slow cycle time. This is normally fine for most tasks, but neopixels take a 800kHz signal, or a change every 1.25 microseconds, taking 8 bits per color per pixel – meaning 24 bits per pixel. The led would be very slow changing with the CPU1 thread, especially with animations. However, just this year, James Pearman helped implement the special neopixel protocol into SDK 20220726 as a built-in function (since built-in functions run on the CPU0 thread). We immediately started trying to add RGB onto our robot when we saw that, and it makes our robot look very special. We use one strip which is soldered onto a cut 3 wire extension cable that goes around the robot, which we also use power injection on the end to ensure the same brightness across the whole strip. For the animation we tested several methods, but we stuck to the simple method of linear interpolating between two colors to form a gradient– which finds colors in between using by finding the difference between each channel of the colors, then dividing it by the amount of pixels, and then incrementally adding this to each neopixel. We then use the std::rotate function to move the vector index pointer, which moves each pixel to the right with each specified interval.

SCRIMMAGE!

11/11/2023

Present: Jack L, Bao Nhi, Aaron, Jack W, Dung (Alex), Derek, Hung (left early), Wayne

Today was our first scrimmage of the over-under season!

Alex programmed some RGB for us last night and attached the strips on the base of our robot.

They are purely for decoration!

unfortunately, we got eliminated our first round in the elim matches

we placed last overall, but learned a lot of things today!

(0W / 4L)

Scrimmage cont.

Reflection (evaluate + identify the problem)

Catapult

After some matches, our catapult severely bent, and the metal snapped!

We still had an elimination match to go, so we tried zip-tying the catapult arm back together:

This brings us to things we need to improve on our catapult.

- increasing the rigidity of our bucket + stop
- add better dampening to stopper
- lengthen arm / bucket for more leverage?
- add a limit switch for resetting - better for auton?

something we failed to consider in the beginning

We felt that since catapults had a moving arm, it makes it difficult for us to align the triball / place it (since we have to place it on a moving part)

Therefore, we should be open to considering other potential shooter designs, such as:

- flydick
- puncher
- linear puncher
- flywheel / rollers

Another problem we faced was being blocked / pushed.

Team 1599N used their ~~[illegible]~~ catapul (that wasn't working properly) as a blocker.

Therefore, after the scrimmage, we learned that we should go for a higher arc instead.

Although this ~~do~~ makes AWP (cross-court shooting) tougher, we think it is worth it since we can shoot match loads more consistently.

~~Out~~

Wings

our wings were also being inconsistent, some things we want to fix:

- have the locking mechanism lock properly
  - double acting pistons?
  - new design?
  - not have it be able to poke so much that it gets the dt stuck
  - also make it lock onto another thing so that it won't wear and tear the dt bearing zipties

(our wings jammed our drivetrain in 1 match)

Scrimmage Cont.

- heighten the wings to push more reliably

Intake

we noticed that a lot of the taller intakes did well since they were able to easily capture the triballs

rubber bands + sprockets seemed pretty common and worked well

we think that a retractable intake could be useful for keeping a low-profile bot:

Also, since we didn't keep the sleds in mind while designing our intake, our intake made it more difficult for us to cross the barrier.

Also, it's difficult since we need to "aim"

So, some of our notes on the intake:

- have it retractable
- currently, it is too flimsy - Aggresive plays may break the intake
- Change the intake system to using motors (for ease) → easier to drive/control, less need to "aim"

Sleds

Crossing the barrier was pretty inconsistent. Some ideas/notes from the scrimmage:

- stop the middle from grinding the bar (still getting stuck between wheels)
- add something to the back of the ~~wheel~~ dt (like a wheel) to move

Drivetrain

~~Decrease~~ During this scrimmage, we were also being pushed a lot.

A problem we encountered was when team N pushed us into the match load bar (while we were loading), our robot got stuck on the bar (between the wheels)

something like our dt getting stuck on barrier

And when we were pushed a second time, our ~~sc~~ brain screen turned blank (white screen)

we think this is because the battery disconnected for a short moment.

So, some things we could change / some notes:

- Decrease the spacing between wheels?
- Change the battery mount so we don't get white screen
- Find a more secure battery placement, safer places for cable connection

DAY 31 - 13/11/2023

Present: All (see prev)

Post-Scrimmage Reflection (evaluate + identify the problem) + brainstorm solutions

we discussed some potential solutions to problems we were encountering during the scrimmage and our next steps ~~for~~ to prepare for FORMOSA on Dec 2-3.

Drivetrain

A problem we have with our current drivetrain was that it is difficult for us to cross the barrier.

This is an advantage to having more wheels that we couldn't see before, the ~~space between the wheels becomes~~ robot becomes longer, making it easier to go over the barrier:

⇒ longer robot!

Therefore, for our next iteration of the drivetrain, we want to change it to 8 wheels instead of 6. we can do this by adding traction wheels in the middle of each side:

The gearing remains the same,

Also allows us to place the motors flat!

A big robot will also give us more space to build ~~both~~ the subsystems, which is something we felt like we needed after the scrimmage and our main reason for making this change.

Shooting Mechanism (Cata?)

we looked at different shooting mechanisms. Since our shooting mech's main purpose ~~was~~ is match-loading, we felt like a catapult was difficult since we had to load it on a moving arm. suggested ideas from discussion:

	Pros	Cons
① Catapult (current)	· adjustable arc · somewhat consistent	· needs lots of torque (motor) · difficult placement (moving arm) · stress (structure) · rubber bands
② Puncher (angular)	· easier placement · small size	Same w/ ~~cata~~ but easier placement
③ Roller (flywheel)	· not limited by tempo · no stress	· RPM drop · lots of space · spin up delay
④ Flydic	· small/simple	· practice heavy · rpm drops
⑤ linear puncher	· good horizontal · less arm stress	· easy to block · space · tempo · slow reload · rubber bands

Day 31 cont.

The 3 options we wanted to look further into for our shooting mechanism were catapults, punchers (angular), and rollers (flywheels) (revisiting ~~a~~ shooting mechs!)

① Catapult (current)

· For catapults, we would do something similar to our current one or keep it and make improvements (such as better dampening)

this time, we would put dampening into consideration while figuring out launch angle

· However, something we ~~could~~ didn't like about catapults were that we had to load the triballs onto a moving arm.

② Angular Puncher (inspiration: speed of light)

Something similar to a catapult, except it's shooting from a platform

platform is still, making it easier to place triballs

we like the puncher because it's like a catapult except it solves the issue we had with the catapult (moving arm)

③ Flywheel (Roller)

We saw a lot of roller flywheels during the scrimmage:

Another type of flywheel is "fly die", which is a flex wheel spinning really fast

Something that we don't like about triballs is that they take a while to "spin up" and reach max RPM - and we had to wait. This was a problem we found with our ~~tri~~ flywheel last season (22-23 Spin-up)

Also, the RPM of the flywheel might drop after every time it shoots, leading to inconsistent results. Therefore, knowing the timing between shots so that it can spin-up again would be important for loading a flywheel.

Day 31 cont

we used a decision matrix to decide between the 3 and got that an angular puncher would be ideal.

Thus, we decided we will be building an angular puncher for our new ~~tou~~ shooting mechanism.

Since there were time constraints (formosa coming Dec), we will try to finish this in 7 days.

shooting mech decision matrix - 13/11/2023

	Weight (/5)	Catapult		Angular Puncher		Flywheel	
		Rating (/5)	Weighted Score	Rating (/5)	Weighted Score	Rating (/5)	Weighted Score
Consistency	5	4	20	4	20	3	15
Replaements (e.g. rubber bands)	2	2	4	2	4	5	10
Arc	4	5	20	5	20	2	8
Customizability	3	5	15	4	12	2	6
Sizing	3	2	6	4	12	2	6
Ease of launching	4	1	4	4	16	3	12
Stress	2	2	4	2	4	1	2
Ease of building	1	3	3	4	4	2	2
Tempo	2	3	6	3	6	5	10
Total Score			82		98		71

Intake

① Top roller

For this type of roller, we could use:

- rubber bands + sprockets

- flex wheels

- mesh wrapped around rubber bands / sprocket

one of the 3 above

this intake was pretty common during the scrimmage.

A benefit of this intake is that it is easy to maneuver - as soon as it comes into contact with a triball, we "own" it.

② side rollers

r(1599R)

During the scrimmage, we saw team R have this intake. It was really percise/accurate.

However, we think that this intake is tough to drive since we need to get the triball into the small space in between the flex wheels for it to go in

③ Claw

we didn't really like claws since it was hard to grip, aim, and also really slow.

Day 31 cont.

Decision Matrix for Intake - 13/11/2023

	Weight (/5)	Top Roller		Side Rollers		Claw	
		Rating (/5)	Weighted Score	Rating (/5)	Weighted Score	Rating (/5)	Weighted Score
precision	4	3	12	5	20	5	20
ease of intaking	5	5	25	3	15	1	5
grip	4	5	20	5	20	3	12
ease of building	1	4	4	3	3	5	5
range	3	5	15	2	6	2	6
Total Score			76		64		48

We used a decision matrix and got top rollers.

We now need to decide whether we will be using rubber bands + sprockets, flex wheels, or mesh.

Climbing

our current motor distribution plan is:

drivetrain	6	×	11 w	=	66 w
Shooting	1	×	11 w	=	11 w
intake	1	×	11 w	=	11 w
					88 w

This means we don't have any motors left for the endgame.

We were thinking of using the wings to balance instead

Schedule + Time Constraints

Since we have Formosa coming up soon, our plan for finishing the new robot:

Task	Due Date
8w drivetrain + new sleds	Mon 20/11
Puncher	Mon 20/11
Intake	Fri 24/11
wings	Mon 20/11

Also, for programming, we decided that Bao Nhi will work on the head-to-head program while ~~Dy~~ Alex will work on skills.

