Simplicity In Design

By Adam Heard (2135)

Adam Heard

- 16* years FIRST experience
- 2008-2019 973 Lead Mentor (2020 remote)
  - 18 Blue Banners
  - 3x World Champions
  - 3x Champs #1 seed
  - 2016 CVR Woodie Flowers Finalist Award
- Mechanical Engineering Manager at J&J Robotics & Digital Solutions (formerly Auris Health)
- 254 2019-onward
- 2135 2020-onward

Purpose

Learn some methods for architecting the simplest robot possible for the level of competition your team is striving for

Other Workshops & Themes

Other Workshops

- Absolutely critical
  - Karthik’s Strategic Design & Effective FIRST Strategies (2015) (2019 P1 & P2)
  - Mike Corsetto’s Strategic Design (pdf)

- Bonus
  - Mike Corsetto’s Goal Setting
  - Simbotics Scouting & Match Strategy
  - Longer version of this workshop

Karthik's Golden Rules

- Golden Rule #1: Always build within your team's limits.
    - Evaluate your abilities and resources honestly and realistically.
    - Limits are defined by manpower, budget, experience.
    - Avoid building unnecessarily complex functions On the other hand, as you get more. experienced, start cautiously pushing a few boundaries.
- Golden Rule #2: If a team has 30 units of robot and functions have maximum of 10 units, better to have 3 functions at 10/10 instead of 5 at 6/10.

Simplicity Means Less Options...

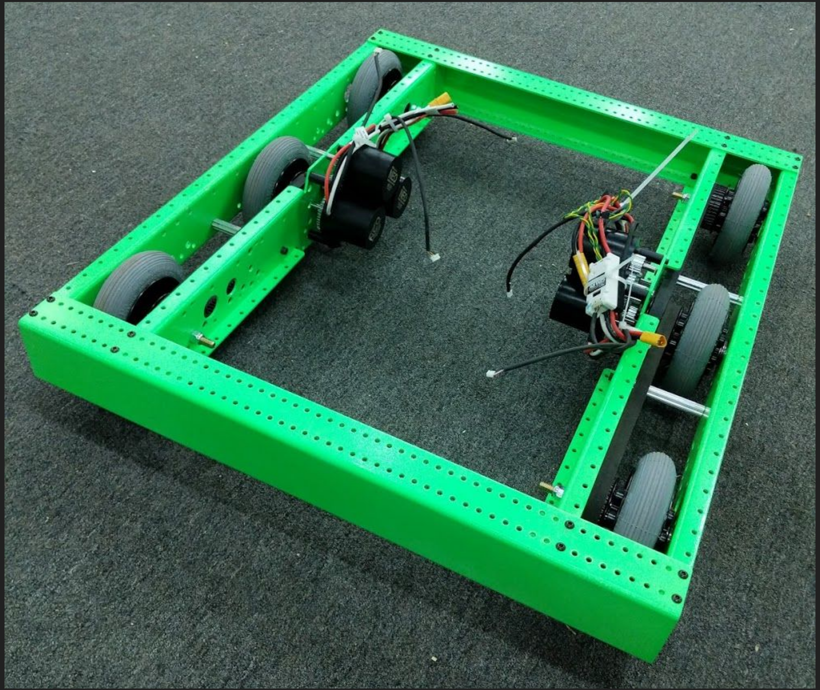
- Making decisions in stressful situations can be difficult and time consuming.
- At home
  - Less features means less work to get everything tuned to 10/10
  - “Your team should be celebrating everything you decide to not do as then you have less work to do” -Mike Corsetto
- Pre-match
  - Drive teams will have an easier time driving a robot that can do less things. Their decision making tree is simpler. This allows better practice, and enables you to cover map out hypotheticals more completely at the strat level
- In pit at events
  - Less tradeoffs of what features to work on, repair, tune, etc…

Competing at Optimal Efficiency

- Perfect application of Karthik’s golden rule yields a robot that competes at 100% of it’s theoretical capability with few or no unspent resource points.
    - Many teams compete at 90% or far less than their robots capability.
    - MUCH easier to add capability to a simpler robot with unspent points versus salvaging a more complicated robot.
        - Teams rarely regret being too simple
- Don’t fall into the off-season trap!
    - Peak at champs or earlier, not after!

