Advanced Controls Using WPILib

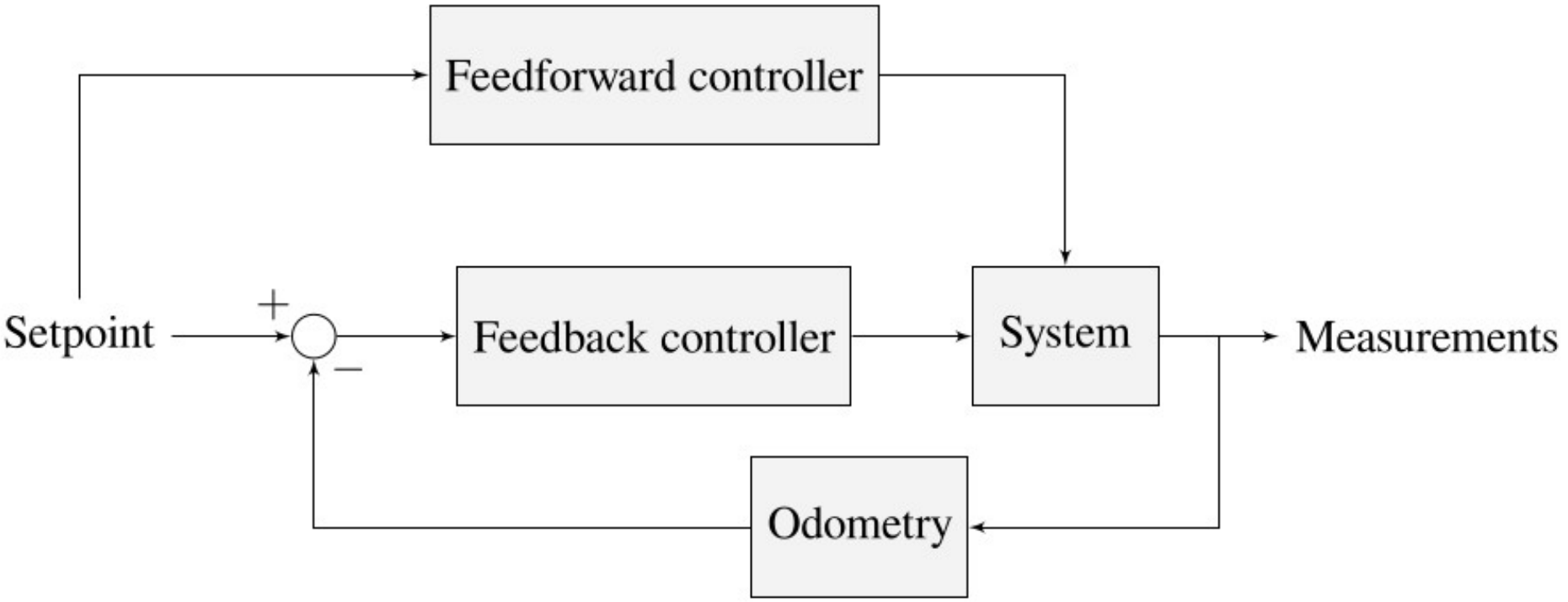
Tyler Veness

WPILib
Development Team

Target zone 6 ball autonomous

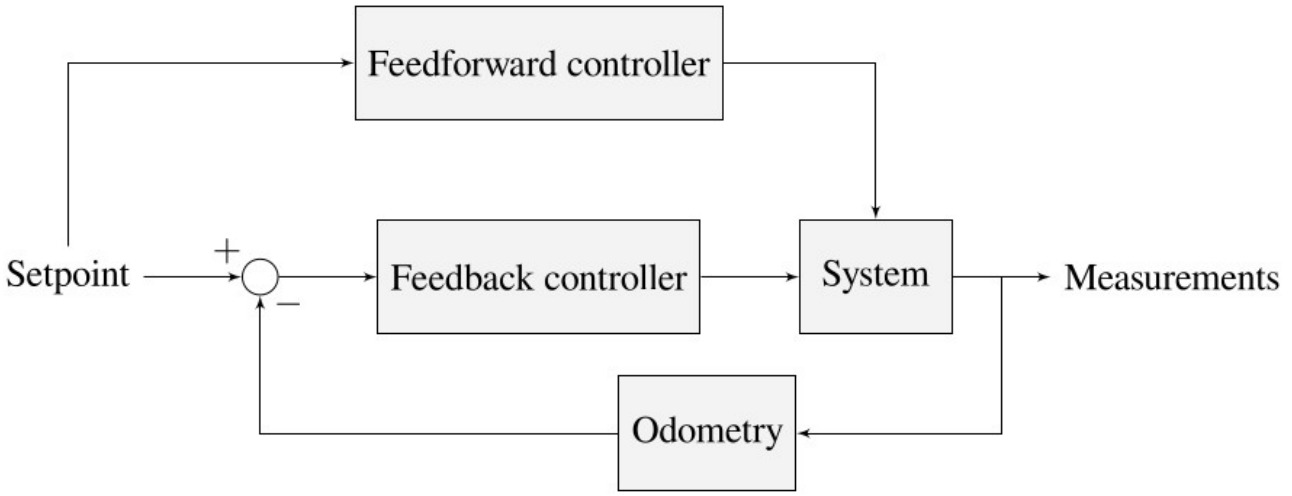
WPILib
Development Team

System overview



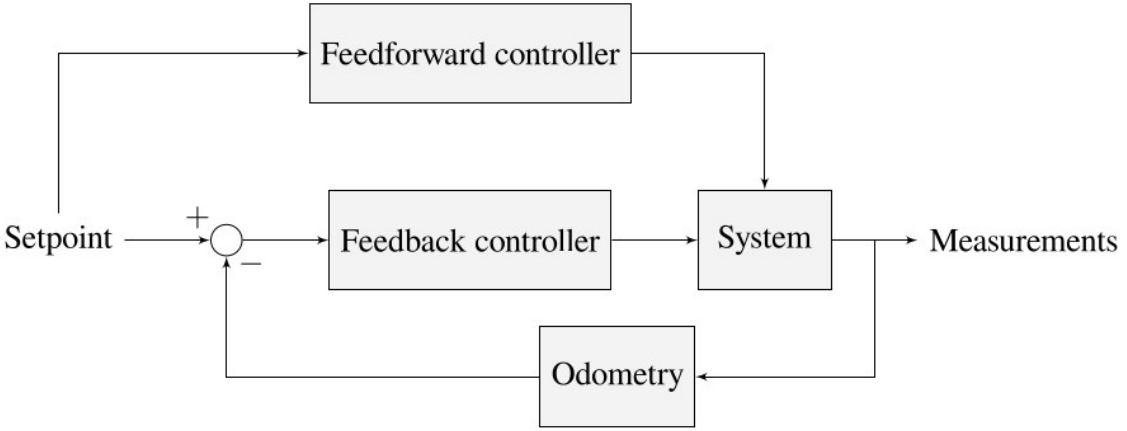
Controls engineering in FRC

- **Modeling** - equations representing how the robot behaves
- **Localization** - where the robot is in 2D
- **Motion planning** - planning how to get from here to there consistently
- **Control** - actually getting there
- WPILib has features to make these easier!



Modeling

