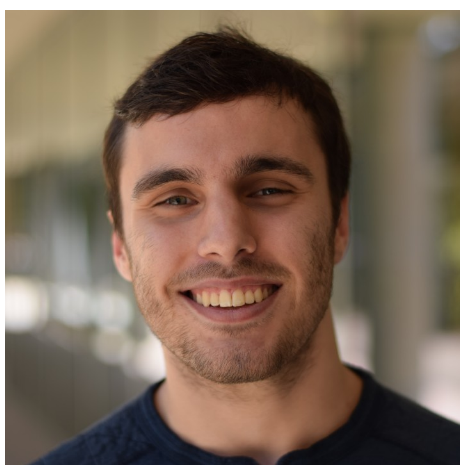
Model-based Validation of Autonomous Driving Using WPILib

Tyler Veness

WPILib
Development Team

Your speaker

- **2012 – 2013**: Student on FRC team 3512
- **2013 – present**: 3512's software/controls mentor
- **2018**: B.S., Robotics Engineering (UC Santa Cruz)
- **2015 – present**: WPILib developer



Autonomous driving in FRC

- No-op :(
- Time-based
- Trapezoid profiles with point-turns
- Spline-based trajectories

Goals

- Introduce localization, motion planning, and control
- Implement them using WPILib 2020
- Simulate it against FRC team 3512's 2020 robot

Localization

WPILib
Development Team

What is localization?

- Using external measurements to obtain agent pose and orientation

Odometry – Euler integration

$$x_{k+1} = x_k + v_k \cos\theta_k \, T$$
$$y_{k+1} = y_k + v_k \sin\theta_k \, T$$
$$\theta_{k+1} = \theta_{gyro,k+1}$$

where T is the sample period. This odometry approach assumes that the robot follows a straight path between samples (that is, $\omega = 0$ at all but the sample times).

Odometry – Pose exponential

- Exponential map from encoder distance measurements to global pose

$$
{}^{G}\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} = \begin{bmatrix} \cos\omega t & -\sin\omega t & 0 \\ \sin\omega t & \cos\omega t & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{R}\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} dt
$$

$$
{}^{G}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{R}\begin{bmatrix} \frac{\sin\Delta\theta}{\Delta\theta} & \frac{\cos\Delta\theta - 1}{\Delta\theta} & 0 \\ \frac{1-\cos\Delta\theta}{\Delta\theta} & \frac{\sin\Delta\theta}{\Delta\theta} & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{R}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix}
$$

Controls

WPILib
Development Team

What is control theory?

- An application of algebra and geometry used to:
  - Analyze and predict the behavior of systems
  - Make them respond how we want them to
  - Make them robust to disturbances and uncertainty

Types of controllers

- Feedforward
  - Plant inversion
  - Unmodeled dynamics
- Feedback
  - Ramsete
  - PID controller
  - Linear-Quadratic regulator

Feedforward – Plant inversion

Theorem 5.10.2 — Linear plant inversion. Given the discrete model $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$, the plant inversion feedforward is

$$\mathbf{u}_k = \mathbf{B}^\dagger(\mathbf{r}_{k+1} - \mathbf{A}\mathbf{r}_k) \tag{5.15}$$

where $\mathbf{B}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{B}$, $\mathbf{r}_{k+1}$ is the reference at the next timestep, and $\mathbf{r}_k$ is the reference at the current timestep.

Feedforward – Unmodeled dynamics

- Examples
  - Elevator or single-jointed arm gravity
  - Gearbox friction

Feedback – PID controller

Definition 2.4.1 — PID controller.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de}{dt} \tag{2.4}$$

where K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, $e(t)$ is the error at the current time t, and τ is the integration variable.

Figure 2.8 shows a block diagram for a system controlled by a PID controller.

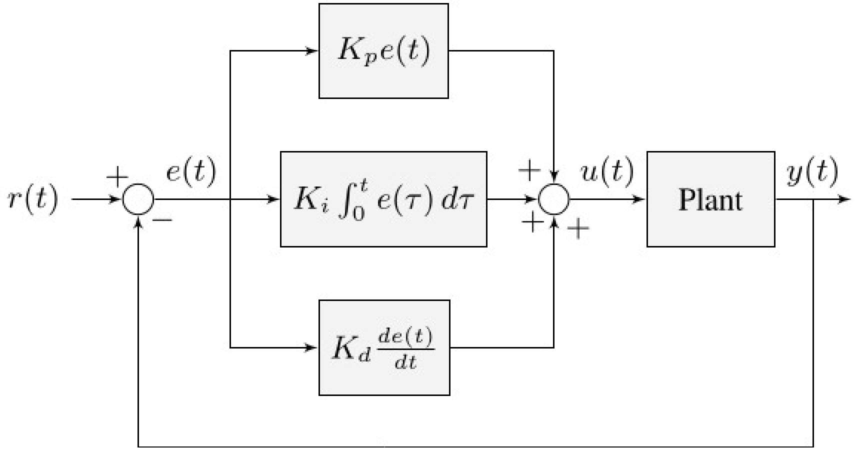


Figure 2.8: PID controller block diagram

Feedback – Ramsete

- Nonlinear control law for global pose

Feedback – Ramsete

Theorem 8.6.2 — Ramsete unicycle controller.