Opportunity Cost

- Everything you do is something else you’re not doing
- Constantly assess that you’re efficiently using your resources



Schedule Drives Decisions

- Don’t let design decisions drive the schedule
- Recommended schedule
  - Drivetrain wired and running day 9
  - Comp bot “done” day 29
- If you’re team can’t meet the agreed upon schedule, de-scope design (ideally earlier)

FIRST ROBOTICS COMPETITION
WINNER
2009 LOS ANGELES REGIONAL
FRC
FIRST Robotics Competition
WINNER
2011 GALILEO DIVISION
FRC
FIRST Robotics Competition
WINNER
2011 FIRST CHAMPIONSHIP

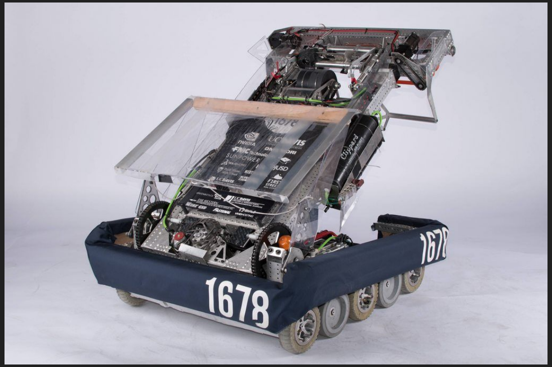
Themes in Simplicity

Simplicity Metrics

- Integration
- Invention
- Tactical features
- DOFs
- Coordinated motion & sequencing
- Mechanism types
- Tuning
- COTS use & fabrication speed

Robot Architecture

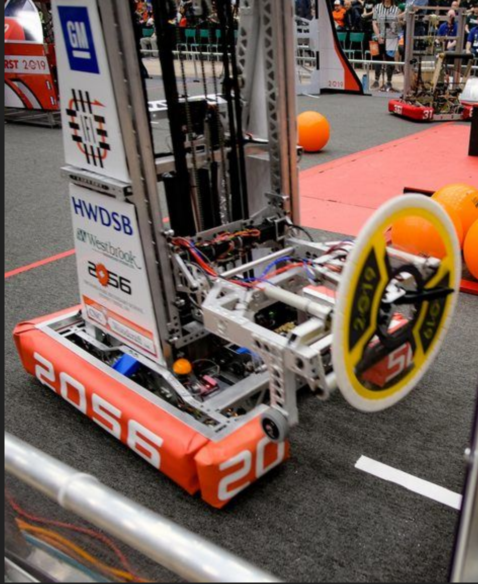
- How integrated is your architecture?
  - Is it a claw on the end of an elevator?
  - A tightly packaged maze of features?
  - A single utility arm?
  - Highly integrated architectures can be tough to get right the first time, and tougher to iterate on.



- Does your robot's proposed architecture require new systems to be designed or adapted to FRC?
  - Or is your architecture based on adapting known workable FRC designs to the current year.

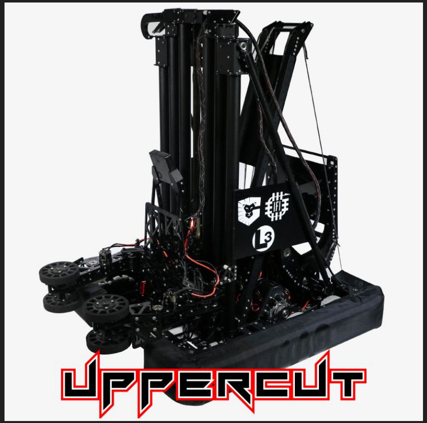
Tactical Features

- Doing all the things - STRATEGY
  - Score high rocket, shoot balls, pick up gears, climb, etc...
- How you do the things -TACTICS/DESIGN
  - Unblockable shots, intake opposite scoring side, buddy climb vs dual climb (19).
- Can you solve something with more practice?
  - Tough if you’re the team always finishing right before events
- WW2056D?
  - Generally skip tactical features