← leaving a week for driver practice

bnhi | 1599V | 76209R 11/11/2023 6:31 PM
hey guys do u have any tips for bent metal in catas? my team's cata broke in the middle of comp today. probably from the repeated impact on the stopper making it bend and split. any advice at all would be appreciated, tq in advance :D

my team is thinking abt changing to 2-wide c-channels (instead of the 1x1 angle bar) and using steel instead of aluminum

Sohail | 2915E 11/11/2023 6:57 PM
2 wide aluminium should be fine
boxed 1x1 would also probably work
gonna assume you use dampening

@bnhi | 1599V | 76209R my team is thinking abt changing to 2-wide c-channels (instead of the 1x1 angle bar) and using steel
Luke || gcec 2918D 11/11/2023 6:59 PM
It's interesting that it broke on the bolt hole
Dose the cata stop on the bolt

@Luke || gcec 2918D Dose the cata stop on the bolt
bnhi | 1599V | 76209R 11/11/2023 7:09 PM
yeah i think the stopper is hitting at that part

@Sohail | 2915E gonna assume you use dampening
bnhi | 1599V | 76209R 11/11/2023 7:10 PM
any suggestions on what to use for dampening? we just wrapped some anti-slip mat around the stopper, idk if it's helping much
our stopper is currently a long standoff going across btw

@bnhi | 1599V | 76209R any suggestions on what to use for dampening? we just wrapped some anti-slip mat around the stop
Luke || gcec 2918D 11/11/2023 7:10 PM
You got a spare flex wheel

bnhi | 1599V | 76209R 11/11/2023 7:11 PM
oooh woah
thanks

@bnhi | 1599V | 76209R it was an angle bar
Sohail | 2915E 11/11/2023 7:13 PM
but yea boxed 1x1 plus traction tire dampening should be fine for ur cata

Sohail | 2915E 11/11/2023 7:18 PM

this stuff is great

Sohail | 2915E 11/11/2023 7:18 PM
It's like the best we tested
out of like 6 things

Luke || gcec 2918D 11/11/2023 7:18 PM
Do you use them inside out or the right way

@Sohail | 2915E It's like the best we tested
Luke || gcec 2918D 11/11/2023 7:19 PM
Yeah for the edge it would work alright

Sohail | 2915E 11/11/2023 7:19 PM
yea
which is every puncher and cata

After the scrimmage, Bao Nhi also asked a vex community on discord about dampening.

They suggested cutting green traction wheels, which ~~is definetly~~ we will definetly try to incorporate into our puncher so the metal doesn't snap again

we also decided to switch to a tall bot, since we ~~th~~ didn't see a lot of double zoning in our games (our opponents saw our bot and didn't double-zone), and we think there are more advantages to a tall bot.

DAY 32 - 14/11/2023

Present: All (online research)

we did some research and looked at teams on Youtube that used a top roller. we put together a Google Doc (shared document) on some of our findings:

Intake Day 6 (research)

Qualities of a good top roller intake

Good qualities	Examples
High RPMs	All most all teams try to achieve high RPM intakes when using any type of rollers
Extreme range of floating	Robokauz Vex Over Under Cornerstone Tournament Elims 2...
Wide intake range	Self explanatory: more space ⇒ easier to aim intake helps in auton, prog skills, and drive control.
Chain driven	Rocky Pits & Parts | 57711X Rocky | Over Under | FUN R... Hard Wired Pits & Parts | 6277C Hard Wired | Over Under Rob...
Collapsable	Hard Wired Pits & Parts | 6277C Hard Wired | Over Under Rob...
Contractible	Wigglebots Pits & Parts | 8926W Wigglebots | Over Under | F... Ace Robotics 229V VEX OVER UNDER #1 WORLD SKILLS 326
Chain Redundancy	Wigglebots Pits & Parts | 8926W Wigglebots | Over Under | F...

Comparing Intake Styles

	Extreme float range	Contractible
Pros	+ Simple mechanism to build + works effectively when being bashed by other robot + less joints required	+ it can extend further to increase efficiency + can receive match load for auton + "more" space for other systems
Cons	- has to be built at a relatively tall level. - limited to the robot's starting size. - Might have tension issues?	- is more complex / has more joints (weak points) - requires an unloading mechanism which takes us space - to retract when playing defensively will increase complexity
Evaluation	For the simpler mechanism, using an intake with an extreme float range will be much better than making a contractible intake. This is mainly due to: • 1 week time constraint • Its simplicity in design • Lack of joints	

DAY 33 - 15/11/2023

Present: All

Intake Day 7

After researching, we went back to looking at what material / parts to make our top roller out of. Our options included

① Rubber bands + sprockets (insp. 21417A)

- we don't really like this option since we used it for our first intake last season. we found that the rubber bands fell out more, and also the tension from the rubber bands pulls the sprockets together.

A couple issues with this type of roller:

1.

2.

3.

② Flex wheels

- this is what we ended up using for our intake last season.

③ mesh around the rubber bands + sprockets

- same disadvantages with ①

We used a decision matrix to decide between the 3:

Intake material decision matrix - 15/11/2023.

	Weight (/5)	rubber bands + sprockets		flex wheels		mesh	
		Rating (/5)	Weighted Score	Rating (/5)	Weighted Score	Rating (/5)	Weighted Score
wear & tear	3	3	9	5	15	2	6
grip	5	3	15	5	25	4	20
ability to "hook"	4	5	20	2	8	3	12
malleability	2	4	8	5	10	3	6
replacing between matches	3	3	9	5	15	1	3
Total Score			61		73		47

Day 33 cont.

Intake Designing and CAD modeling

After researching about top rollers (pg 78) and deciding on using flex wheels, here are some of the design concepts that we came up with for our intake:

For the first concept, we came up with an intake that was mounted in the far front of the robot. It used multiple c channels and standoffs.

One issue was that the motors were put on the inside, which will touch the triball when it gets into the intake. Moreover, the intake's sides are very wide, wasting a lot of space.

For this second intake design, we put the motors on the outside of the intake. In addition with this, we reduced the amount of c-channels used.

Another element was added, which were the intake sleds; this allows for the intake to smoothly open when it comes into contact with any solid object in front of it. This aids with pushing the ~~intake~~ triballs into the goals and protecting the intake if the front of our robot gets hit.

DAY 34 - 16/11/2023

Present: All

Drivetrain day 8

Planning / CAD

we started by ~~cad~~ CADing out what ~~an 8w~~ our new drivetrain would look like, based on decisions we made previously (pg. 73)

Building!

We worked together to build the 8w (new) drivetrain, based on the CAD.

It was really similar to our previous drivetrain, except for a 30wide c-channel and the addition of a traction wheel in the middle.

We also moved the 3rd motor (compared to old design) down since there was space to put it there.

Skills Auton Update - PID explanation

Entry by: Alex (Skills programmer)

Edited by: Bao Nhi (H2H programmer)

The simplest method of autonomous movement is Dead Reckoning, but it has no feedback and is very unreliable. An example is given that our robot travels 6 in/s with 6 volts applied, if we want to move forward 6 in, we need to apply 6 volts for 1 second. There are a lot of reasons why this wouldn't work, the largest being acceleration– the robot won't reach the speed immediately (it needs time to accelerate) so it will travel less than calculated.

An improved version can take acceleration into account. Even then though, it still has many problems:

- A motor will not always spin the same speed with the same input voltage
    - This is because the battery level alters the amperage of the motor
- The friction will also differ on different floor surfaces

Therefore, this algorithm is very rarely used.

The way to measure the unpredictable factors is odometry. The simplest odometry uses encoders with the V5 motors. The encoders of the V5 motors act as a rotation measure for the motor that outputs a square wave.

A tachometer encoder has a disk with many holes for light to shine through. Then, optical sensors measure the speed of oscillation.

There are also two optical sensors whose pulses can be compared (to get the direction) by seeing which pulse goes high first. This means encoders can help give us feedback on how much we have to spin the motors.

A better way of dead reckoning is to go forward until the encoder spins the desired amount. An issue, however, is the sensors might never output 0.00 – so we can't set a dead band by `abs(desired error) < 0.1`. The issue is this will overshoot, because when we stop, the wheels will still have momentum. This can be reduced by short braking, which short circuits the motor by a ShuntFET across windings of the motor. However, this will not always stop immediately, and strains the motor. We can solve this by slowing down when near the target, which is called a

P-controller (P stands for proportionate).

```
Unset
desired = 6 //in
desired = desired / 3.25 * 2 * M_PI //rad
initial = motor.encoder(rad)
while((desired - (inital - motor.encoder (rad))) = 0):
        drive.spin()
```

P - Proportionate

In a P-controller, the speed is proportionate to the remaining distance, so the robot will slow down when near the desired location. An additional value called kP is proportional gain, which is a constant that we get by turning. It decides at which distance the robot will start slowing down. If it's before that instant, it will output more than the motor's max speed. Therefore, we clamp the value to ± 12 volts.

```
Unset
desired = 6 #in
while True:
        error = desired - current
        speed = error * kP
        speed = clamp(speed, -12, 12)
        drive.move(speed)
```

This is a simple controller that works well, but there are many things that can still be improved. When approaching the target, the error of a P controller will always be a non-zero value. This is because the controller outputs such a low value that the robot doesn't overcome friction. We can solve this with an integral term.

I - Integral

By definition, an integral is the area under a curve. In this case, we added the sums of the instantaneous errors over time, giving us the accumulated offset that should've been corrected previously. The accumulated error is multiplied by the integral gain (K_i), and then added to the controller output.

