CLIPSE

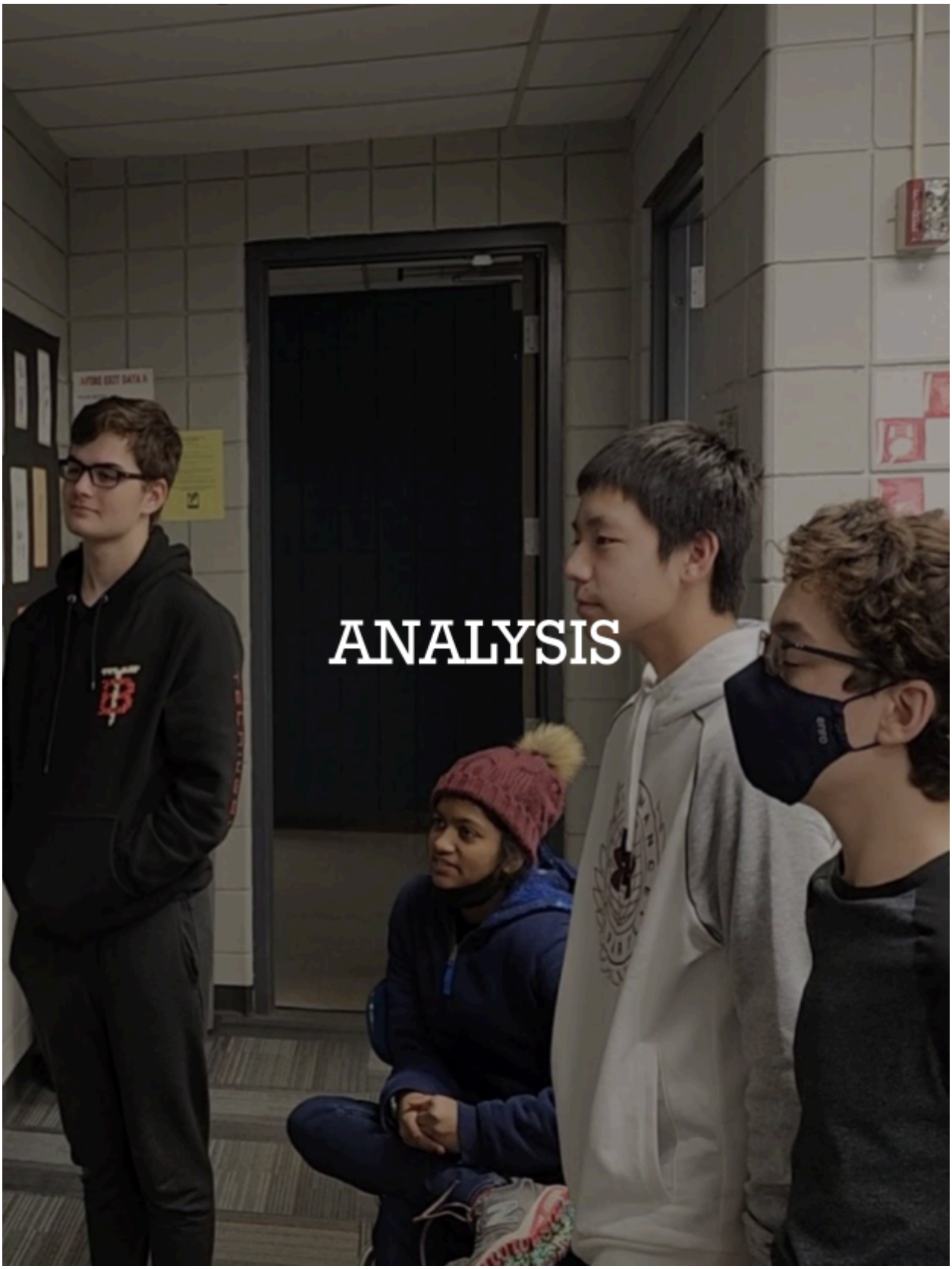
2023

TECHNICAL BINDER

FOREWORD

To involve all WorBots members, our entire team attended the season kickoff in person. This event was followed by a brainstorming session for designs, which helped our team establish an initial list of requirements. From this, multiple prototypes of each subsystem were designed and fabricated throughout the season. As these prototypes proved their practicality, final products were machined while testing and driving practice began. This technical binder covers the subsystems, design process, and programming that went into our robot for the 2023 FRC season.

TABLE OF CONTENTS



ANALYSIS

GAME ANALYSIS

Charged Up is essentially a "pick and place" game similar to 2019's Deep Space. This year's game revolves around 3 core point scoring components: the cone, the cube, and the charging pad. In order to be regional champions, we decided that the following must be accomplished:

Qualifications

- Earn as many ranking points as possible. The more ranking points earned, the higher ranked our team will be allowing for more flexibility in choosing our alliance partners in playoffs

Playoffs

- Maximize our score while also minimizing the score of the opposing team in order to win each match

Our goals are set up so that scoring points will lead to earning ranking points.

Qualification Match Strategy

Ranking points (RPs) are awarded as follows: 2 for winning a qualification match, 1 for completing 5 links, and 1 for earning at least 26 total charge station points in auto/teleop.

- If each alliance completes a link in the coopertition grid, the link RP threshold lowers to 4
- Score 3 links in a match ourselves to minimize the amount of links that our alliance partners need to complete
- Dock and engage the charge station during both the autonomous and teleop period in order to maximize the number of points
- Dock in such a way that our alliance partners have space to dock as well

Assuming that our robot can consistently do the above, we increase the chances of winning the match and achieving the 2 winner RPs, thus granting a high likelihood of getting 4 RPs per match

Playoff Match Strategy

In playoffs, we have a higher likelihood of playing with teams that can auto-level and score 2-4 links individually,

- Maximize scoring cones and cubes in auto (items have increased score and there is no defense)
- Score high links to maximize our points per cycle ratio.
- Minimize cycle time by pre-positioning robot while on the move
- Deliver items accurately to reduce repeat cycles
- Push through defense if necessary

GAME ANALYSIS cont.

