

# Linux Based Robotics

---

Wesley Aptekar-Cassels  
FRC Team 1678



# About 1678



# What is FRC?





# Season Recap 2015



# 2016 Game

- **AUTO LINE:** a 3 in. wide line of green gaffers tape that is 12 in. from the edge of the MATING
- **BOUNDING LINE:** a 2 in. wide gaffers tape that is 8 in. wide of the CASTLE and is 27.5 in. from the CASTLE WALL
- **ALIGNMENT LINE:** A 2 in. wide line of green gaffers tape that is 11 in. from the CASTLE WALL



2.2.15

© 2015 by the University of California, Berkeley. All rights reserved.



1678

# Talk Summary

- Our Workflow
  - Git
  - Bazel
  - Vim
- FRC Control System
  - RoboRIO
  - Driver Station
  - Field Management System
- Programming
  - UnitsCPP
  - PID Control
  - Lemonscript
  - Vision



# Our workflow

- Git
- Bazel
- Vim



# Our workflow » Git

- Distributed VCS that we all know and love
- Fast branching
- Decentralized
- Various frontends/UIs, most are bad
  - Having a visualizer (gitg, etc) can be good



# Our workflow » Bazel

- Build system based off Google's internal system
  - Open Source
- {Fast, Correct} - Choose Two
  - Only rebuilds needed files
  - Reproducible - Same code, same binary

# Our workflow » Vim

- The best text editor

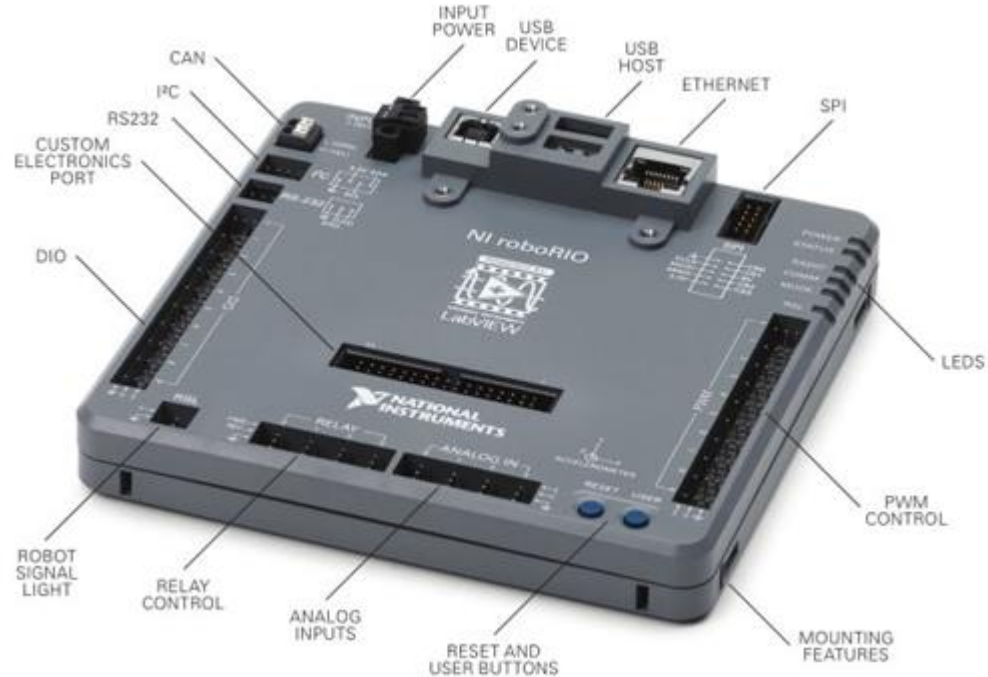
# FRC Control System

- RoboRIO
- Driver Station
- Field Management System



# FRC Control System » RoboRIO

- Linux-based computer
  - Produced by National Instruments
  - Runs realtime Linux
- I/O
  - Ethernet, USB, PWM, CAN, DIO, Analog Input, I<sup>2</sup>C, SPI, and more
- Only used for FRC



# FRC Control System » Driver Station

- FIRST Provided software to control robot
  - Windows only :(
  - Communicates via wifi/ethernet, custom protocol
- QDriverStation
  - Open source, cross platform (including mobile)
  - Sketchy, not competition legal