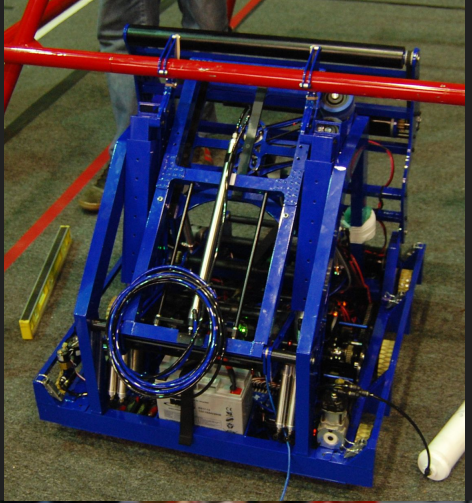
Degrees of Freedom (DOFs)

- Total DOFs
- Servoed DOFs
- Sequencing or gamepiece handoffs
  - Do actuations need to occur in a specific order?
- Coordinated Motion
  - Do actuations need to occur with precision in regards to each other?
- Sensitivity
  - How precise does control need to be?
  - How easy is it to adjust?



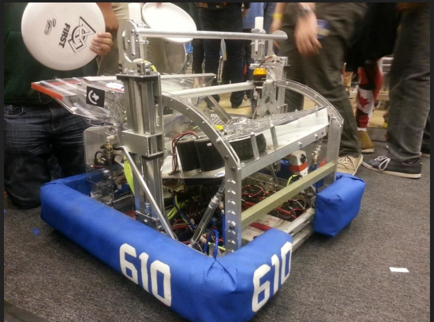
Some Design Trends

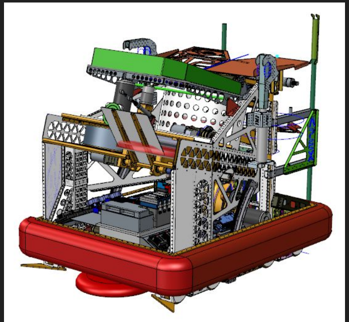
- Rotary motion can be easier to make, often harder to control
  - Especially when multiple joints are involved!
- High force mechanisms can be tough!
  - Climbs, lifts, etc…
  - Launchers, winches, etc… that shift under high load
- Continuous vs non-continuous mechanisms
  - Flywheels, conveyors, etc… versus handoffs, catapults, punchers...