Subsystem Strategy

General

- Tank drive takes up less space on platform and has a high torque drive
- Swerve drive is omnidirectional and allows for faster cycle time
- Low center of gravity to prevent tipping during high acceleration maneuvers
- Be able to cross/jump over the charging station if need be

Manipulator

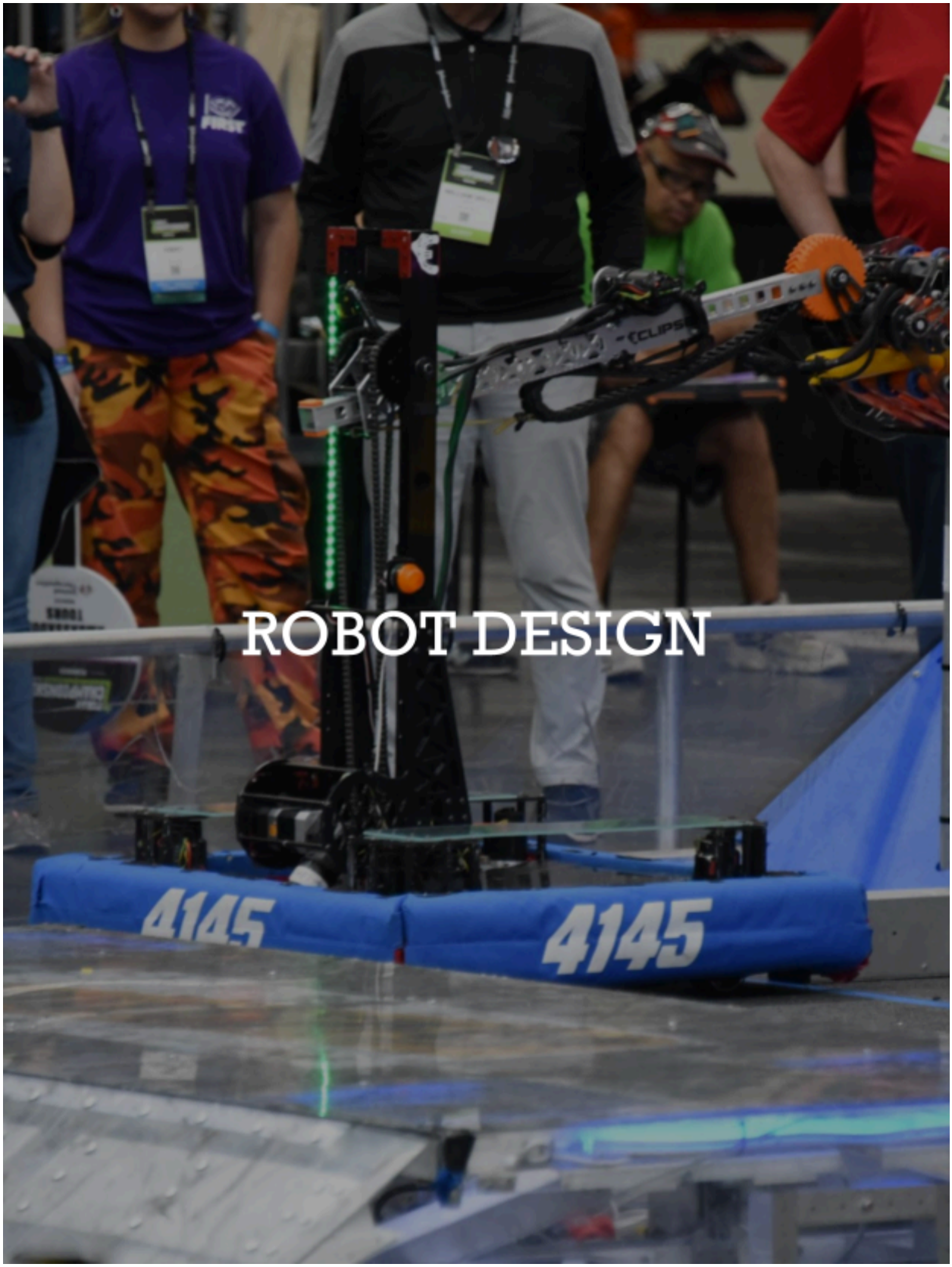
- Touch it, own it, secure it (fast, reliable, and secure method of intaking both cones and cubes)
- Wristing mechanism to increase versatility
- Automatically detect if an item is contained within the manipulator
- Reliability (>90% accuracy in both intaking and delivering cones and cubes)