# FRC Control System » FMS

- System used to control robots at competition
  - One VLAN per team
- FMS communicates with driver station, which controls robot



# Programming

- UnitsCPP
- PID Control
- Lemonscript
- Vision

# Programming » UnitsCPP

- Unit aware programming
- Compile time errors for unit mismatches
- Different units are different variable types
- Not created by our team

# Programming » PID Control

Control theory is an interdisciplinary branch of engineering and mathematics that deals with the behavior of dynamical systems with inputs, and how their behavior is modified by feedback. (Wikipedia)

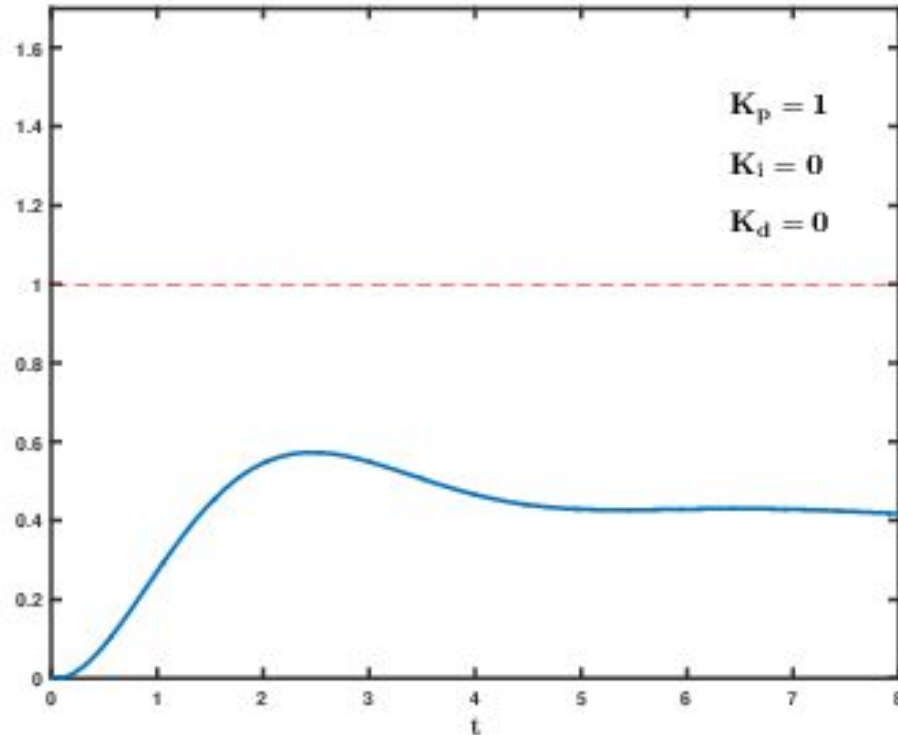
- How to get from point A to point B
- PID is one of the simplest ways to do this
- Used on drivetrain, arm (pivot), and elevator



# Programming » PID Control (cont.)

- Proportional
  - Correct for error - just get there
  - Can cause overshoot
- Integral
  - Correct for accumulation of error
  - “Integral windup”
- Derivative
  - Prevent overshoot/oscillation
  - “Dampen” system
  - Can slow response time

# Programming » PID Control (cont.)



# Programming » Lemonscript

- Domain Specific Language written by one of our mentors
- Used to tell the robot what to do during autonomous routines
- User defined C++ functions called on runtime
- Has while loops, etc
- Variables Coming Soon™

# Programming » Lemonscript (cont)

CheckArmCalibration

SetArmPosition: 4

DriveStraight: 14.0

SetArmPosition: AUTO 3

PointTurn: -15.0

Wait: 1.5

Align

Shoot

Wait: 0.6

AbsolutePointTurn: 0.0

DriveStraightAtAngle: -9.5, 0.0

# Programming » Vision

- 2016 game has small targets, hard to aim
- Computer vision is used to automatically align
- Camera + LED ring
- Process and get angle on driver station currently, but will move to coprocessor
- Find targets/shape detection with OpenCV

# Programming » Vision



# Any Questions?