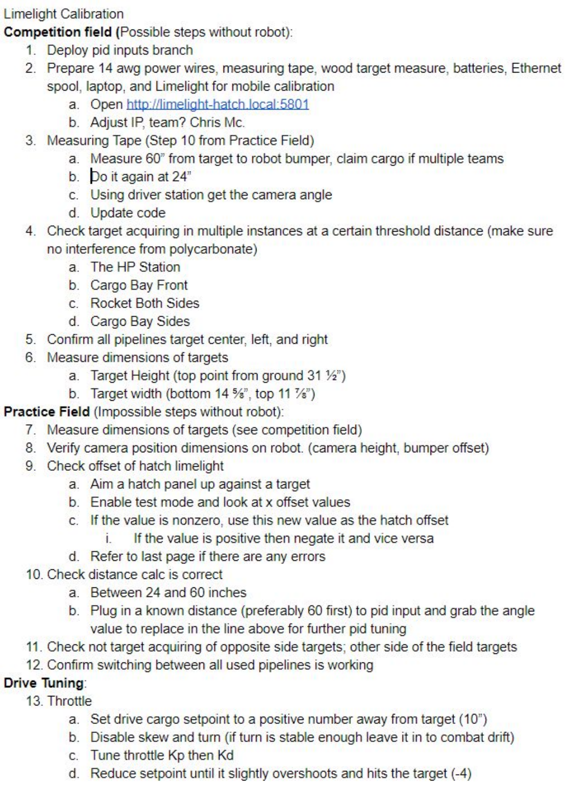
Simplicity In Tuning

- How do you adjust your system? Do you need…?
  - A practice field?
  - A mock goal in pit?
  - To download code?
  - Swap parts out?
  - Adjust assemblies (hole patterns, slots, etc…).
- How sensitive is your robot to adjustment?
  - Does your robot need to be re-tuned after replacing or re-mounting any components?





Tuning Case study



Limelight Calibration

Competition field (Possible steps without robot):

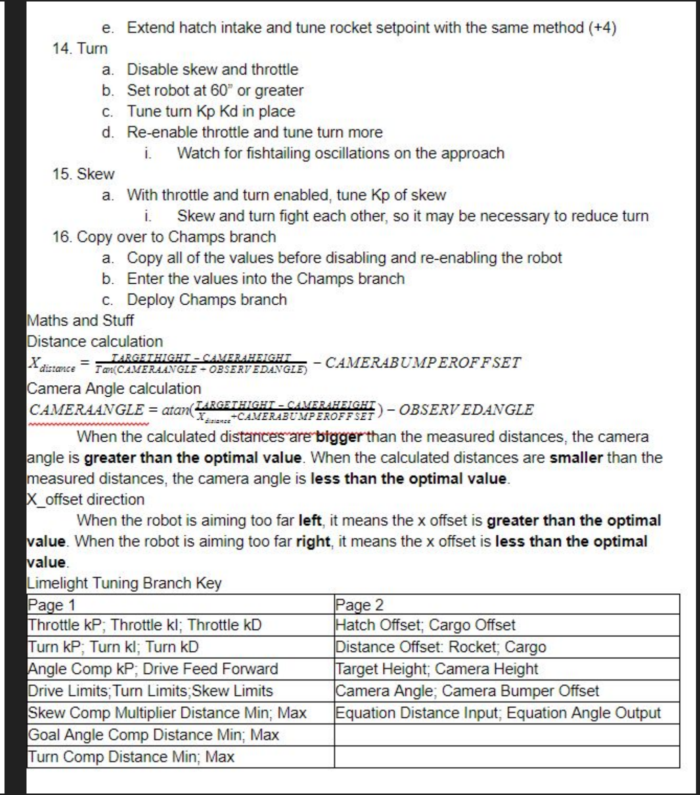
1. Deploy pid inputs branch
2. Prepare 14 awg power wires, measuring tape, wood target measure, batteries, Ethernet spool, laptop, and Limelight for mobile calibration
   a. Open http://limelight-hatch.local:5801
   b. Adjust IP, team? Chris Mc.
3. Measuring Tape (Step 10 from Practice Field)
   a. Measure 60" from target to robot bumper, claim cargo if multiple teams
   b. Do it again at 24"
   c. Using driver station get the camera angle
   d. Update code
4. Check target acquiring in multiple instances at a certain threshold distance (make sure no interference from polycarbonate)
   a. The HP Station
   b. Cargo Bay Front
   c. Rocket Both Sides
   d. Cargo Bay Sides
5. Confirm all pipelines target center, left, and right
6. Measure dimensions of targets
   a. Target Height (top point from ground 31 ½")
   b. Target width (bottom 14 ⅝", top 11 ⅞")

Practice Field (Impossible steps without robot):

7. Measure dimensions of targets (see competition field)
8. Verify camera position dimensions on robot. (camera height, bumper offset)
9. Check offset of hatch limelight
   a. Aim a hatch panel up against a target
   b. Enable test mode and look at x offset values
   c. If the value is nonzero, use this new value as the hatch offset
      i. If the value is positive then negate it and vice versa
   d. Refer to last page if there are any errors
10. Check distance calc is correct
    a. Between 24 and 60 inches
    b. Plug in a known distance (preferably 60 first) to pid input and grab the angle value to replace in the line above for further pid tuning
11. Check not target acquiring of opposite side targets; other side of the field targets
12. Confirm switching between all used pipelines is working