By accumulating the integral of past errors, the integral adds a correction that grows over time in response to the persistent steady-state error. This eliminates the steady-state error while reducing settling time.

```python
integral = integral + error
speed = (error * kP) + (integral * kI)

if abs(error) <= 1:
        integral = 0
if integral > big_balls:
        integral = 0
```

There are two issues to this approach:

1. When the error is 0, the integral is still non-zero, which means the speed is non-zero too. This is mitigated by setting the integral to 0 if the error is within the accepted values.
2. Integral wind up: when going a far distance and when approaching the set point, the integral is unusably high. So, we set the integral to 0 every time it reaches a high value.

An issue with a PI loop is oscillation and rapid unpredicted changes, which is helped with a derivative term.

D - Derivative

Derivative is the instantaneous rate of change, or the slope of the line tangent to the curve at that point. We used a derivative to dampen the robot's movements based on what is expected to occur. It will be a signal with the direction opposite to the direction of travel, with a greater magnitude for greater speed. It will typically be outweighed by the proportional and integral components. But if there is some deviation from normal, the derivative will compensate. It is calculated by measuring the difference between the previous error and current error.

```python
derivative = error - preverror
preverror = error

power = (error * kP) + (integral * kI) + (derivative * kD)
```

Implementation

The simplest way to implement PID is by calculating the amount the wheels need to spin (to go a certain distance), and then comparing it with the motor encoder value. We can calculate the amount needed for turning by using the arc length formula, $s = r\theta$ (r being half of track width or distance between the tracking wheel and tracking center), which gives the distance to move each wheel. We can use the arc length formula again to calculate the angle we need to spin each wheel. We turn a certain amount of angle, then we drive a certain amount of distance.

We did this last season with a custom tuned PID loop. Although the most common way to do it is on the built in Cortex M0 on each individual motor, this chip is very slow. So, we bypass it and send a direct PWM signal to each motor – for more accuracy. An issue with this method is if we're pushed by another robot between motions, the motion becomes inaccurate. Even with this issue, this implementation really helped us during worlds, because the fields were sinking in the middle (due to the fields being elevated). This ruined a lot of people's autonomouses, but our PID worked well.

Day 34 cont.

Intake Day 8

Prototyping and Building

This was the first build based off ~~off~~ the concept designs made in the CAD.

However, a few changes were made with the hinge point being lower to the ground. This made it so that the intake doesn't have to be suspended up high.

The main issue with this design was the triballs touching the intake chain and axle, due to it being close to the end of the intake.

Examples of intake interfering with triballs

we fixed this by moving the motors back along with the gears.

This intake composition allows the triball to be collected without much interference

With this, the only missing part is the polycarbonate intake sleds.

DAY 35 - 17/11/2023

Present: All

Puncher day 1

(Identify the Problem)

Goals for our puncher:

- matchload the triballs
- Get it across the barrier/bridge and (the triballs) near the goal so our alliance partner can push them in
- be short enough so we can still go under the elevation bar

what do we consider success:

- fast enough to get as much match loads in
- consistent / accounts for human error
- easily repairable / easy maitenance
- automated / sensors
- strong

we looked at different gear ratios:

Motor RPM	Torque (Nm)	Efficiency Constant (%)
100	2.1	60

Gears			Impossible		W/ Eff Constant	
Input Gear	Output Gear	Gear Ratio	RPM output	Torque Output	RPM Output	Torque Output
(a) 12	36	0.3333333333	33.33333333	6.3	20	3.78
(b) 12	24	0.5	50	4.2	30	2.52
(c) 24	48	0.5	50	4.2	30	2.52
(d) 36	48	0.75	75	2.8	45	1.68
(e) 36	60	0.6	60	3.5	36	2.1
(f) 36	84	0.4285714286	42.85714286	4.9	25.71428571	2.94
(g) 48	60	0.8	80	2.625	48	1.575
(h) 60	84	0.7142857143	71.42857143	2.94	42.85714286	1.764

Some notes:

- 12t has steel gear
- Skills - 44 triballs
  Head-to-Head - 22 triballs
- ~~In skills, only gear ratios (b) and (c) may succeed in getting all the match loads into the field in time~~
- gear ratio (a) looks worst on paper due to its slow speed, which is insufficient for shooting all the match loads onto the field
- In a skills match, only (b) and (c) may succeed in getting all the match loads into the field on time.

Day 35 cont.

Automation and sensors

· To automate functions, we should use color vision sensors to detect triballs:

and also a limit switch to calibrate the puncher arm:

Wings Day 5

After the scrimmage, we want to switch to a more robust wing mechanism. Alex designed new wings that were more flush with the drivetrain - reducing chance of damage.

We also tried a new technique of using HS shafts, they are made of steel, so very strong for their size. Drilling holes through the shafts are very tedious (we need to do this to put screws through though...). Also, the holes have to be closer to the center as the bit is barely smaller than the shaft.

We CADed out this wing concept:

DAY 36 - 18/11/2023

Present: Alex, Jack L

Wings Day 6

we finished building a prototype of the wings we designed last time!

The prototype worked well, and we could clearly feel that it was stronger and more solid than our previous iteration.

We replicated this for the other side (LHS) and added both onto the robot (implementation plan on CAD on pg. 84)

Wedges Day 1

We want to add wedges in addition to our wings so that we can push triballs over the barrier.

A wedge should look something like this:

We brainstormed some ways to attach a plate at an angle:

① zip-ties

we could zip tie the plate onto our robot

Pros:
- easy

Cons:
- can slip
- easy to break

② angle couplers

we could use angle couplers to form an angle

Pros
- holds angle well

Cons
- bendable:
- not very adjustable

we wanted to add polycarb to expand the area contact w the tribal

③ hinges

Pros
- can pull up if we're overize

Cons
- might be too floppy:
  → we can add support though

Decision matrix

	①	②	③
pros	2	3	5
cons	-4	-2	-1
	-2	1	4

Final decision: Hinges!

A CAD concept we designed for wedges with hinges

← Testing out hinges (we only used 1 in the middle)

we changed to plates instead of bars (CAD) since plates felt more stable

Final version

2 hinges, screwed on →

Skills Auton Update - Odometry

Entry by: Alex (Skills programmer)

To fix the previously mentioned problems, we experimented with a new concept called APS. It was taught to us by Skills World Champions πlons, whom we met during worlds. The *absolute positioning system* is a concept of always giving directions to the robot, relative to the field. This means the robot will be able to account for error in any direction, even when there is no movement. At the core of this system is odometry– which keeps track of the robot's absolute position and heading in cartesian coordinates using encoders. We will use the motor encoders, but we also have plans for using external tracking wheels– wheels that are rubber banded so that they push towards the floor to ensure no wheel slip. However, we are still waiting for rotational sensors to ship, so we will use motor encoders for now.

Some definitions:

ΔS tracking center movement

ΔR right wheel movement

ΔL left wheel movement

$\Delta\theta$ change in heading

Δx change in x

Δy change in y

s tracking center to wheel (constant)

r_A tracking center radius

r_R right wheel radius

r_L left wheel radius

There are a few important points to realize from this diagram. Firstly, the arc angle is the same as the change in orientation of the robot.

This angle can easily be calculated by starting with the arc length formula, rearranging, then plugging the two functions into each other.

$$\Delta L = r_L \Delta\theta \qquad \Delta R = r_R \Delta\theta$$

$$\Delta L = (r_A + s)\,\Delta\theta \qquad \Delta R = (r_A - s)\,\Delta\theta$$

$$\frac{\Delta L}{\Delta\theta} = r_A + s \qquad \frac{\Delta R}{\Delta\theta} = r_A - s$$

$$r_A = \frac{\Delta L}{\Delta\theta} - s \qquad r_A = \frac{\Delta R}{\Delta\theta} + s$$

$$\frac{\Delta L}{\Delta\theta} - s = \frac{\Delta R}{\Delta\theta} + s$$

$$\Delta L - s * \Delta\theta = \Delta R + s * \Delta\theta$$

$$\Delta L - \Delta R = \Delta\theta * 2s$$

$$\Delta\theta = \frac{\Delta L - \Delta R}{2s}$$

We can then figure out the amount moved relative to the field using trigonometry. We pretend that the movement is a straight line (because the loop is very fast), then we use cosine and sine (because we know the hypotenuse = average travel between two wheels) to get the opposite and adjacent sides. We then get the relative position by adding this to the previously known coordinate to get the current coordinate.

$$\Delta S = \frac{\Delta L - \Delta R}{2}$$

$$\Delta x = \Delta S * cos(\Delta\theta)$$

$$\Delta y = \Delta S * sin(\Delta\theta)$$

$$p' = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{pmatrix}$$

We did testing today and found the motor encoders to be quite accurate and drift very little if we stay under 280 rpm. However, the turning odometry was quite inaccurate, so we decided to use the gyroscope. We then use a Kalman filter, which uses a prediction model to estimate probable values (using previous values, maximum velocity, and acceleration). Then, we compare it to sensor values (odometry and gyroscope) which gives us the most likely value.

We also tested GPS, which triangulates the position of the robot using a camera and GPS strips. We saw it was very inaccurate when moving. It has a very low refresh rate, but it could be useful for starting a match (sometimes we place the robot wrong ⇒ movement is off for the whole match). Therefore, we will use GPS for getting the initial position.

DAY 37 - 20/11/2023

PROJECT: 41

Intake Day 9

Today we tested running the intake and put it on our robot. The intake worked!

↙ us in the process of adding our intake onto the robot

Puncher Day 2

For iteration 1, we made a puncher design based off our research. However, there are many flaws with this design.