- What?
  - Equations that take in voltage and output sensor measurements over time
- Why?
  - More accurate drivetrain simulations
  - Controller autotuning with minimal trial and error

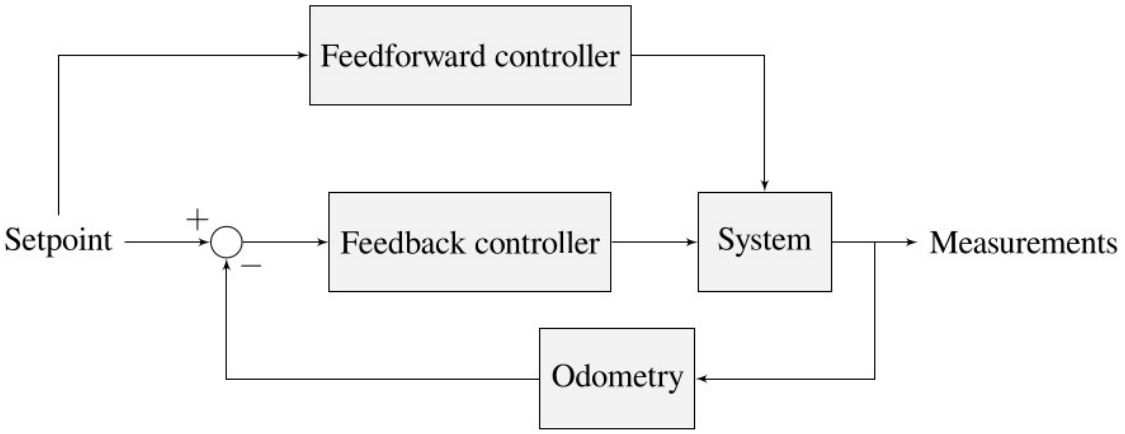


Modeling

- System identification
  - Use measured data from encoders and gyroscope to make a model that fits
- frc-characterization
  - Gathers data to make a feedforward model
  - https://docs.wpilib.org/en/stable/docs/software/wpilib-tools/robot-characterization/
- LinearSystemId functions
  - Makes a more general dynamical model for control later