Drive Tuning:

13. Throttle
    a. Set drive cargo setpoint to a positive number away from target (10")
    b. Disable skew and turn (if turn is stable enough leave it in to combat drift)
    c. Tune throttle Kp then Kd
    d. Reduce setpoint until it slightly overshoots and hits the target (-4)



    e. Extend hatch intake and tune rocket setpoint with the same method (+4)
14. Turn
    a. Disable skew and throttle
    b. Set robot at 60" or greater
    c. Tune turn Kp Kd in place
    d. Re-enable throttle and tune turn more
       i. Watch for fishtailing oscillations on the approach
15. Skew
    a. With throttle and turn enabled, tune Kp of skew
       i. Skew and turn fight each other, so it may be necessary to reduce turn
16. Copy over to Champs branch
    a. Copy all of the values before disabling and re-enabling the robot
    b. Enter the values into the Champs branch
    c. Deploy Champs branch

Maths and Stuff

Distance calculation

$$X_{distance} = \frac{TARGETHIGHT - CAMERAHEIGHT}{Tan(CAMERAANGLE + OBSERVEDANGLE)} - CAMERABUMPEROFFSET$$

Camera Angle calculation

$$CAMERAANGLE = atan\left(\frac{TARGETHIGHT - CAMERAHEIGHT}{X_{distance} + CAMERABUMPEROFFSET}\right) - OBSERVEDANGLE$$

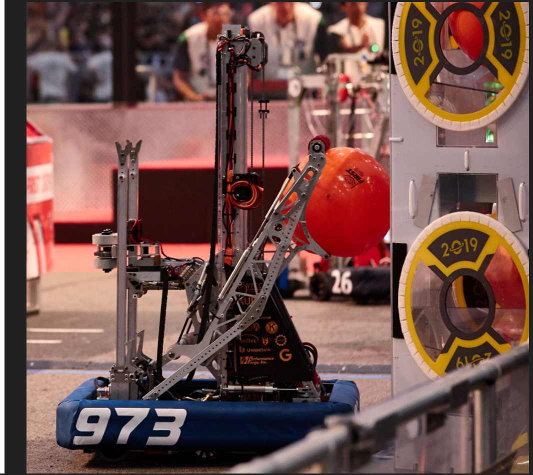
When the calculated distances are **bigger** than the measured distances, the camera angle is **greater than the optimal value**. When the calculated distances are **smaller** than the measured distances, the camera angle is **less than the optimal value**.

X_offset direction

When the robot is aiming too far **left**, it means the x offset is **greater than the optimal value**. When the robot is aiming too far **right**, it means the x offset is **less than the optimal value**.

Limelight Tuning Branch Key

Page 1	Page 2
Throttle kP; Throttle kI; Throttle kD	Hatch Offset; Cargo Offset
Turn kP; Turn kI; Turn kD	Distance Offset: Rocket; Cargo
Angle Comp kP; Drive Feed Forward	Target Height; Camera Height
Drive Limits;Turn Limits;Skew Limits	Camera Angle; Camera Bumper Offset
Skew Comp Multiplier Distance Min; Max	Equation Distance Input; Equation Angle Output
Goal Angle Comp Distance Min; Max	
Turn Comp Distance Min; Max	



COTS Use And Standardization

- Resources required to implement an architecture will wildly vary depending on your teams unique setup and how much COTS you're using.
- Study other teams to find faster ways to build the same thing.
  - 1678 and 973 are great examples of 2D plate and tube builds
- Ways to save time
  - Use COTS components where possible
  - Standardize
    - Material selections
    - Fasteners
    - Fabrication and design methods