1. The puncher does not hit the triball on its flatter surface:

2. There is no designated rubberband mount
3. The connection to the drive base is very poor
4. There is no ratchet to keep the arm down

Overall, ~~the~~ d this design has to be redone to be tournament worthy.

we went with ~~24:48~~ 24:48 since ~~it~~ it was 1 of the 2 options that was fast enough (see pg. 89) and 12:24 was too small (we couldn't attach the c-channel on it)

DAY 38 - 23/11/2023

Puncher v. 2

Present: All

Autonomous Day 1 (Bao Nhi, Jack W)

Today Jack W did some research watching videos of different teams' autonomouses to compile one that was compatible with our robot.

We wanted to score at least 1 triball for both sides. we also wanted to score as many as we can without making it too complicated.

① release triball

1 triball auton for defense

① release triball

② pick up triball

③ open wings

(hopefully) 4 triball auton for offense

Puncher Day 3 (everyone else)

For iteration 2, we fixed a few aspects of our puncher to cover the faults of the previous iteration

1. The triball has a designated plate to be put on
2. The connection to the drive base is wider

However, there isn't a rubber band mount / ratchet on our puncher.

After testing this design in real life, using a 24:48 ratio doesn't seem to work compared to a 12:36. Also, another factor for motor heat up may be the placement of rubber bands.

DAY 39 - 25/11/2023

Present: All

Puncher Day 4

For our iteration 3 design, we try to cover the previous flaws of our 1st and 2nd iterations.

We tested our iteration 2 puncher again and here are some notes we took: (iteration 2 + test + evaluate)

- it has a sturdy frame that can distribute impact to many directions, reducing total recoil
- the rubber band mount is very inefficient
  - + this includes an unoptimal position in relation to the arm
  - + the rubber bands keep falling off
- the arm and the stop are better enforced than previous catapult iteration
- the width of the puncher could be widen to ~~ad~~ allow more space
  - + the slip gear is currently getting caught on the arm's c-channel
  - + the rubber band mount can't be added to the sides
- poor gear ratio

Iteration 3 - plan

The rubber band connection accomondates a range of 3"-5.5", which aligns with our intended target specifications after some observations with the rubber bands in our inventory. Regarding the triball platform, it has improved as the triball can be placed in the intended angle. For the ratchet, we could use the hold function of the motor for now, although we should design one after Formosa. We also went back to using a 12:36 ratio.

- the motor was also overheating from strain
- no ratchet mechanism!
  - this is mainly for allowing our arm to stay down without putting stress on the motor
- no calibrating sensor to detect where the puncher is.

After some testing, we realized that the gear ratio didn't really affect shooting speed:

For every time the gear slips, the puncher launches.

unsuspecting big gear...

The slip gear will be on the bottom and attached to the motor.

This means that no matter how big the top gear is, it will always shoot 100 times/minute.

The top gear (output) only determines angle, not speed!

A big logic/reasoning mistake we made!

Day 39 cont.

Puncher alternative (Hung, Bao Nhi, Jack W)

we ~~d~~ were unsure about the placement of the triball in iteration 3 of the puncher since it felt rather unstable.

Therefore, 3 of our team members (Hung, Bao Nhi, Jack W) branched off to work on a second option for the puncher that prioritized easier placement.

Then, we could test both punchers to compare advantages / disadvantages (we need to test since punchers were a new concept to us)

Iteration 3 design of main puncher

a puncher with placement that we liked
credit: 1469A on YouTube

The reason that we liked this design more was because the platform was wide and flat, which didn't require much aiming to place properly

alternative puncher placement

Therefore, the plan for our alternate puncher:

we also liked the idea of having the stopper on the bottom, since we didn't want any bent metal (PTSD from the scrimmage)

~~Day 39 cont.~~

DAY 40 - 27/11/2023

Present All

Puncher Day 5

Today we continued building our puncher based on iteration 3 of the design (see pg 88 for planning)

This is the puncher on our robot! →

we haven't loctite-d the parts of our puncher yet, which is something we need to do before formosa.

We still wanted to also build the alternative puncher on the side.

So, we continued building it based on our plan (p 89)

Some photos of our progress so far:

~~DAY 40~~ DAY 41 - 28/11/2023

Present: All

Intake Day 8

we found out that our intake was slightly oversize (> the 18" constraint).

We only had to fit within this constraint when starting though, since we had a 36" horizontal expansion limit during the match.

we tried putting the preload triball into the intake to see if it would help:

but it was still oversize, so we used a zip tie on a collar lock instead

then, by spinning the intake, the zip tie slides out

which allows the intake to collapse into place.

Puncher Day 6

we continued working on the alternative puncher (based on plan in pg 89)

we cut out green traction wheels and cut them out to use as dampening. we asked members on the VEX community, who said the green traction wheels were best for dampening over 6 things they tested. (pg. 77)

QR code of video:

Then, we put the gears and axles on the robot.

we found out how many teeth to shave off the 36T by:

we shaved the teeth with a file!

Auton Day 2

with the robot now practically done, we began to write our head-to-head auton based on the path we planned earlier (pg. 87).

For our head-to-head auton, we programmed it step by step using drive_forwards(distance) and my_turn(θ, k). This means the robot only moved in straight lines.

we determined the distance and angle using some right triangle trigonometry; for example:

$\theta = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right)$ (soh cah toa)

$|\vec{v}| = \sqrt{(\Delta x)^2 + (\Delta y)^2}$ (pythagorean)

Therefore, turn(−θ) and go forwards (|v⃗|).

we approximated each distance and angle on pg. 87. Then, we tested it.

Defensive:

Offensive:

We ~~tosse~~ used an online pathing tool ~~to create~~ for ease

Offensive pseudocode:

go forwards 180mm

turn right 90°
vomit out triball
go forwards 40mm
wait 100 ms
go forwards -30mm (backwards)

turn right 150°
intake triball
go forwards 80mm

turn right (-180°) (turn left)
vomit out triball
go forwards 70mm
go forwards -50mm (backwards)

intake triball

turn right (-120°) (turn left)
go forwards 80mm
go forwards -30mm.

Defensive Pseudocode:

go forwards 180 mm

turn right −90° (left 90°)

vomit out triball

go forwards 40mm

we reused a PID (only P) function from last year. (next pg)

Our drive forwards function:

```
void drive_forwards(double distance){
  //drives forward distance mm. Circumference of wheel in mm = 3.25π(in) * 25.4(cm/in)
  left_motor_group.setVelocity(600, rpm);
  right_motor_group.setVelocity(600, rpm);
  double revolutions = distance / (3.25 * 2.54 * M_PI);
  left_motor_group.spinFor(revolutions, rev, false);
  right_motor_group.spinFor(revolutions, rev, true);
}
```

we divided the distance by the circumference of our wheels ($2\pi r = 3.25\pi$")

This gave us the # of revolutions, which we spin each wheel side for.

Day 41 cont.

Our PID (but only P) turning function:

```
void my_turn(double expected, double coefficient){

  expected = expected + Inertial.rotation(degrees);

  double error = expected;

  while (std::abs(error) > 2){
    actual = Inertial.rotation(degrees);
    error = expected - actual;
    speed = error * coefficient;

    left_motor_group.setVelocity(speed, percent);
    right_motor_group.setVelocity(-1 * speed, percent);

    left_motor_group.spin(forward);
    right_motor_group.spin(forward);

    wait(150, msec);
  }

  left_motor_group.stop();
  right_motor_group.stop();

}
```

It uses the inertial sensor to determine the actual ~~error the~~ angle the robot turned.

Then, we compared the actual angle with what we wanted and turn until the error is <2°.

"Coefficient" ~~is~~ (the 2nd parameter) is a constant we choose.

We use 0.6 for our autos.

Defensive code:

```
void opp_color(){
  xPos = -90;
  yPos = -150;
  drive_forwards(180);

  my_turn(-90, 0.6);
  intake.spinFor(reverse, 1, rev, false);
  drive_forwards(40);
  wait(100, msec);

}
```

Offensive code:

```
void same_color(){
  drive_forwards(180);

  my_turn(90, 0.6);
  intake.spin(reverse);
  drive_forwards(40);
  wait(100, msec);
  drive_forwards(-30);

  my_turn(150, 0.6);
  intake.spin(forward);
  drive_forwards(80);

  my_turn(-180, 0.6);
  intake.spin(reverse);
  drive_forwards(70);
  drive_forwards(-50);

  intake.spin(forward);

  my_turn(-120, 0.6);
  drive_forwards(80);
  drive_forwards(-30);

  intake.spin(forward);
  my_turn(-60, 0.6);
  wings_expand();
  drive_forwards(-100);
  my_turn(180, 0.6);
  intake.spin(reverse);
  drive_forwards(20);

}
```

QR codes to test videos:

Offensive auton

Defensive auton

We pack for Formosa tomorrow!

FORMOSA

01/12/2023 - 02/12/2023

Attending: Aaron, Alex, Jack L, Jack W, Bao Niu, Hung
(Wayne and Derek are with us spiritually)

We attended the Vex Formosa competition in Taipei!

our alliance partners for elims,
6546C

The event was competitive but fun, we learned a lot from other teams, whether they were our alliance partners or opponents!

Formosa cont.

