Estimating Vehicle Kinematics Using Onboard IMU and Wheel Speed Sensors

Investigation of Actuator and Measurement Delay

Ross Alexander
Rushil Goradia
Trey Weber

6.1.2021

Problem Description

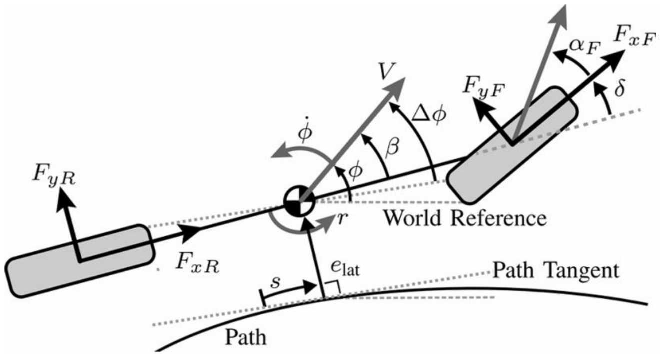
Design an efficient filter architecture for a vehicle that can estimate vehicle states using only an IMU and wheel speed sensors subject to actuator and measurement delays

- Rapid deployment of autonomous vehicles to public roads has made safety essential
- Accurate state estimates are required for safe and effective path-following control
- Market for autonomous vehicles may not support use of expensive, high-precision sensors using GPS, GNSS
- Cheaper sensors like IMUs and wheel speed sensors can be used, but these signals must be filtered effectively in order to construct a suitable state estimate

Dynamics Model

Dynamics model

- Dynamics follow a dynamic bicycle model
- State vector is body-frame longitudinal velocity, lateral velocity, and yaw rate
- Control vector is steering angle and front and rear longitudinal tire forces*



$$\mathbf{x} = [U_x, U_y, r]^\top \qquad \mathbf{u} = [\delta, F_{xf}, F_{xr}]^\top$$

Path tracking

- The dynamic state vector and dynamics model can be augmented with a path state vector and path dynamics for path-tracking

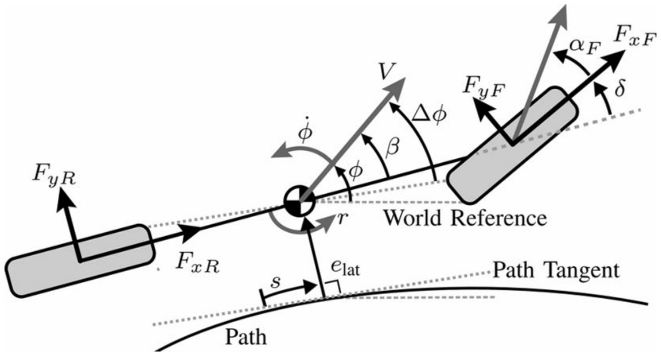
$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$$

$$\begin{bmatrix} \dot{U}_x \\ \dot{U}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(F_{xf}\cos\delta - F_{yf}\sin\delta + F_{xr}) + rU_y \\ \frac{1}{m}(F_{xf}\sin\delta + F_{yf}\cos\delta + F_{yr}) - rU_x \\ \frac{1}{I_z}(aF_{xf}\sin\delta + aF_{yf}\cos\delta - bF_{yr}) \end{bmatrix}$$

Measurement Model

Measurement model

- Measurement vector is rear wheel angular velocity and body yaw rate
- Sensors are rear wheel speed sensor and an inertial measurement unit (IMU)
- Assumes that wheel speed is equal to longitudinal velocity*