Localization

- Use external measurements to obtain robot's position and heading on the field
- Why?
  - Dead reckoning with encoders alone accumulates error
  - If you can measure it, you can compensate for it with a controller

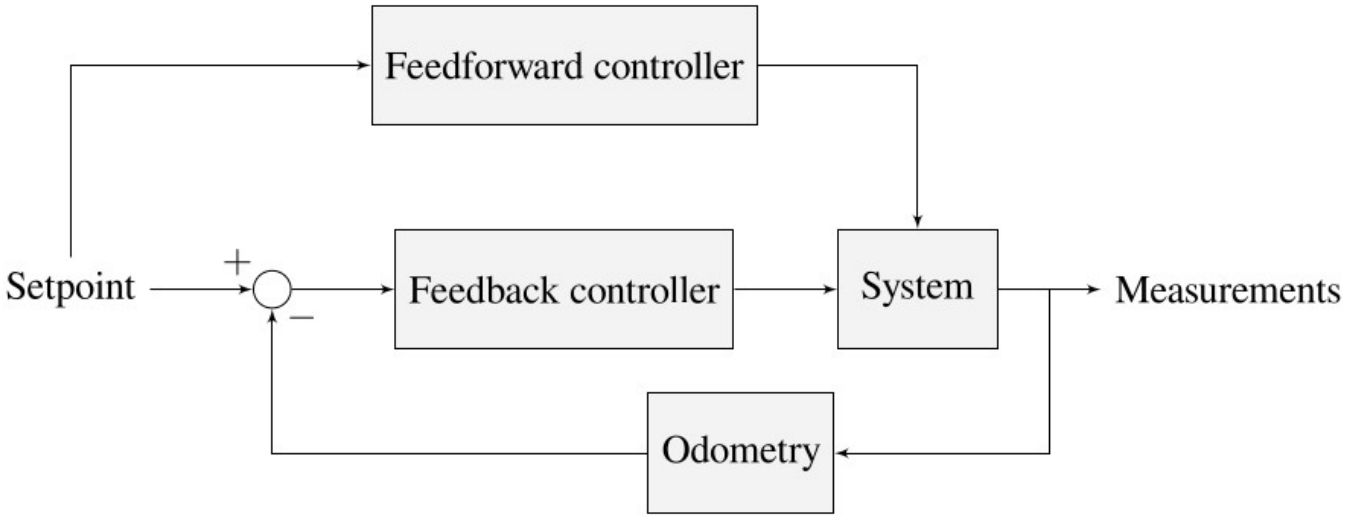


Localization

- DifferentialDriveOdometry class
  - Inputs: wheel encoders, gyroscope heading
  - Outputs: Robot position and heading
  - https://docs.wpilib.org/en/stable/docs/software/kinematics-and-odometry/differential-drive-odometry.html
- DifferentialDrivePoseEstimator class
  - Drop-in replacement for odometry class that incorporates vision measurements
  - Compensates for CV pipeline delays!
  - https://docs.wpilib.org/en/latest/docs/software/advanced-controls/state-space/state-space-pose_state-estimators.html

Motion planning

- Why?
  - Smooth, predictable motion over time
  - Only change the setpoint as fast as the system is able to physically move
  - Allows better open-loop control