Formosa International VEX Over Under 2023

Practice #7	Dec 1st at 9:22 AM	75016A	1599V	78	901B	4253S	91
Practice #12	Dec 1st at 10:03 AM	1599V	3818A	48	6546A	4253H	108
Qualifier #1	Dec 1st at 12:30 PM	1599V	4813A	125	8861D	4253K	90
Qualifier #10	Dec 1st at 1:18 PM	4253V	15383A	60	1599V	8861C	91
Qualifier #23	Dec 1st at 2:25 PM	1599V	4813D	32	14683A	901B	89
Qualifier #28	Dec 1st at 2:50 PM	901A	6546B	73	4813B	1599V	115
Qualifier #34	Dec 1st at 3:22 PM	1599V	4253R	88	4253H	8861A	89
Qualifier #43	Dec 2nd at 7:31 AM	14683B	901Z	40	1599V	6546C	41
Qualifier #54	Dec 2nd at 8:33 AM	3818C	8861B	99	1599V	3818B	60
R16 #2-1	Dec 2nd at 10:11 AM	6546C	1599V	62	6546A	15383A	96

Rank	19
WP / AP / SP	8 / 24 / 443

Autonomous

Our autonomous sometimes worked and sometimes didn't. It was very inconsistent, but it got us 3 triballs at most.

We think this is because the robot isn't always moving straight (uneven friction in it ??) so we had to align it slightly slanted when starting.

So, we tried to make a makeshift jig for our last matches out of a piece of paper. We folded it and aligned it to the wall and edge of the tile.

Also, we realized that our defensive auton "broke the plane", which means our robot went under our opponent's goal - which was not allowed unless they were double zoning.

Therefore, we reduced the distance that the robot drives into the goal and also pulled ~~backwards~~ out, moving backwards after. This fixed our issue.

Alternate puncher

We also finished building our alternate puncher in the hotel.

However, we realized the rubber band mounting was inefficient and required a lot of force to push down.

We decided to set this aside for now while we focus on our main robot, since we are in a competition. After all, our initial, main puncher was working. After some practice, we all got used to match loading on the main design, and it isn't as difficult as we thought.

Formosa Cont.

Blockers

We noticed that team with blockers generally did well since they could stop their opponents from scoring efficiently.

The blockers we saw looked something like this:

We learned that blockers were an incredibly effective game strategy, especially for defensive sides (which is the role we played most in this competition). Therefore, we hope to be add a blocker to our robot too.

Intake

The intake was working, but not as well as we hoped.

Whenever we tried to intake a triball, the robot was pushing it away:

We think this is an issue with our intake not moving quickly enough to intake the triball.

We are currently using 2×5.5 W motors, which run at 200RPM. Perhaps we could switch to a 1×11W 6:1 motor, which runs at 600RPM, 3x the speed.

Youtube Playlist-
Formosa matches

DAY 42 - 04/12/2023

Present: All

Intake Day 9

Reflecting from Formosa

After reflecting, we came up with some issues with the current intake during Formosa.

They included:

- The intake getting stuck inside the goal
- issues with triball intaking (there was not enough tension too)
- feeding was poor - took long time to intake triball and "gain control" of it
- Another thing we wanted was for the intake to be able to score without having the robot go inside the goal ("breaking the plane")

Wings Day 7

also reflecting from Formosa

Some notes we took on wings from our discussion:

- The nylock came loose and we often had to retighten - maybe use loctite?
- Overall, it was good and worked well for us
- We considered adding polycarb wedges on the wings, but felt that it was impractical after some discussion

We ~~had~~ have a competition coming up soon (Fullbright), so we made a list of tasks we want to get done:

Progress	Priority	Task	Category
Complete	Complete	Auton Skills - Position Tracking	Code
In Progress	Do soon	Auton Skills - Boomerang controller	Code
Complete	Complete	Driver Control	Code
In Progress	Do soon	15s Auton - 1 triball	Code
In Progress	Do soon	15s Auton - 4 triball	Code
In Progress	Do soon	15s Auton - 6 triball	Code
Complete	Complete	Tweak the intake for drive practice and testing	Intake
Unbegun	Do soon	Redesign the new intake	Intake
In Progress	Do soon	Research of puncher designs	Puncher
Unbegun	Do later	Fix triballs getting stuck	Wedge
Unbegun	Do soon	Loctite it	Wings
Unbegun	Do later	Design new smaller wings	Wings
Unbegun	Do soon	Replace collar locks with spacers	Drivetrain
Unbegun	Do later	Re apply loctite	Drivetrain
Unbegun	Do later	Change to 450 rpm	Drivetrain

DAY 43 - 06/12/2023

Present: Bao Nhi, Alex, Hung

Puncher Day 7

while our main puncher was done, we still wanted to work on ~~th~~ improving the 2nd puncher to see how well it can work.

we saw that team N (1599N) had a good puncher, so we asked them for some tips about rubber band mounting.

They suggested that instead of putting 2 rubber bands, we can ~~loop~~ fold 1 rubber band:

before: 2 bands

after: 1 band folded into 2 "layers"

We tried this on our puncher and the triballs shot much further!

06/12/2023
Puncher test

Skills Auton Update - PID Tuning

Entry by: Alex (Skills programmer)

APS PID

The way we implement PID with APS is we turn towards the point we want to go using an angular PID, and then drive towards it using a linear PID. Moving linearly is very simple (we just use the pythagorean theorem to find the distance), but turning is not so simple. To find the angle to turn, we need to use trigonometry. Since we only know deltaX and deltaY (which are opposite and adjacent to angle θ), we use $\tan^{-1}$, aka atan in code. The issue with atan is its range is $-\frac{\pi}{2} < \theta \leq \frac{\pi}{2}$ which is not the full 2π. This means it doesn't recognize opposing headings since it can't tell the difference between them. For example, if the input is $\frac{1}{1}$ (45°) or $\frac{-1}{-1}$ (-135°), it both inputs 1 so the function can't tell the difference. We can solve this with several logic statements to check the quadrant, and then add π to it if it's in quadrants II and III.

However, after testing this in the VEX VR simulator, we found that it was quite inefficient. We came up with a more efficient algorithm without logic statements. We created this after learning about the atan2 function while reading documentation. This function has a range of $-\pi < \theta \leq \pi$ as it has two inputs as opp, adj so it can tell the difference between quadrants. We tested this algorithm in VR and saw that it worked well.

Tuning PID

One of the most tedious and time consuming parts of a PID controller is tuning– we have to find and input optimal constants to ensure minimal oscillation, minimal steady-state error (error after the robot stops moving), and minimal settling time. The steps to tune is the following:

1. Increase kP until slight oscillation
2. Increase kD until no oscillation
3. Increase kI until no steady state error

Parameter	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
kP	Decrease (faster)	Increase (further)	N/A	Decrease (more precise)	Worsens
kI	Decrease (faster)	Increase (further)	Increase (takes longer)	Decrease (more precise)	Worsens
kD	N/A	Decrease (closer)	Decrease (quicker)	N/A	Improves*

* Wikipedia

As mentioned, we do this to ensure the robot is very accurate while saving time, so we spent a lot of time tuning. We also use a script we made in rust that lets our computer connect to the brain using bluetooth and give us a live graph of things we want to see. We will most often graph *time* v *distance* because we can see the oscillation (look at peaks), the steady state error (looking at distance from position to desired), and

the settling time (looking at the slope). This script has helped us alot because instead of relying on visual feedback, we can tune with very accurate feedback, allowing us to ensure the accuracy is within 1 inch, and also no oscillations. We spent today tuning the controller using the above mentioned method for both linear and angular PID.

DAY 45 - 08/12/2023

Present: Bao Nh., Alex Hung, Jack L

Puncher Day 8

Cross-court shooting and AWP

We wanted a puncher that could shoot high to evade blocking, but it would also be nice if we could shoot cross-court (for AWP).

We tried adding more rubber bands to our robot. We would need to find some way for these rubber bands to detach after the first shot, so that our puncher can return to normal after autonomous (we would only shoot once in auton).

This is a potential way we could score a solo AWP, without relying on our alliance partner.

Ratchet

From the previous ratchets we designed and tested, (pg. 38), we designed a new ratchet to put on the puncher.

We wanted to use a ratchet to reduce strain on the motors that is pulling against the rubber band.

ratchet prototypetesting

DAY 46 - 11/12/2023

Present: Bas Nh., Alex, Thug, Vaughn

Puncher Day 9

Testing our AWP theory

We continued testing our theory of using additional rubber bands to shoot cross-court for the autonomous win point.

We ended up testing with 18 rubber bands (9 per side).

we could place it on a box to see the trajectory. it shot very high instead of far laterally.

AWP puncher test

versus what would be ideal for a cross-court shot:

we would prefer if the arc was low and strong to have a higher chance of pushing it in the goal

Also, pulling back the puncher required a lot of force that we think the motor wouldn't be able to handle.

Intake Day 10

Prossessing the triball

When analyzing the intake, we noticed two things:

1. The height of which the intake wheels come into contact with the triball's top is very important. With the design taken to Formosa, the wheels ~~some~~ were too low in relation to the height of the triball
2. The intake plate taken to Formosa was not ideal. The plate was not solid enough, it was too high up, and it was too far back for the robot to score without going into the goal.

To address these issues, we chose to increase the height of the intake to be closer to the tip of the triball. This was done in two ways. Firstly, the hinge of the intake was placed higher. Secondly, we raised the height of the stop so that the intake is angled higher. With these changes, there was no longer a need for the intake to be heavily tensioned down, and the intake can no longer get stuck under the goal.

In addition to the height of the intake, we also changed the plate of the intake to rubber bands. We found a lot of success with possessing the triballs quicker while using rubber bands. However, when testing the intake on the field, there was an issue with the rubber bands touching the floor. This not only makes the intake less reliable with the bands wearing out faster, but it also makes driving the bot slower. Some alternative solutions we thought about were:

- Adding a plate connected with rubber bands so that there is less friction when the triball is collected
- Using a bar instead of rubber bands
- Changing the placement, orientation, and shape of the polycarbonate plate.