Arm

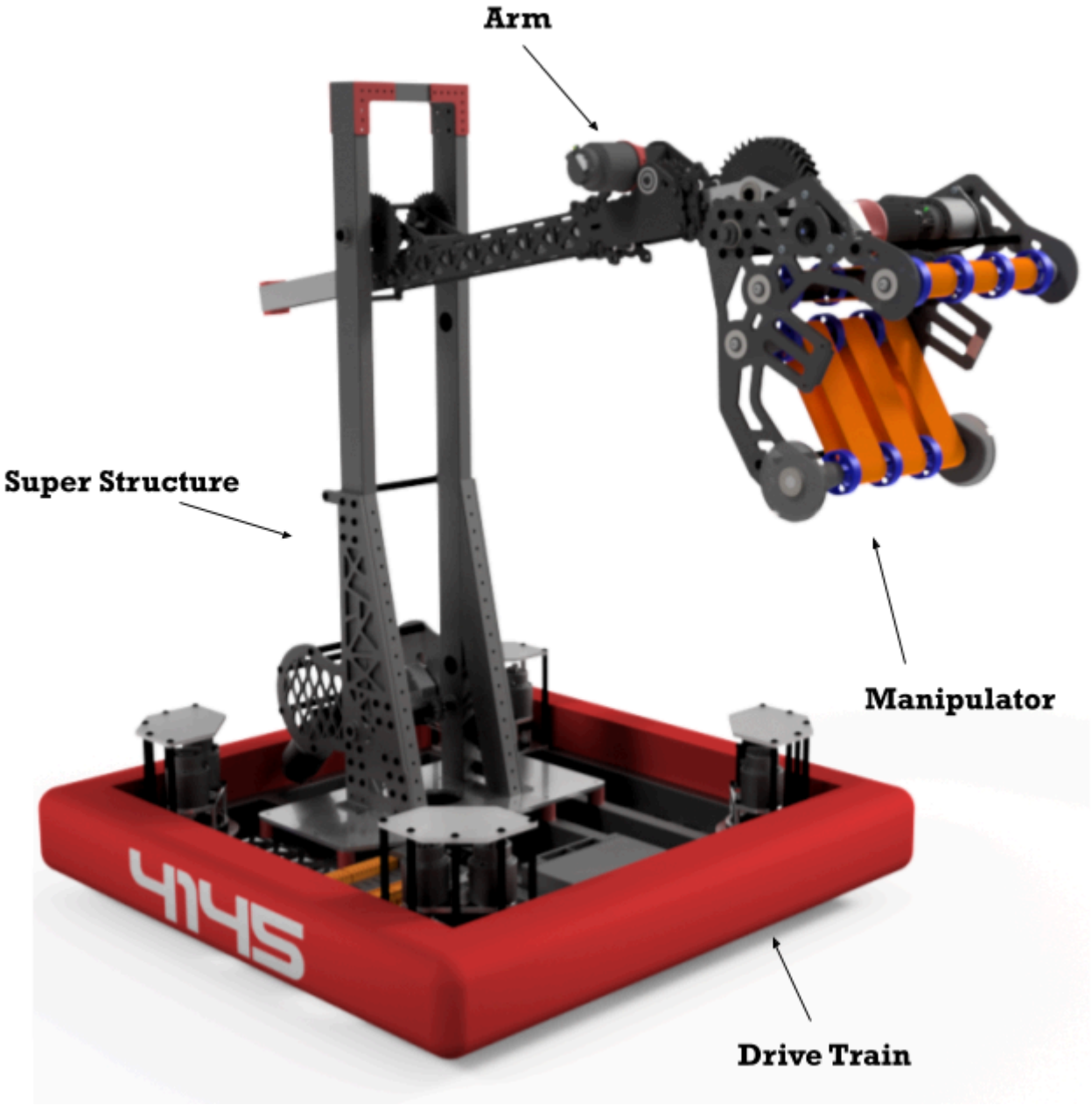
- Fast and reliable telescoping arm
- Be rigid enough to reduce oscillation
- Pivot up and down to access all possible scoring positions

Superstructure

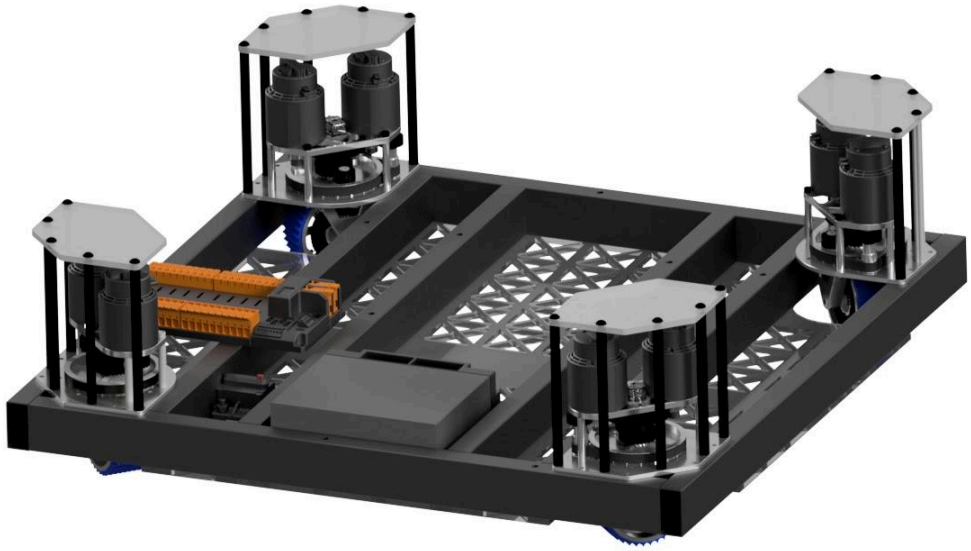
- Rigid and durable to take impacts from other robots
- Be strong enough to support the arm and the variable moment acting upon the structure
- Light weight to allow for quick rotation and to reduce the chances of tipping



ROBOT DESIGN


Arm
Super Structure
Manipulator
4145
Drive Train

DRIVETRAIN



A drivetrain needs to be robust enough to support all subsystems while also having the speed to play an offensive game. A custom chassis with 4 MK4 swerve modules does just that.

Chassis

- Rectangular 2" by 1" aluminum tubing makes a lightweight structure
- Overall frame is 29" by 29"
- 4 27" tubes prevent base plate from bowing
- Swerve modules protected by 6in aluminum churro cage

Base Plate

- 3 9.5" by 29" Quarter inch polycarbonate plates make up the base plate
- Electronics mounted around base plate for easy access
- Custom cutouts for swerve modules

SWERVE MODULE

The swerve module provides proper gear reduction from 2 Falcon500 motors that control the direction and speed of the motor. This omnidirectional locomotion is a massive advantage to fast scoring.

Gear Train

- Wheel direction (Yaw) is controlled by a hybrid gear and pulley system with a 1.75:1 gear ratio.
- Wheel rotation is through a 6.12:1 reduction

4 Falcon 500 Brushless Motors

- 4 Falcon 500 motors provide more power than a NEO and are more compact than CIM motors
- Falcon 500 motors come with an integrated high resolution encoder to provide highly accurate autonomous movement

High Grip Wheels

- 1.5” wide billet wheels for maximum ground contact
- Blue nitrile high grip treads

SUPERSTRUCTURE

The superstructure raises and lowers the telescoping arm with a powerful hybrid gearbox sprocket system, allowing the arm to reach all three levels of the power grid.