1 DOF motion profiles

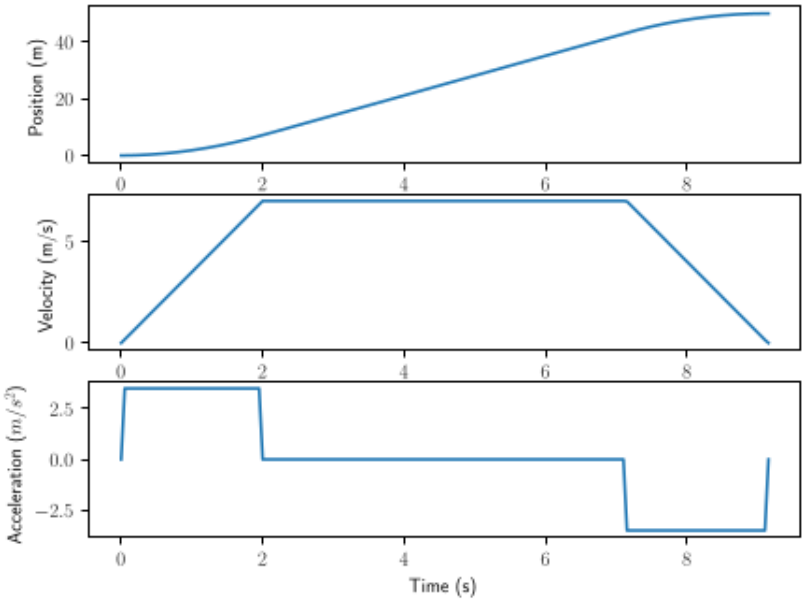


Figure 15.1: Trapezoidal profile

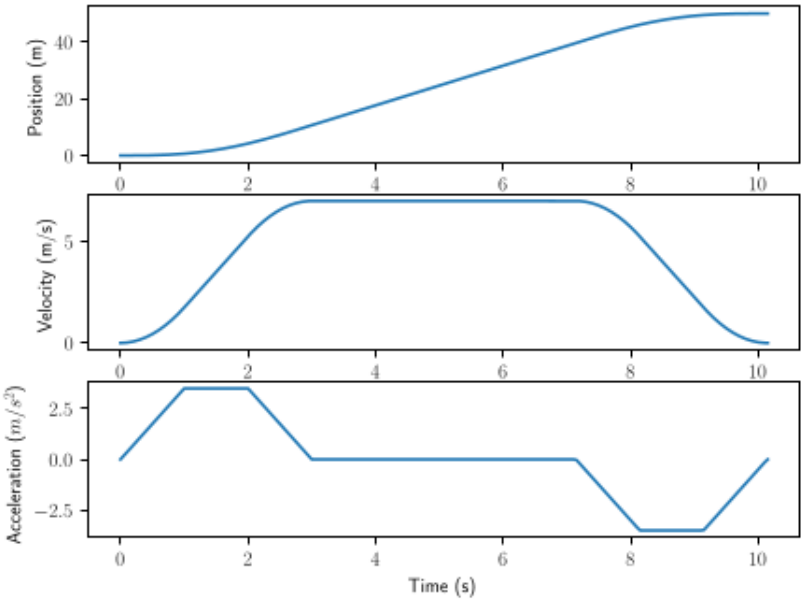


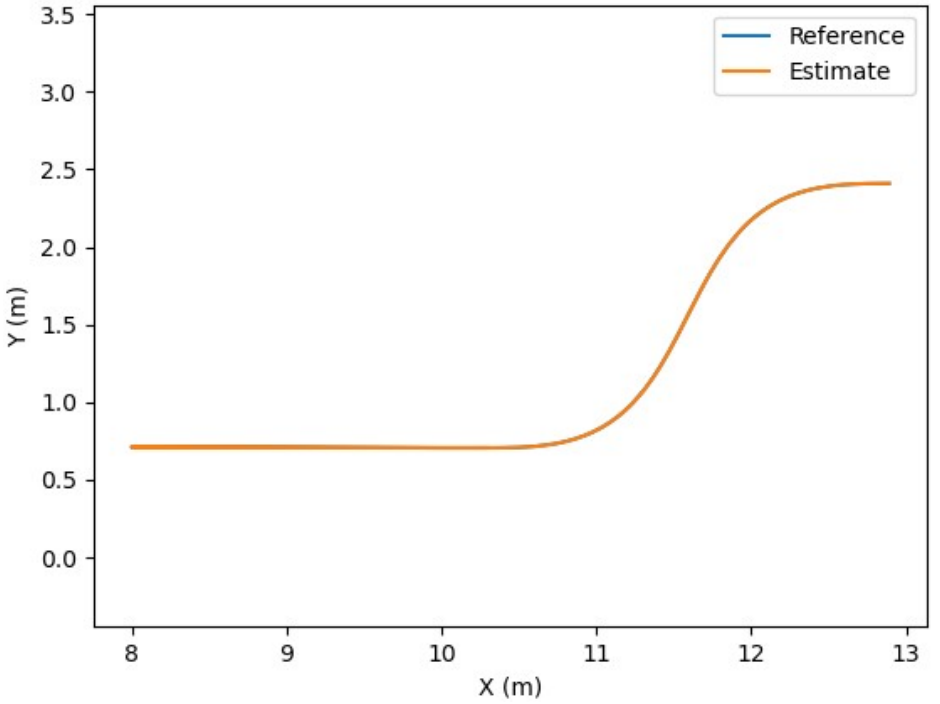
Figure 15.2: S-curve profile

1 DOF motion profiles

- TrapezoidProfile class
  - https://docs.wpilib.org/en/stable/docs/software/advanced-control/controllers/trapezoidal-profiles.html
- ProfiledPIDController class
  - https://docs.wpilib.org/en/stable/docs/software/advanced-control/controllers/profiled-pidcontroller.html