DAY 47 - 13/12/2023

Present:

Drivetrain Day 9

Collar locks

As we were driving the robot around, the collar locks kept falling out :(.

Therefore, we wanted to replace the drivetrain collar locks. We used spacers to move the collar locks to the center, making it easier to retighten.

The CAD plan for the spacing

The rest will be the same with our previous drivetrain, only we are rearranging the spacers and collar locks for the 6 axles connected to motors.

We used 2× 12mm spacers and 1× 1/16" click spacer.

Puncher Day 10

We wanted to redo/replace our puncher since it was extremely flimsy. Since it was being held up with standoffs, it got loose very quickly. The rubber band mounting was also not ideal since it was tall and floppy.

We tried to put the alternate puncher on our robot, but we faced a problem. The gears on the alternate puncher were horizontal:

However, we didn't have much horizontal space on our robot, since it was taken up by other parts like the intake and pneumatic tanks.

We had a lot of vertical space though. Therefore, we had to change to a vertical gear arrangement.

Alex wanted to try a 12:36:36 ratio:

This would work by attaching 2 gears on the 2nd level.

The bottom gear (input) is 12T and spins at 100 RPM. The slip gear is a 36T double slip. This gives us:

$$\underbrace{\frac{12T}{36T}}_{\text{gear ratio}} \times \cancel{100RPM} \times \underbrace{100\frac{\cancel{Rev}}{min}}_{\text{motor rpm}} \times \underbrace{2\frac{shots}{\cancel{rev}}}_{\text{double slip}} \approx 66.67 \text{ shots/min}$$

which is slower than previously but still fast enough for us.

Over Formosa, we realized that flat platforms weren't as bad as we thought. With a little practice, we learned how to effectively place triballs on the setup.

Although it may be a bit unintuitive to someone new to our bot, our entire team has learned how to place it - so it shouldn't be much of an issue anymore.

contrary to what we initially thought, placing it in the middle like this doesn't shoot as well.

we found that grabbing the triball by the tip and placing it so that the bottom is slightly slanted towards the front. This shot our triball with be a better arc.

Therefore, we will try to implement the vertical gearing (12:36:36) and flat platform on our alternative puncher.

CADs that we have for concept design:

we also plan to add standoffs for bracing, since our last puncher came loose and became very flimsy.

We plan to add this on both sides of the puncher for even more structural support.

We also are thinking about using a small piece of polycarbonate in this shape:

This allows the puncher to go through while the rubber band mounting can stay on the side.

It also helps hold the top of the puncher together.

DAY 48 - 14/12/2023

Present:

Intake Day 11

Revising the Intake Sleds

With the intake sleds, there was a lot of room for improvement:

1. The placement of the sled could have been moved to be in contact with the intake arms, instead of being held out with standoffs.

2. The centerpoint of the sled's curve should be placed higher to direct tangential force upon contact with the sled to always aim up and away from the hinge.

Instead of going straight into a CAD, we decided to prototype our intake sleds with cardboard before cutting the polycarbonate.

Prototype 1

For the first prototype, it was a simple design which tries to remove any concave curves. However, the center-point of its curve was too low, causing the intake to not move up. This is because the direct tangential force upon contact with the sled aims towards the bottom of the hinge, not upwards enough.

Prototype 2

Therefore, in the second prototype, the sled's center point for the curve was moved upwards to be leveled to the top of the arm. When testing this prototype, this sled moves the intake up as intended.

DAY 49 - 16/12/2023

Present:

Intake Day 12

Cutting the sleds

With the second prototype for the sleds (pg. 104), we can move on with cutting the polycarbonate.

Extending the sled slightly up provided a nicer shape too.

Puncher Day 11

we built our puncher based on the CAD design (pg. 105)

we ran out of high strength shafts, so we had to use low strength for the 2 lower ones.

this caused the low strength shafts to bend after some testing. Therefore, we added a ziptie to try to keep the 2 shafts parallel.

Also, the rubber bands were slipping upwards when we were shooting

so, we added a spacer on top so it won't slide out if it slides to the top

We also added some dampening

we did this by adding some rubber cut from the traction wheels (pg. 77)

we did this by using zip ties

Day 49 cont.

~~DAY~~

Blocker Day 1

Throughout Formosa, we realized the importance of blockers. They were able to block shooting bots from shooting their matchloads, greatly impeding the opponent's ability to score goals.

We wanted to make a two-stage blocker:

We can implement the second stage using string:

It would be ideal if we could use 2 pistons (one per side), but due to limited resources, we could only use 1. This makes the blocker a little bit wobblier.

Therefore, we will only add a piston to the RHS of the blocker.

Photos from building and prototyping:

Sleds Day 7

we encountered some issues with our sleds (pg 63) during testing (driving practice). They were unreliable and not smooth.

We think this is because it had a very steep angle.

We fixed it in our next iteration by making the slope flatter.

Programming Update!

we changed the wings to being toggled and added the same thing for the blockers.

```
void wings(){
  piston_wings.set(!piston_wings.value());
}

void blocker(){
  piston_blocker.set(!piston_blocker.value());
}
```

```
//wings
Controller1.ButtonR2.pressed(wings);

//blocker
Controller1.ButtonA.pressed(blocker);
```

we did this by using boolean values: the "!" operator flips the boolean for whether the wings are on/off, so:

on (open) → off (closed)
off (closed) → on (open)

when we click the button that runs the function.

Curve

After some driving practice, our driver (Jack L) felt that our current controls were too sensitive.

We were using a linear system (no curve). Our ~~pc~~ formula was:

$$f(x) = \frac{12}{100}x$$

where x was the joystick input $(-100 \le x \le 100)$
and $f(x)$ is the output $(-12 \le f(x) \le 12)$

However, our driver wanted to try a curve again. we used the following formula:

$$f(x) = \left(e^{-\frac{t}{10}} + e^{\left(\frac{\text{abs}(x)-100}{10}\right)} \cdot \left(1 - e^{-\frac{t}{10}}\right)\right) \cdot \frac{12x}{100}$$

where,

- x was the input from the controller joystick in percent $(-100 \le x \le 100)$
- $f(x)$ was the output, in volts $(-12 \le f(x) \le 12)$
- t was a constant that we can adjust to the driver's liking

Day 49 cont.

On graph, this ~~too~~ curve looks like:

when t=1,

when t=5,

We guessed + checked different values of t until we found one that Jack L was satisfied with. The curve worked this time!

After testing different values, Jack L was most comfortable with t=4.2.

Thus, the curve would look something like this:

t= 4.2

In code:

```
float driveCurve(float input, float scale) {
    if (scale != 0) {
      return (powf(2.718, -(scale / 10)) + powf(2.718, (fabs(input) - 100) / 10) * (1 - powf(2.718, -(scale / 10)))) * 0.12 * input;
    }
    return input;
}
```

We made a function with the 2 parameters

input, the position (%) of ~~axis~~ axis $\frac{1}{3}$, which is the horizontal axis

scale, which serves as t from our curve formula (constant)

If scale is not 0, it returns the curved value of input.

Day 49 cont.

we would use this curve on turning, which should be added on our ~~"throttle"~~ "turn" value in the arcade drive:

pseudocode:

~~throttle~~ turn = curved value of Axis ~~3~~ 1 position
throttle = Axis 3 position × 12 / 100

left value = throttle + turn
right value = throttle − turn

```
if (Controller1.ButtonR1.pressing()){
  puncher.spin(vex::directionType::fwd, 100, rpm);
}
else if (Controller1.ButtonB.pressing()){
  puncher.spin(vex::directionType::rev);
}
else{
  puncher.stop(brakeType::hold);
}

if (Controller1.ButtonL2.pressing()){
  intake.spin(vex::directionType::fwd);
}
else if (Controller1.ButtonL1.pressing()){
  intake.spin(vex::directionType::rev);
}
else{
  intake.stop();
}
```

This was the remaining part of our while loop.

We used if statements to hold buttons down and run motors.

R1 runs the puncher

and

L2 runs the intake

Autonomous Day 3

During Formosa, some of the angles for our autonomous were off. However, the overall steps were right so we just needed to tweak some numbers.

We guessed and checked some values until we found the right ones. Here are our final autons:

defensive autonomous

```
void opp_color(){
  xPos = -90;
  yPos = -150;
  drive_forwards(180);

  my_turn(-90, 0.6);
  intake.spinFor(reverse, 1, rev, false);
  drive_forwards(40);
  wait(100, msec);

}
```

offensive autonomous

```
void same_color(){
  drive_forwards(180);

  my_turn(90, 0.6);
  intake.spin(reverse);
  drive_forwards(40);
  wait(100, msec);
  drive_forwards(-30);

  my_turn(160, 0.6);
  intake.spin(forward);
  drive_forwards(80);

  my_turn(-180, 0.6);
  intake.spin(reverse);
  drive_forwards(70);
  drive_forwards(-50);

  intake.spin(forward);

  my_turn(-113, 0.6);
  drive_forwards(80);
  drive_forwards(-30);

  intake.spin(forward);
  my_turn(-60, 0.6);
  wings_expand();
  drive_forwards(-100);
  my_turn(180, 0.6);
  intake.spin(reverse);
  drive_forwards(20);

}
```

FullBright!!!

17/12/2023

Present: Jack L, Bao Nh, Aaron, Jack W, Alex, Derek, Hung, Wayne

Our 69-point skills auton

We competed in the Fullbright tournament, placed 7th overall, and made it to the semi-finals!

Our alliance partner was team 15996 from our school.

blocker

some of our cool elevation working! →

Fullbright cont.