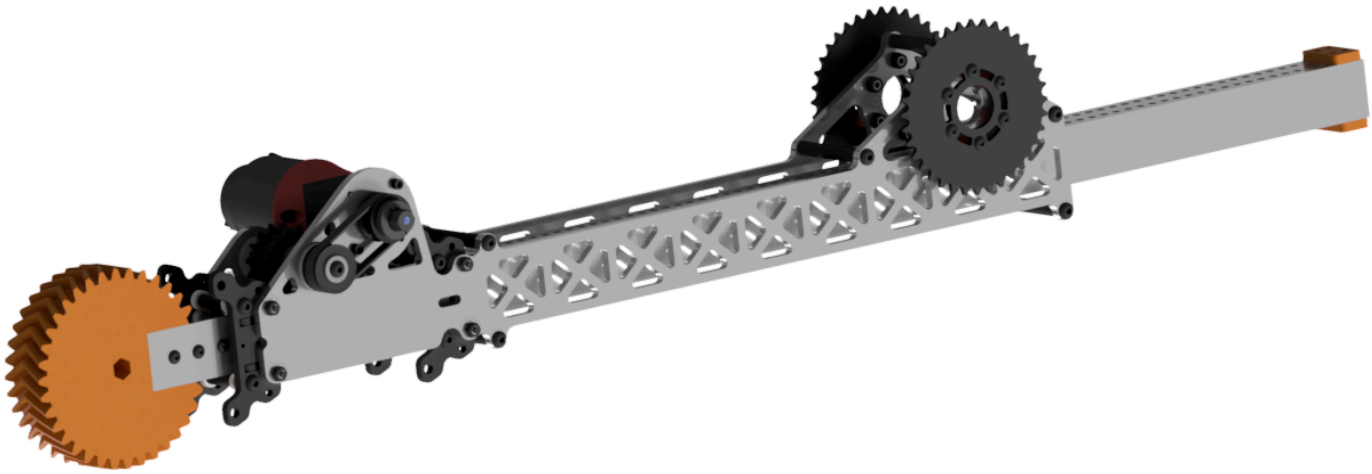
Tower

- Aluminum frame made up of 2" x 1" rectangular tubing provides a strong base
- Several aluminum brackets are used to mount the tower to the chassis and give support to the gearbox
- 18" by 10" quarter inch aluminum plate mounts superstructure to chassis with 1" cylindrical stand offs

Drive

- Sprockets that move the arm are driven by a 80:1 multi-stage gearbox using one Falcon 500 motor
- The two-sprocket system on its own outputs a 1:3 ratio, controlling the arm with high torque and precision

ARM



The arm retracts the intake for a compact stow and extends explosively with a motor-driven sprocket rack system. A modified Thrifty Bot's Thrifty Telescoping Tube Kit is used for the rollers and has served us well in past competitions.

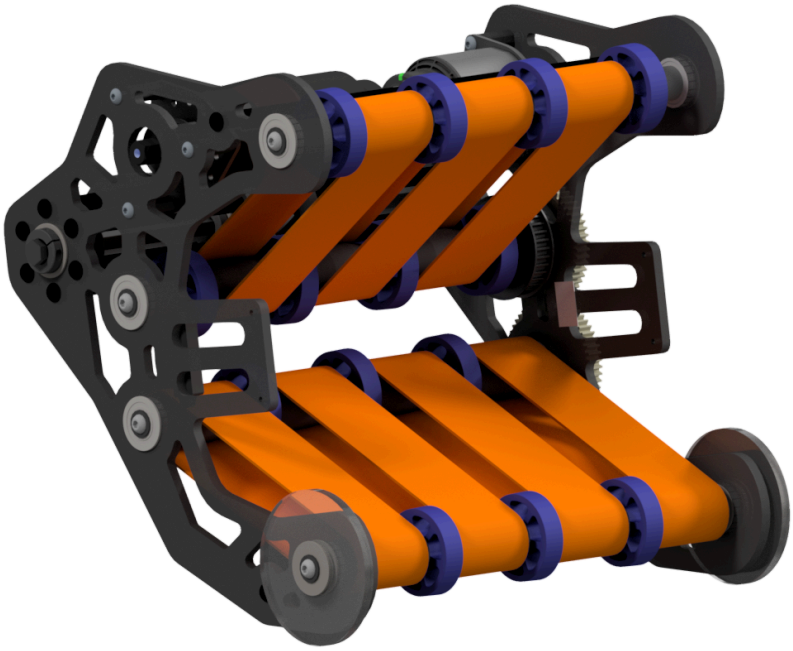
Rack System

- 15 tooth double hub sprocket meshes with a track of holes in the main tube
- Sprocket driven by a pulley system to give motor a compact profile
- Motor is a Falcon500 with a 15:1 VersaPlanetary gearbox
- 27" stroke

Structure

- Quarter inch aluminum plating attached via standoffs
- Open face design allows for ease of repair
- 1.5" Aluminum Tube holds the wristing mechanism

MANIPULATOR



The manipulator acquires game pieces quickly and precisely, utilizing a mouth like belt system that is wide enough for a cube while still being able fit the tip of a cone through the back of the conveyor.