2 DOF motion profiles

- Degrees of freedom for drivetrain are x and y axes
- Trajectory includes:
  - X-Y path
  - Wheel velocities

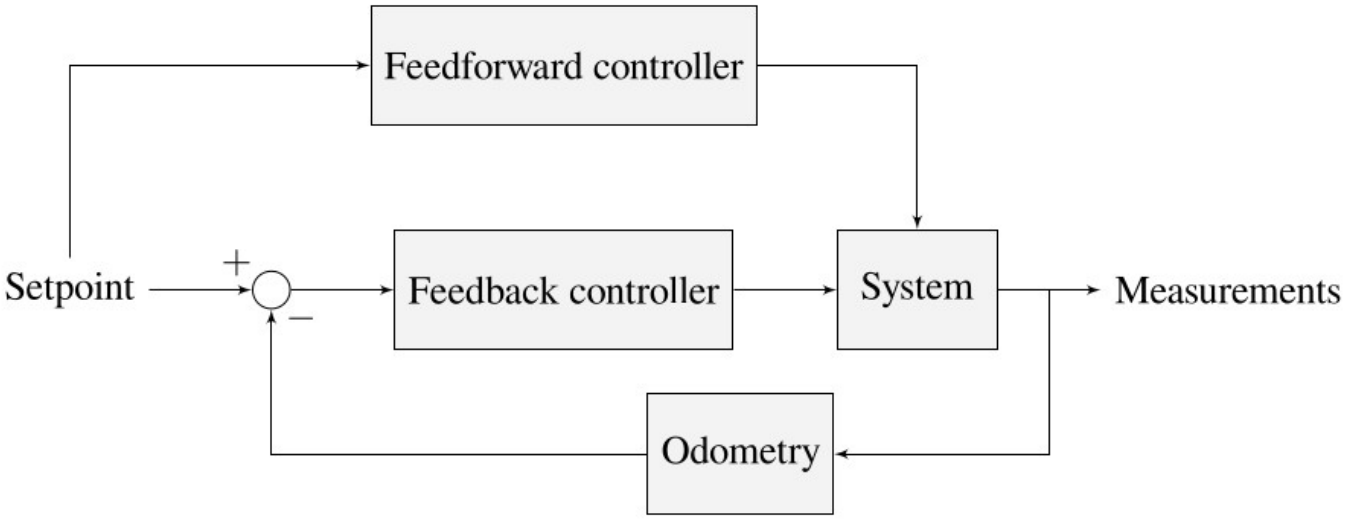


2 DOF motion profiles

- TrajectoryGenerator class
  - https://docs.wpilib.org/en/latest/docs/software/examples-tutorials/trajectory-tutorial/index.html
  - https://docs.wpilib.org/en/latest/docs/software/advanced-controls/trajectories/troubleshooting.html

Control

- Why?
  - Make robot respond how we want it to
  - Automate robot maneuvers
  - Make robot robust to disturbances and uncertainty



Feedforward controllers

- Plant inversion
  - We know how the system behaves and how we want it to behave, so find a voltage that makes it behave that way
  - SimpleMotorFeedforward, ElevatorFeedforward, etc.
  - https://docs.wpilib.org/en/stable/docs/software/advanced-control/controllers/feedforward.html
- Unmodeled dynamics
  - Gravity compensation, gearbox friction

Feedback controllers

- Ramsete
  - Nonlinear control law for field position
  - RamseteController class or RamseteCommand class
- PID controller
  - Commanded by Ramsete to go to wheel velocities
  - PIDController class or PIDSubsystem/PIDCommand
- Linear-quadratic regulator
  - P controller, but with gains chosen by math
  - LinearQuadraticRegulator class
  - Can use the model we got from frc-characterization!

More resources

- WPILib projects
  - https://github.com/wpilibsuite/allwpilib
- WPILib documentation
  - https://docs.wpilib.org/en/stable/
- My book on controls engineering in FRC
  - https://controls-in-frc.link/