Vietnam VRC Tournament 2023

Qualifier #1	Dec 17th at 9:01 AM	1599R	1599V	96	36070D	89206A	67
Qualifier #7	Dec 17th at 10:24 AM	36070D	36070E	114	1599C	1599V	29
Qualifier #8	Dec 17th at 10:40 AM	36070A	1599X	22	1599N	1599V	77
Qualifier #15	Dec 17th at 11:59 AM	1599V	36070A	58	36070M	36070X	156
Qualifier #16	Dec 17th at 1:11 PM	89206A	1599X	52	1599V	1599D	60
Qualifier #21	Dec 17th at 1:52 PM	1599V	1599D	74	36070F	1599C	89
Qualifier #25	Dec 17th at 2:31 PM	36070E	36070X	83	1599N	1599V	105
Qualifier #30	Dec 17th at 3:18 PM	1599E	36070F	74	1599V	1599R	131
SF #2-1	Dec 17th at 4:04 PM	1599R	1599V	85	36070X	36070M	114

Rank	7
WP / AP / SP	10 / 32 / 459

~~Puncher~~

— Reflection —

Puncher

- our puncher wasn't great. It shot inconsistently and the axles bent, causing proble
- However, it was reliable enough for us to score 69 points in auton skills witho moving.

- We think this is because our puncher was hastily built last-minute, so we didn't have a lot of time to plan and go through the design process.
- we hope to be able to build a new, more reliable puncher.
- Also, our current puncher is very thin, making it more wobbly compared to our opponents who had wider punchers:

Blocker

- our blocker ~~allowed~~ was the thing that helped us win a lot of matches.
  - we were able to block very good shooters, lowering the scores of our opponents.
- our string mechanism didn't really work and fell apart. we didn't have time to fix it so we took the second stage off.
  - Although we only had 1 stage, we were still able to block most of the shots

Fullbright cont.

- we also felt that our puncher was flimsy/wobbly
  - For the next iteration, we hope we can use 2 pistons (1 per side).

Game Strategy

- throughout the tournament, we noticed there were many teams strong on ~~offense~~ defense (shooting), but not so many teams were strong on ~~defense~~ offense (blocking).
  - therefore, we want to focus on improving our blockers and wings (we will still try to make a good puncher for skills or when our alliance partner can't shoot match loads).
- since our puncher was quite inconsistent, we played on offense (pushing triballs and blocking opponent's shooter) while our alliance partner shoots over to us.
  - This is why we chose to form an alliance with 159R (they had a strong puncher)

Programming (H2H driver)

when we tried to implement the curve (pg. 107-to 108), we implemented it on the wrong variable:

before code:

```
// Throttle is how fast we want to be moving foward or backward.
// Turn is how fast we want to spin about the center.
double throttle = Controller1.Axis3.position() * 12 / 100;
double turn = Controller1.Axis1.position();

//arcade drive
double leftVal = driveCurve(throttle + turn, 4.2);
double rightVal = throttle - turn;
```

This caused us to not be able to drive properly during one of our matches.

The curve was applied to "leftVal", which is the wrong variable. Instead, it was meant to apply to "turn" so we could control turning.

Putting it on "leftval" would make only the left side of our drivetrain wonky. Therefore, for that match, we weren't able to move properly.

We quickly noticed that our robot wasn't driving properly and debugged the error.

After code:

```
double throttle = Controller1.Axis3.position() * 12 / 100;
double turn = driveCurve(Controller1.Axis1.position(), 4.2);
```

For the matches after this, our robot drove normally.

Full bright cont.

· Therefore, the main while loop of our program:

```
while (1) {

  // Throttle is how fast we want to be moving foward or backward.
  // Turn is how fast we want to spin about the center.
  double throttle = Controller1.Axis3.position() * 12 / 100;
  double turn = driveCurve(Controller1.Axis1.position(), 4.2);

  //arcade drive
  double leftVal = throttle + turn;
  double rightVal = throttle - turn;

  if (reversey){
    leftVal = -1 * (throttle - turn);
    rightVal = -1 * (throttle + turn);
  }

  left_motor_group.spin(directionType::fwd,leftVal, voltageUnits::volt);
  right_motor_group.spin(directionType::fwd,rightVal, voltageUnits::volt);

  if (Controller1.ButtonR1.pressing()){
    puncher.spin(vex::directionType::fwd, 100, rpm);
  }
  else if (Controller1.ButtonB.pressing()){
    puncher.spin(vex::directionType::rev);
  }
  else{
    puncher.stop(brakeType::hold);
  }

  if (Controller1.ButtonL2.pressing()){
    intake.spin(vex::directionType::fwd);
  }
  else if (Controller1.ButtonL1.pressing()){
    intake.spin(vex::directionType::rev);
  }
  else{
    intake.stop();
  }

  wait(5, msec); // Sleep the task for a short amount of time to
                 // prevent wasted resources.
}
```

Programming (H2H auton)

Our H2H autonomous worked for 3 triballs.

However, it was quite inconsistent.

We made a makeshift jig out of paper (again) which helped fix the issue a bit.

However, we kept losing our jig 💀. This led to a few matches where our auton went wrong.

We hope to be able to build more jigs for nationals (in case we lose one) and also make them more rigid/stable.

Fullbright cont.

Skills Auton

Rank	Team	Driver Attempts	Driver Highscore	Programming Attempts	Programming Highscore	Total Highscore
7	1599V	2	50	1	69	119

we placed 7th overall and scored 69 points in programming skills!

This is the program we used for programmer skills:

```
void auton_1() {
    dt_left.set_brake_modes(mode: pros::E_MOTOR_BRAKE_HOLD);
    dt_right.set_brake_modes(mode: pros::E_MOTOR_BRAKE_HOLD);

    chassis.setPose(x: -46,y: 60,theta: 135);
    puncher_motor.move_velocity(velocity: 100);
    pros::delay(milliseconds: 45000);
    puncher_motor.move_velocity(velocity: 0);
    chassis.moveToOld(x: -40,y: 55, timeout: 1000, forwards: false, async: false, maxSpeed: 48)
    chassis.moveToOld(x: 36,y: 60, timeout: 1000, forwards: false, async: false, maxSpeed: 48);
    chassis.moveToOld(x: 58,y: 30, timeout: 1000, forwards: false, async: false, maxSpeed: 48);
    chassis.moveToOld(x: 48,y: 46, timeout: 1000, forwards: true, async: false, maxSpeed: 48);
    chassis.moveToOld(x: 8,y: 0, timeout: 1000, forwards: true, async: false, maxSpeed: 48);
    wing_solenoid.set_value(true);
    chassis.moveToOld(x: 40,y: 0, timeout: 1000, forwards: false, async: false, maxSpeed: 48);
```

This program got us 69 points.

Some notes of the program:

- it was able to shoot match loads over the barrier
- However, when it was supposed to go to the other side and push the triballs into the goal, ~~it~~ the robot was only moving 1 inch when it was supposed to move 24".

~~we think~~ this affected our ~~a~~ run, and we weren't able to score as much as the program intended to.

We think this is because the program was written last-minute.

Alex wrote it on the bus to the competition, so we didn't have time to test and debug it.

Next time, we will definetly try to write the program earlier and test it more, not save it for last-minute.

DAY 50 - 19/12/2023

Present: Bao Nhi

Autonomous Day 4

Testing our autonomous

After fullbright, we wanted to test our autonomous again.

(Since our defensive auton was simple and consistent, we focused on our offense auton)

On our 4th attempt, we got what should've been a 4 triball auton. However, our ~~wings weren't~~ pneumatics weren't filled, so the wings didn't open.

Test results

attempt	# of triballs /4	notes
1	0	first movement too much towards right - messed up everything else
2	3	touched but didn't intake triball 3
3	2-3	missed 2 (need to turn right a bit more), 3 only had 1 vertex in
4	4	should be a 4-triball auton if wings opened properly

attempt 1

attempt 2

attempt 3

attempt 4

DAY 51 - 26/12/2023

Present: Alex, Bao Nhi, Jack L, Jack W.

We are on our winter break now, so we will be working at Bao Nhi's home for the next 2 weeks. Our main focus for this winter break will be:

1. improve / rebuild the puncher
   - make it shoot consistently
   - make the arc higher so it can shoot over low blocking
2. improve & stabilize the blocker
3. work on our skills autonomous

Puncher Day 12

We ~~didn't~~ don't like our current iteration of the puncher.

Some of the problems we hope to fix in the next iteration (identify the problem):

1. shafts bending

- our low strength shafts were bending. After consulting with team 1599R (before break), they suggested we use spacers to strengthen the shaft.

Therefore, some solutions to the problem:

① use high-strength shafts

we could use high strength shafts instead of the low strength ones we used in our previous iteration.

Pros:
- very strong, won't bend
- we don't need spacers since the axle won't bend

Cons:
- we need to drill holes
- we don't have HS shafts available to us at this moment

② "makeshift high-strength shaft"

adding HS adapters to help the LS shaft impersonate a HS one

Pros
- we have ~~the parts~~ done this before

Cons
- doesn't feel solid
- might slip back / forth

③ lots of spacers

Pros
- easy
- strong

Cons
- lots of spacers
- difficult to figure out spacing

Since we didn't have HS shafts nor a lot of adapters, we decided to go with ③.

Day 51 cont.

2. Wobbly

· our puncher was thin, so it was quite wobbly. we want to widen the distance

3. Complicated gearing

We didn't really like our current gearing of 12:36:36.

we felt that it was unnecessarily complicated and that we could make it faster.

Therefore, we want to go for a simpler gear ratio.

So, we came up with a rough sketch of what our next iteration puncher should look like:

PLAN

From our previous puncher, since the space between the columns is 6 studs.

This means our puncher should be 5 wide to fit it.