Frame

- Lightweight ¼ "polycarbonate side panels maintain stability while moving
- Free-rotating Polycarbonate wheels allow manipulator to roll on floor while extended
- Mounts on either side are allocated for a time of flight sensor

Belts

- Falcon 500 drives with 1.67:1 belted reduction
- Direction of rotation is inverted for the lower belts with two 54 tooth gears
- Rubber conveyors grip game pieces from any point of contact
- 2" Compliant wheels hold game pieces in place during transport

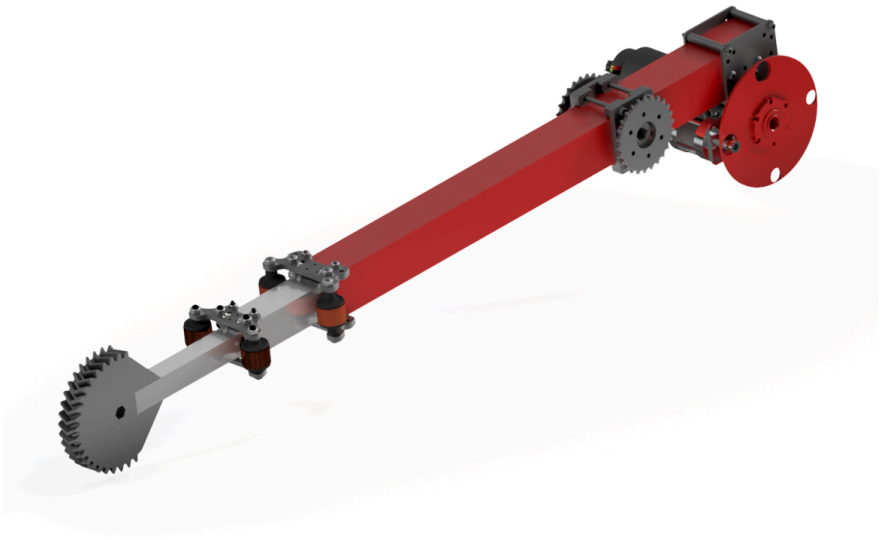
Wrist

- 12 tooth herringbone gear rotates the structure around a stationary 36 tooth gear that has been modified to mount on the arm
- Herringbone gears have a better mesh to stop shifting during rotation
- Falcon 500 drives pulley with a versa planetary 49:1 reduction
- The system allows for a 180 degree rotation

FORMER DESIGNS

Constant Force Spring Telescopic Arm

Revised Week 4



Spool System

- Spool wire runs through the arm internally wound from the 0.75” spool drum
- Spool driven by a Falcon 500 motor with a versa planetary 20:1 reduction
- “Pancake” cylinder wheel system used to lock the arm in a stowed position when in starting configuration

Telescoping Tubes

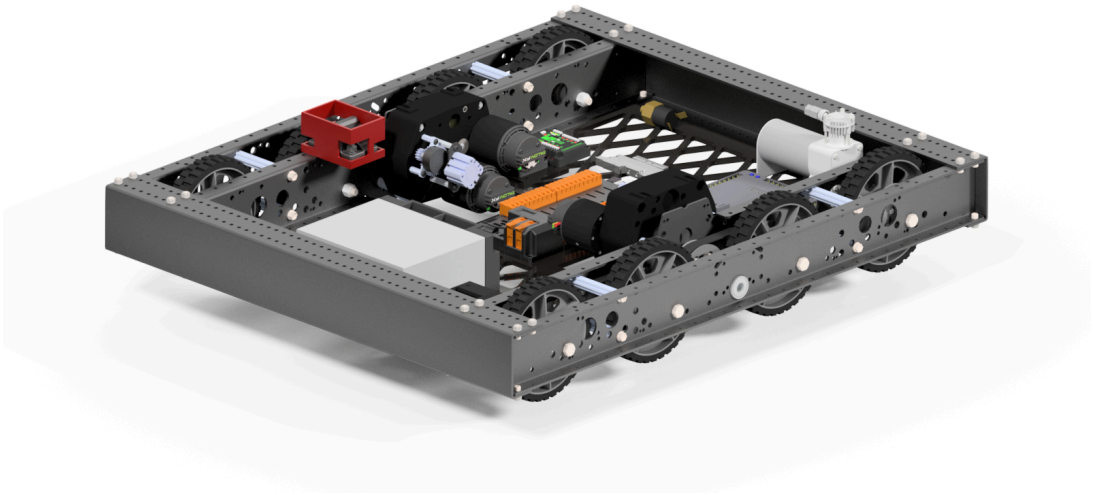
- 1/16” wall square aluminum tubes (2”, 1.5”, and 1”)
- 4 5.94lb constant force springs prioritize quick extension
- 39” long constant force springs prioritize quick extension

Changes

This former design proved to have an unreliable extension as constant force springs were not as predictable as motor movement.This issue enforced the redesign of the subsystem. The new design uses a simple open face rack to control movement which is more reliable, reduces friction, and provides easy access for repairs.

8 Wheel Tank Drive

Revised Week 5



Chassis

- AndyMark AM14U4 8WD Chassis
- 8-wheel drive to maximize stability
- 32” x 27” aluminum frame provides robustness and stability
- 6-inch HiGrip wheels ensure traction
- Familiarity with design
- Narrow enough to potentially fit 3 robots on charging station

2-Speed Ball Shifting Gearbox

- 12 ft/s low gear
- 22 ft/s high gear
- High torque to sprint short distances
- Low gear allows precise control and quick bursts of movement

4 Falcon 500 Brushless Motors

- 4 Falcon 500 motors provide comparable power to 6 mini-CIMs while being contained in a smaller and lighter package
- Falcon 500 motors come with an integrated high resolution encoder to provide highly accurate autonomous movement

Changes

Overall the 8 wheel tank drive worked well early season with a defense oriented strategy. As the season progressed our strategy was revised to be more offensive. Therefore, speed and locomotion was prioritized over torque. This meant converting to a swerve drive.