Path tracking

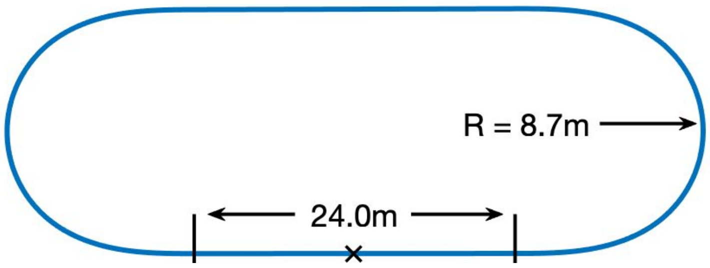
- Path state cannot be directly measured - it must be measured indirectly through dynamic states

$$\mathbf{y} = \begin{bmatrix} \omega_{wheel} \\ \omega_{z,GYRO} \end{bmatrix} = \begin{bmatrix} \frac{1}{r_{wheel}} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_x \\ U_y \\ r \end{bmatrix}$$

Experiments

Experiment setup

- Accelerate around an oval-shaped track with specified speed profile using realistic vehicle parameters and noise distributions
- Estimate dynamics state and path state using EKF, iEKF, UKF, and PF with and without actuator and measurement delay



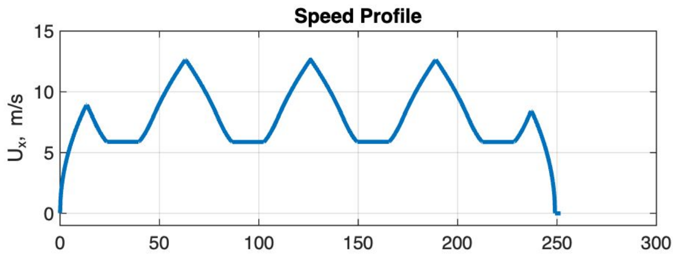
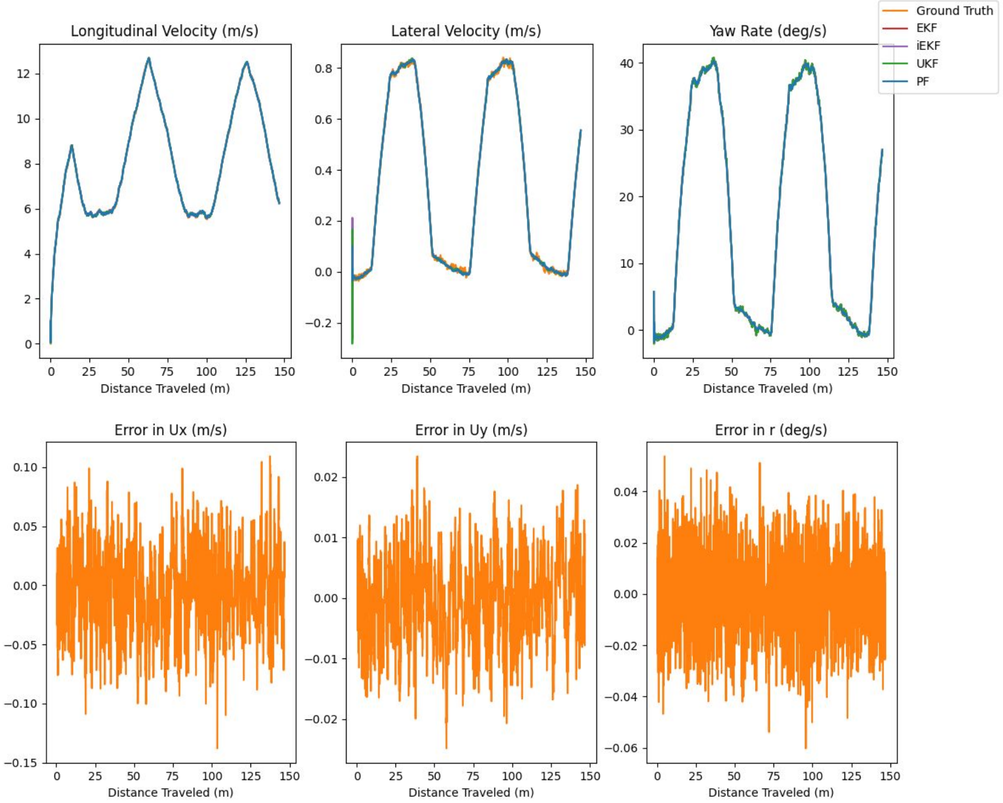