We chose 12:60, since we had a 12T gear w/ 4 teeth shaved on hand. a 60T gear would give

$\frac{8T}{60T} \cdot 360° = 48°$ which was nice

BUILDING

We took off the blocker for now so it would be easier to focus on the puncher (also because we wanted to fix the blocker eventually anyway)

We didn't know where to put our rubber bands.

before, we had added a 5-wide c-channel in this way to make it less wobbly.

it also helped keep the spacing even.

For rubber bands, we tried wrapping it around the area

Day 51 cont.

Also, we only had 1 slip gear (we found it, probably from someone's old catapult

it should be 4 gears shaved off, leaving us with 8 teeth. We hope to be able to get a second one shaved, but we're not sure how yet.

For now, we will build the puncher with only 1 12T gear for testing.

we built it out ~~4~~ and tried launching manually to test the trajectory of the triball

we think there are 3 factors that can affect it:

1. contact point
2. rubber band mounting
3. placement of triball

manual puncher test video

When we tested this version, we noticed that the puncher was pushing the triball down more than up:

we felt like it was pushing the triball into the front standoffs.

This would reduce the arc of the triball trajectory, making it not ideal.

The force was being applied in the wrong direction.

We will ~~took~~ mess around with the 3 factors stated above next time to see if we can improve the trajectory of the triball.

Ideally, we can get a curve like this:

DAY 52 - 28/12/2023

Present: Alex, Jack L, Hung, Bao Nhi

Puncher Day 13

Currently, our puncher only has 1 slip 12T gear. We are missing a second one:

we had plenty of normal 12T gears, and we needed to find a way to shave teeth off.

Since we were working from Bao Nhi's home, we didn't have a dremel / angle grinder either.

We ended up using a knife-sharpening stone, water, and many hours to try and shave the teeth off.

After lots of shaving, we finally did it!

We attached the slip gear on, so our final spacing looked something like this:

We also tested ~~and~~ attaching the rubber bands this way

we felt that by doing it this way, the rubber band would be more "straight".

We used one double-looped rubber band per side.

Also, we felt that the point of contact was too close

perhaps by adding another "layer" to the c-channel across?

This helps increase the angle the tennis ball is being shot at!

which could help us attain a higher arc

We did this by using 1" standoffs and attaching another 5-wide c-channel on top

attaching on all 4 corners

We tested it manually and still weren't satisfied with the arc.

Our goal for next time is to build a ~~platform~~ better platform ~~for~~ for our puncher.

A photo of our puncher at the end of the day →

Skills Auton Update - More Motion Algorithms

Entry by: Alex (Skills programmer)

Boomerang

One issue with turning and driving separately is that we have to come to a complete stop every time we want to switch directions, and the deceleration is not very fast. Therefore, we wanted to implement curves. We saw a desmos graph demonstration from Purdue SIGBots with an algorithm called a *boomerang controller*. We decided to try to implement it, as it looked like a very simple algorithm. How we implemented it is by continuously turning and driving towards a carrot point. However, this point is calculated based on the current position (with a constant called dlead which changes the curvature of the curve), so it will change throughout the motion and get closer to the target. When repeated at a very high frequency, it creates a very smooth curve that follows a quadratic bezier curve. After testing, we very much liked this algorithm as it is faster than regular PID (since the robot moves in a curve). It's very simple, but still accurate.

$$h = \sqrt{(x_{start} - x_{end})^2 + (y_{start} - y_{end})^2}$$

$$x_1 = x_{end} - h\sin(\theta_{end}) \cdot d_{lead}$$

$$y_1 = y_{end} - h\cos(\theta_{end}) \cdot d_{lead}$$

$$\left((1-t)\left((1-t)x_{start} + tx_1\right) + t\left((1-t)x_1 + tx_{end}\right), (1-t)\left((1-t)y_{start} + ty_1\right) + t\left((1-t)y_1 + ty_{end}\right)\right)$$

Pure Pursuit

While researching, we also saw an algorithm called Pure Pursuit that we wanted to test, since we read it is very customizable and fast (also because it was called PP and *someone* on our team found it funny *ahem*). The benefit of this algorithm is it follows a path that can be made and customized in any way we want. It takes an array that has thousands of points which it follows. It fixes errors using a look ahead point by drawing a circle with a radius of a constant, called the look ahead distance (if set too low there will be lots of oscillation but if too high will take long to fix error), and will find the intersection between it and the path that is going forward. Then, it will drive towards this intersection. As mentioned, we switched to this because of customizable paths. We found an interactive web app where we can create/visualize a path and download it into the code. We make this path with a combination of lines, and cubic and quadratic bezier

curves. The reason we have two curve types is because they have their own pros and cons. The cubic bezier has four control points, so it allows for more complex curves. The quadratic bezier (same as boomerang) has three control points, so it is faster to create paths for. The bezier curve was originally created by Pierre Bézier for the bodywork of Renault cars, but we use it because it has the ability to replicate any imaginable curve. How it works is by splitting the control lines into different segments, then drawing lines in between, eventually making a curve with enough iteration.

After testing and choosing the advantages of the 3 options (PID, boomerang, PP), we plan to use pure pursuit for autonomous skills because of its speed. We don't really need the accuracy since we will use wings to push triballs into the goal, so some error is OK.

High Density + Animations (RGB)

Throughout the competitions, our robot stood out a lot because of our rgb. However, there were some things we thought could be improved. The biggest thing was the density of the RGB– our original RGB was quickly put together so we used what we had on hand. We used 30 led/m strips which had a big gap between pixels. This causes hot spots of bright color. We solved this by ordering leds with a higher density, but something we had to consider was the tradeoff between a high density strip and a long strip as the brain's adi port has a maximum amperage limit of 2 amps. Each pixel will use up to 60 mA which means we can only have 30 LEDs. However, this is very little, and we will never have all the leds show bright white. We use the approximation of 20 mA which will allow us to have up to 100 LEDs. We expect to only need around 1 meter, so we decided on using the 96 LED/m strip, which makes the gradient look very smooth with no hotspots. Another thing we didn't like was the animation. We originally used linear interpolation, but the transition between colors were very rough, so we could only use similar looking colors. Also, the animation was very basic and not interesting. We found an algorithm called Perlin noise, which is a randomization algorithm, but based on previous values to create smooth noise. We used this to create an organic looking gradient and a semi-random animation (it is based on a seed, which we can keep constant to keep the same gradient/animation). This makes the animation amazing. However, the Perlin noise algorithm is very computational intensive, which decreases the performance of other important things like odometry. We solved this by only using the noise for every 12 pixels, then we used cubic interpolation. It can guess data in between, like linear interpolation, but it uses a cubic spline, giving it smoother in-between points.

DAY 53 - 31/12/2023

Present: Hung, Bao Nhi

Puncher Day 14

Our goal today was to make a platform for the puncher that is easy to place but can also shoot effectively.

First, we felt that the piece of polycarbonate ~~on the~~ that acts as the backbone of the platform is too floppy.

only attached by 2 screws and 2 zipties!

with so much force from the puncher smashing into it during every shot, we didn't feel super secure with this flimsy polycarbonate.

Also, every time we wanted to move around something, we had to drill ~~a~~ new holes. This was tough since our drill just ran out of battery and we didn't take a charger 💀

Therefore, we switched it out with bars and plates:

The benefit of using an aluminum plate is that it has many holes, making it easier to adjust / test new things for us.

We prototyped it separately this way before we put it on the robot.

This was the layout we were most happy with.

We could place the triballs easily. additionally, putting the short ~~spacer~~ standoff in the back forces the triball to face forward.

exposes flat surface to puncher

⇒ ~~shooter super~~ effective shots!

We can place it on by grabbing the triball from the top

triball slides into place!

We really liked the prototype so we put it on the puncher.

We decided to keep the base plate long (and not cut it more) just in case we wanted to change the placement. It gave us the option to widen the space between the standoffs.

We tested launching triballs manually and it worked great! It was exactly the type of arc we were looking for!!

It didn't take much force too (1 double looped rubber band /side) so we tested it with the motor.

We think this part that we added to keep the spacing even might make the robot oversize. We plan to move it back a few studs.

With the motor, the trajectory was BEUTIFUL. It was even better than when we tried launching it manually.

It was also fast, too, shooting at 100 times /min.

~~Although~~ we're not sure what this looks like on the field, though. We should test it when we get back to school.

manually testing

testing w/ motor

Innovate Award Submission Information Form

Only for events that are offering the Innovate Award: Either this form or an entry with the equivalent information is required to be eligible for the Innovate Award. Digital submissions can additionally use the Engineering Notebook link feature found in the "My Account" dashboard on RobotEvents.com.

Instructions: Please fill out all information, printing clearly. For in-person notebooks, please place this page either inside the front cover of the team's notebook or placed as the **last entry** in the notebook when submitting it for judging. In the case of digital notebooks, a picture of the form can be uploaded and placed either at the beginning of the digital notebook, after the Table of Contents, or entered as the last entry in the notebook. Teams may only submit one aspect of their design to be considered for this award at each event.

Full Team Number: 1599V

Brief Description of the novel aspect of the team's design being submitted:

Our team decided to highlight our addition of RGB LED strips under our robot chassis. Although this function is purely decorative, we take great pride in it because it took lots of research and testing for us to accomplish. It is a fully customizable design feature in which we are able to select different color(s) to be illuminated under our robot on and off the field. This aspect of our robot exemplifies our innovative spirit through our creativity, uniqueness, and ingenuity.

Identify the page numbers and/or the section(s) where documentation of the development of this aspect can be found:

Neopixels - 10/11/2023 - pg. 69.1 (physical), pg. 78 (pdf)

High density + animations - 28/12/2023 - pg. 120.2 (physical), pg. 137 (pdf)