Turret

Revised Week 5

Turret Base

- Falcon500 motor with 21:1 versa planetary reduction in addition to sprocket reduction
- 6.67:1 sprocket reduction turns the superstructure up to 360°
- Sprockets use durable #35 chain
- ¼ ” thick Aluminum frame paired with 2” spacers forms a stable platform for the superstructure

Triple Bearing Support Structure

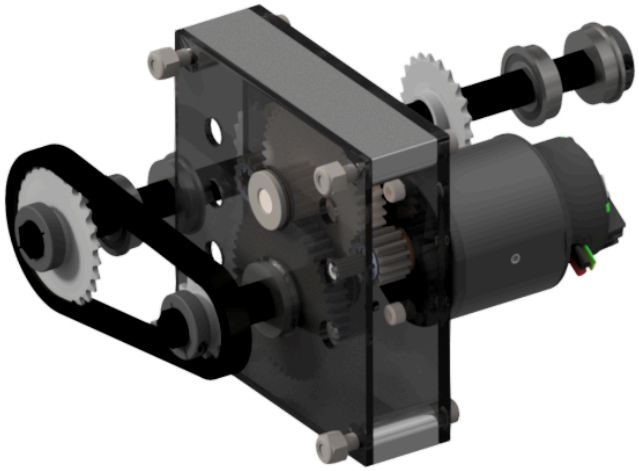
- Turret is mounted to supper structure via 8 #10 screws
- ⅝“ radial bearing separating two ¾“ bearings are inserted on the 8 supports
- Three-bearing combination prevents shifting of the superstructure

Changes

Due to the change from an 8 wheel tank drive to a swerve drive, the turret was no longer needed. While the turret served the tank drive well by rotating the superstructure separately, the swerve drive rotates the entire robot just as well for pick and place gameplay.

Arm Drive Optimization

Revised Week 6

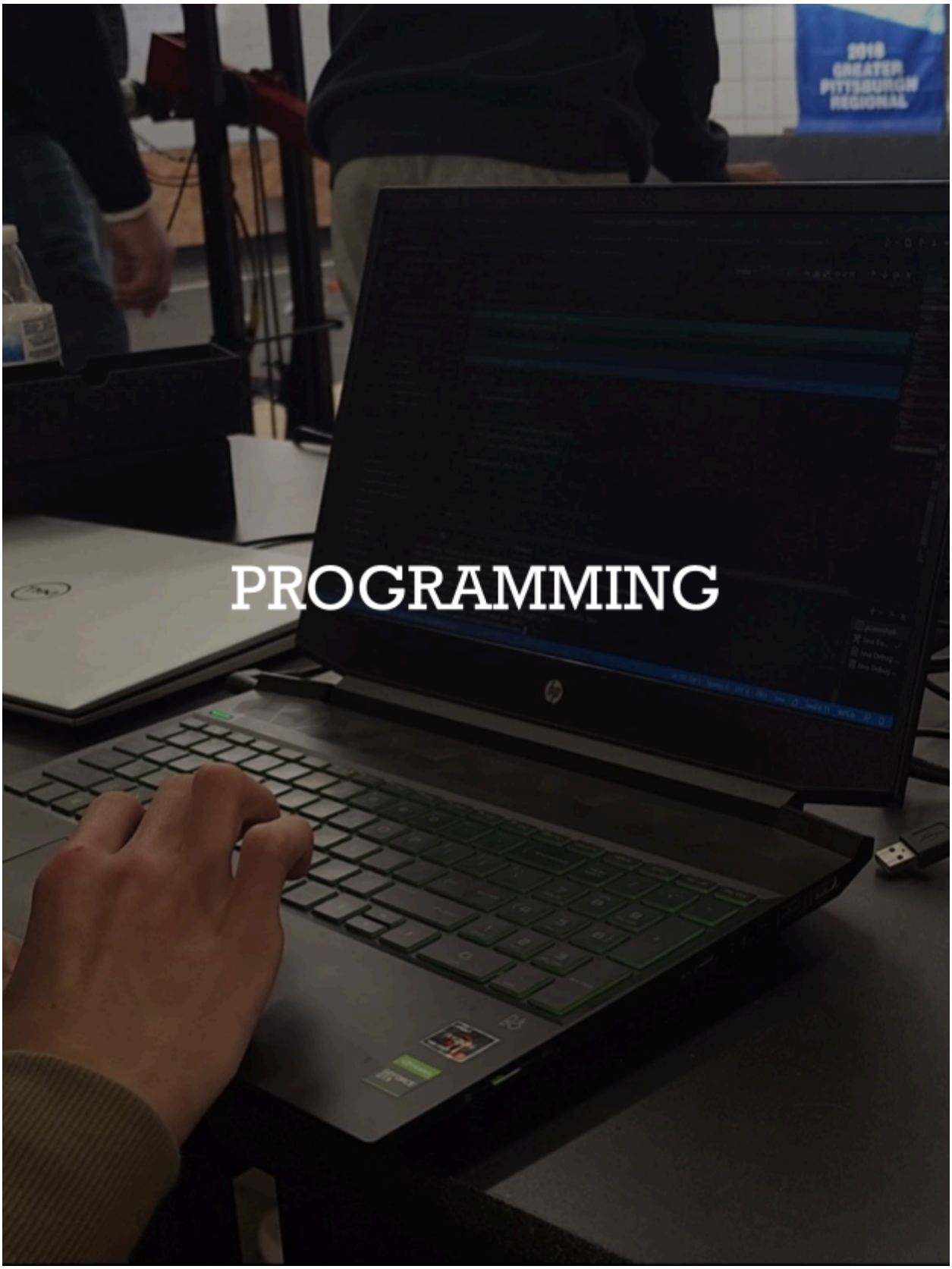


General

- Sprockets that move the arm are driven by a 262.5:1 multi-stage gearbox using one Falcon 500 motor
- The four-sprocket system on its own outputs a 2.19:1 ratio, controlling the arm with high torque and precision

Changes

The main issue of this design had to do with the left most sprockets directly attached to the gearbox. The chain on these sprockets was unpredictable and would often slip. To eliminate having to tension a chain a 1:1 gear ratio has been implemented in their place. With this change the gearbox and sprocket ratios were also optimized throughout the system to ensure precise control of the arm.