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} \tag{8.34}$$

$$v = v_d \cos e_\theta + k e_x \tag{8.35}$$

$$\omega = \omega_d + k e_\theta + b v_d \operatorname{sinc}(e_\theta) e_y \tag{8.36}$$

$$k = 2\zeta\sqrt{\omega_d^2 + b v_d^2} \tag{8.37}$$

$$\operatorname{sinc}(e_\theta) = \frac{\sin e_\theta}{e_\theta} \tag{8.38}$$

v	velocity command	v_d	desired velocity
ω	turning rate command	ω_d	desired turning rate
x	actual x position in global coordinate frame	x_d	desired x position
y	actual y position in global coordinate frame	y_d	desired y position
θ	actual angle in global coordinate frame	θ_d	desired angle

b and ζ are tuning parameters where $b > 0$ and $\zeta \in (0, 1)$. Larger values of b make convergence more aggressive (like a proportional term), and larger values of ζ provide more damping.

Motion planning

WPILib
Development Team

Why motion planning?

- Smooth, predictable motion over time is desired
- Only change the reference as fast as the system is able to physically move
- Allows better feedforward control

1 DOF motion profiles

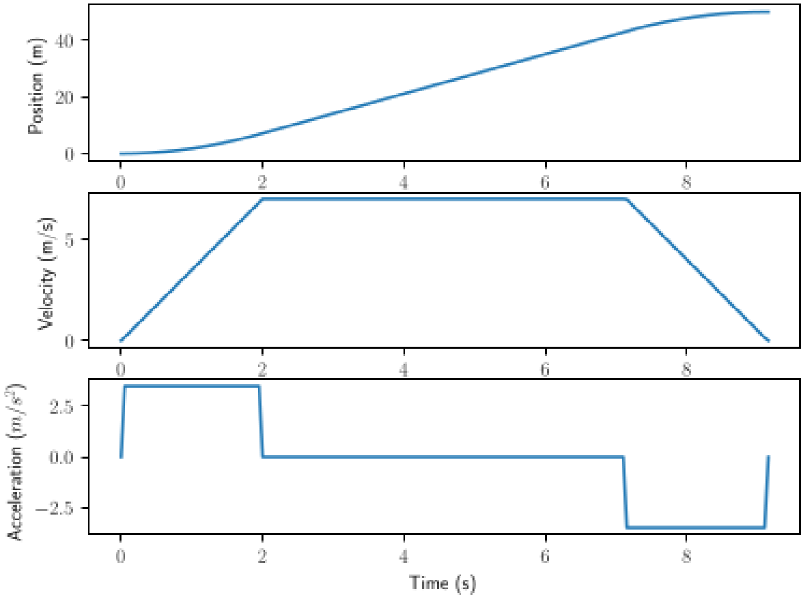


Figure 15.1: Trapezoidal profile

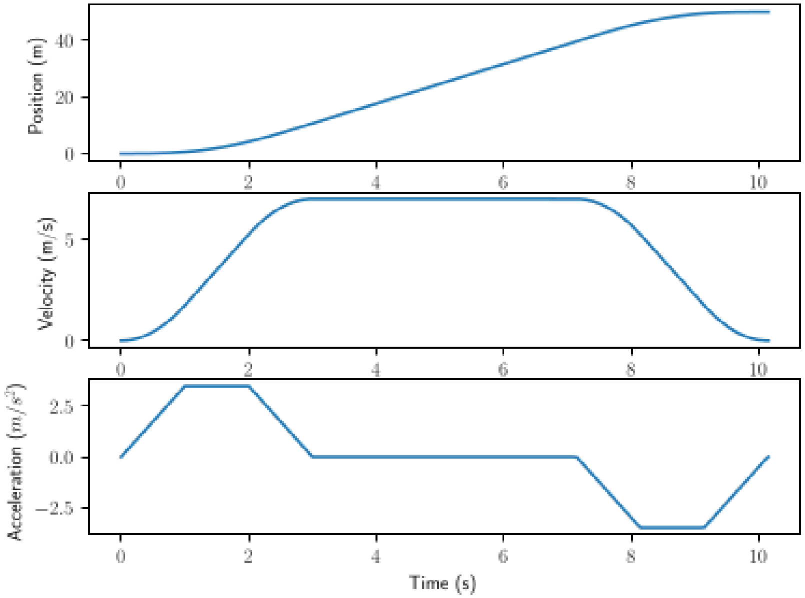


Figure 15.2: S-curve profile

2 DOF motion profiles

- Degrees of freedom for drivetrain are x and y axes
- Path
  - A set of (x, y) points for the drivetrain to follow
- Trajectory
  - A path that includes the states (e.g., position and velocity) and control inputs (e.g., voltage) of the drivetrain as functions of time

Clamped cubic splines

- Use splines to fit path
  - Exterior waypoints are poses
  - Interior waypoints are translations
- Time parameterize with velocity trapezoid profile
- More information
  - https://pietroglyph.github.io/trajectory-presentation

Robot Code Demo

Questions?

- WPILib projects
  - https://github.com/wpilibsuite/allwpilib
- WPILib documentation
  - 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
- My book on controls engineering in FRC
  - https://tavsys.net/controls-in-frc