Image credit: ME 227

Results | Simulated Case

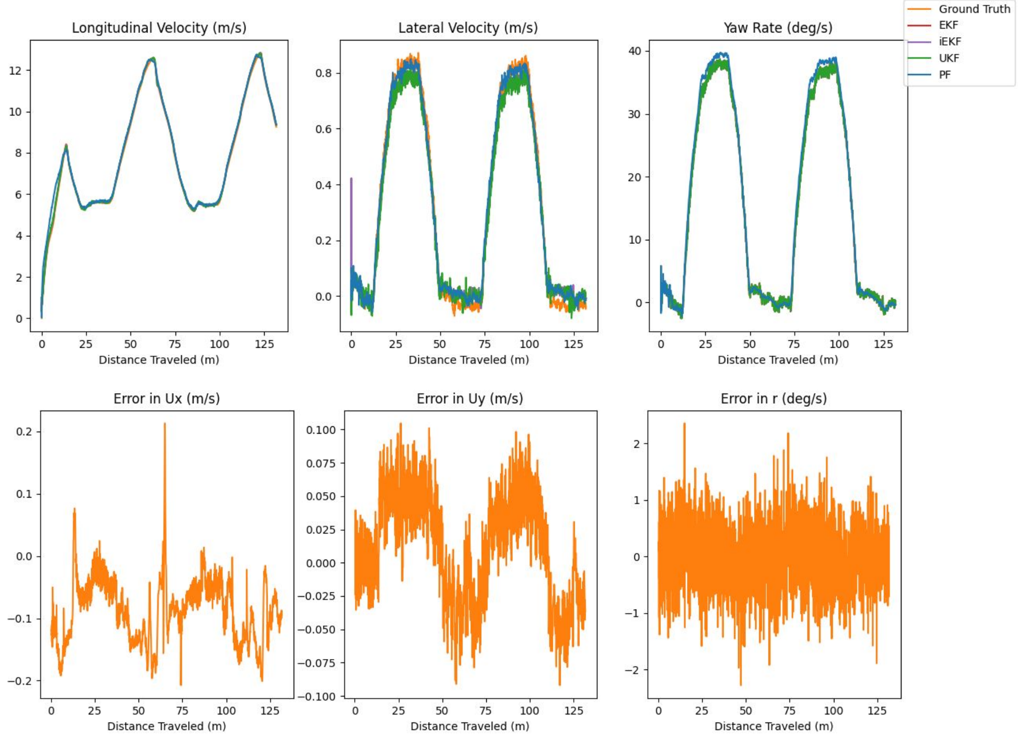
State trajectories are simulated and control inputs are specified through low-level feedforward and feedback controllers for steering angle and throttle/braking

Simulator has dynamic weight transfer, nonlinear tire force model, and actuator delay



Results | Real-World Case

State trajectories are estimated from SOTA navigation box (OxTS-RT3000, used as ground truth) and control inputs are recorded from onboard controller

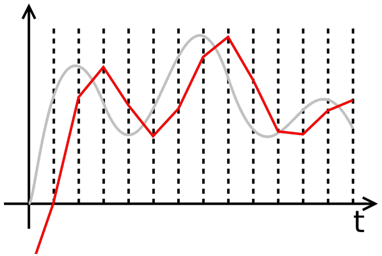


Data credit: ME 227

Future Directions

Model adjustments

- Variable/learned actuator delay
- Incorporation of measurement delay
- Use of measured forces/SWA instead of commanded (IMU)



Code adjustments

- Rewrite in CPP
- Optimize performance (reduce matrix inversions)
- Clean up and make open-source