2018
GREATER
PITTSBURGH
REGIONAL
PROGRAMMING

SYSTEM ARCHITECTURE

1. Subsystems

Subsystems are the individual, independent systems that run on the robot. There is only a single instance of each subsystem that holds access to actual hardware, following an industry-standard pattern known as the singleton. Its primary job is to read input data from sensors, perform calculations, and then output to actuators that perform tasks on the robot.

Every subsystem is managed by the SubsystemManager. For each periodic loop cycle, the manager will run this process:

1. Every subsystem queries data from sensors and stores it in our logging system, PeriodicIO **(readPeriodicInputs)**
2. More complex subsystems such as the drivetrain will do more intensive calculations such as control loops and path following **(registerEnabledLoops)**
3. Every subsystem writes data to actuators **(writePeriodicOutputs)**
4. Subsystems output telemetry data to SmartDashboard **(outputTelemetry)**

The SubsystemManager also starts, stops, and resets subsystems when the robot changes state while maintaining the state of each subsystem. These subsystems then create an API for our autonomous routines and drivers to call into to allow for a further degree of abstraction.

2. Actions

Actions are individual classes that serve as one of the levels of abstraction. It allows for input from either the driver or the autonomous state machine to call for certain methods to run. Similar to the subsystem classes, the action classes use onStart(), onLoop(), and onStop(), but they use getter and setter methods to set motors to certain values or gather information from the robot.

During Autonomous

In the autonomous period, a State Machine Descriptor is run by our finite state machine runner, which steps through our queue of action groups as part of a preset routine. This allows us to dictate timeouts for each action group and sequence adjustments and calls into the subsystem API.

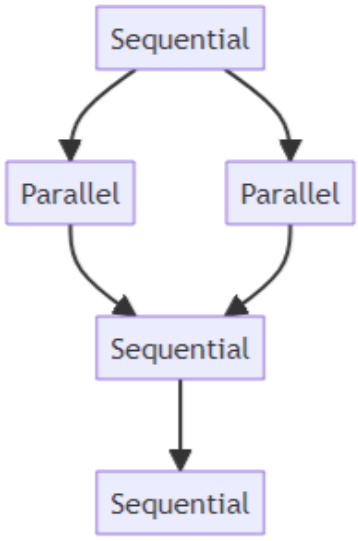
During Teleop

After the autonomous period, drivers take control of the subsystems by calling actions bound to buttons. This is implemented through the same action classes that are sequenced to make autonomous routines, which is a wrapper of the WPI command library.

3. PIDs

PIDs, or proportional controllers, are used in a number of places on *Eclipse*. These controllers allow the subsystems to make smooth motions and correct for errors that can occur over the course of operation. Almost every subsystem leverages PIDs with the drivetrain using them to regulate position, turning, and auto-balancing and the arm uses three loops to allow for a "pose"-based control from the driver.

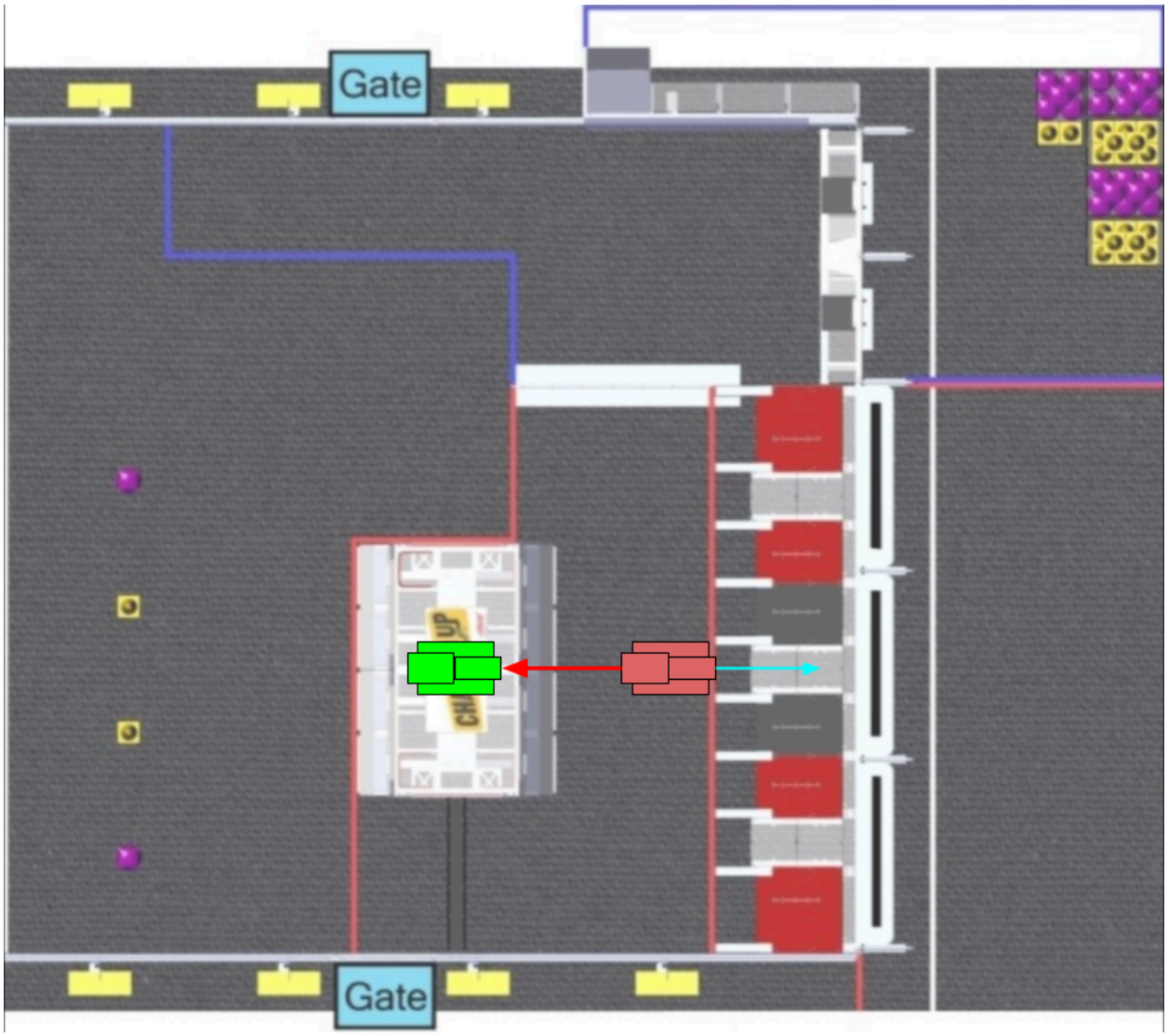
4. State Machine



We implemented a custom linear state machine in order to run autonomous sequences using the same actions that are leveraged during teleoperated control. The state machine progresses through a queue of time-constrained action groups that dictate how the subsystems change state.

AUTONOMOUS

During autonomous mode, *Eclipse* uses a state machine to transition its subsystems through a set of states. *Eclipse* moves in straight paths, using heading correction and manipulating game pieces to score using its arm and intake. The diagram below shows one of the autonomous routines *Eclipse* can do. Blue arrows show arm movement while red arrows show moving to a new position. In the example below *Eclipse* activates auto level once it finishes the movement specified by the red arrow



DRIVETRAIN CONTROL

The drivetrain subsystem is responsible for moving the robot around. The subsystem has multiple states it can be set to depending on the situation. These states make writing autos easier and also allow for assisted teleop.

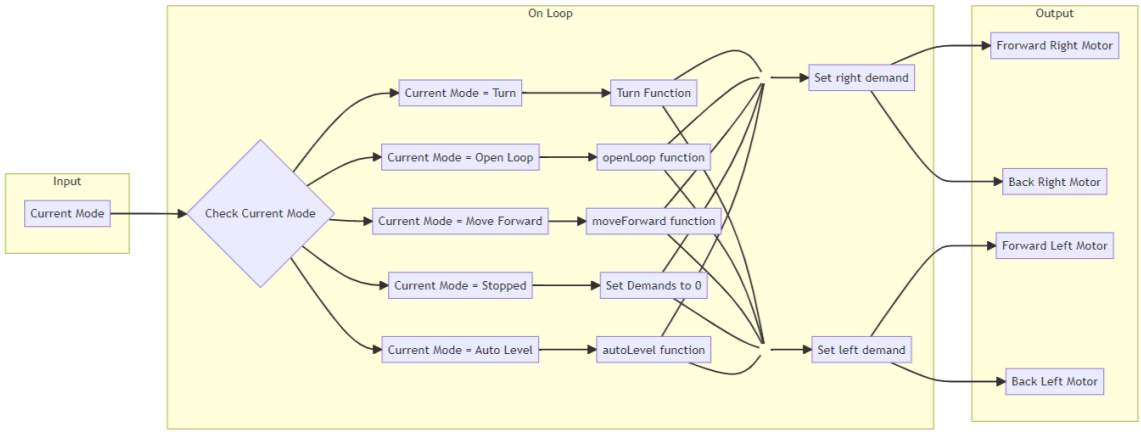
Turn function - Compares the direction *Eclipse* is currently facing to the direction we want to face. This information is then translated to motor inputs to turn *Eclipse*.

Open Loop - Default state of the drivetrain, controlled by the main joystick. Due to its dependence on human input, this mode is exclusively for teleop.

Move Forward - Moves the robot in a straight line. Using the gyro, this mode will also correct back to the original trajectory if it deviates.

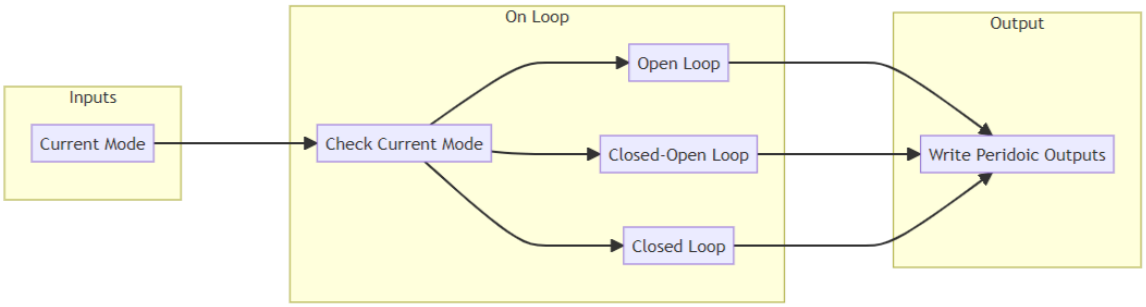
Stopped - Stops the robot by setting all motor inputs to 0.

Auto Level - Used exclusively to dock and engage with the Charge Station. It also returns to the heading it started the match in.



ARM CONTROL

This subsystem controls the arm. The arm can be in open-loop, closed-loop, and open-closed-loop. In open-loop, the arm is controlled by raw data values that are taken from the joystick. In open-closed-loop, the joystick sets encoder goals that the motors on the robot reach. These goals are controlled by PIDs. In closed-loop, desired encoder values are fixed. Joystick buttons are pressed to set the goal positions, once that's done the PIDs once again move the arm into position.



VISION

The vision program uses the Limelight 3 to provide information from reflective vision targets in order to accurately place both cones and cubes on the grid.

Target Acquisition

- Uses the known height of targets as well as the data sent by the Limelight to calculate the goal's position relative to the robot's location
- Auto locks onto the selected scoring grid location when the robot operator selects what position to set the arm

HID HELPER/HUMAN INTERFACE

Eclipse is controlled by one flightstick and one gamepad controller, with each driver having one. The gamepad controls the drivetrain meaning its pilot is responsible for driving the robot around the field. The flightstick controls the arm. This flightstick has significantly more preset inputs as it can command the robot to enter preset positions.

Second Joystick Input Map