Bike Dashboard

EECS 149/249A - 2019

Berkeley | EECS
Electrical Engineering and Computer Sciences

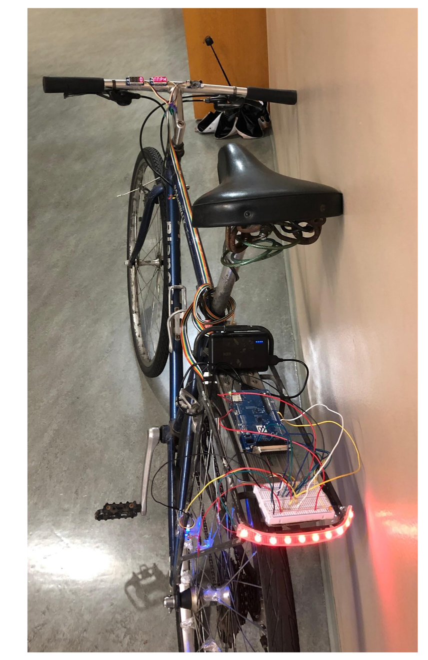
David Darevsky | Joseph Liu | Leland Yu

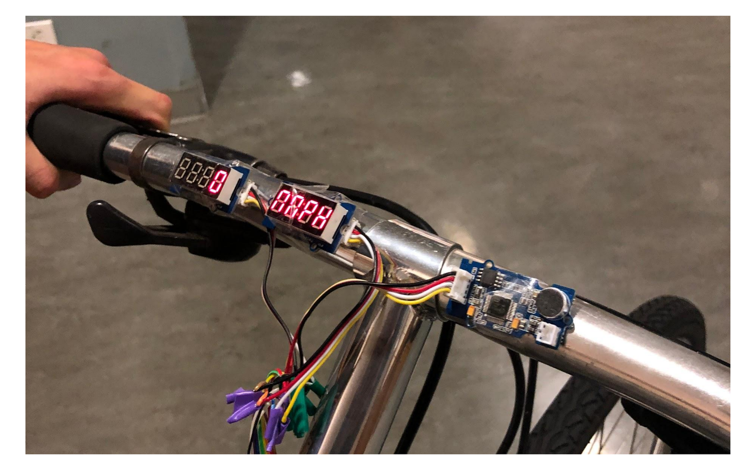
Project Purpose

Create a smart dashboard that helps improve the safety and experience of bike riders.

Features include

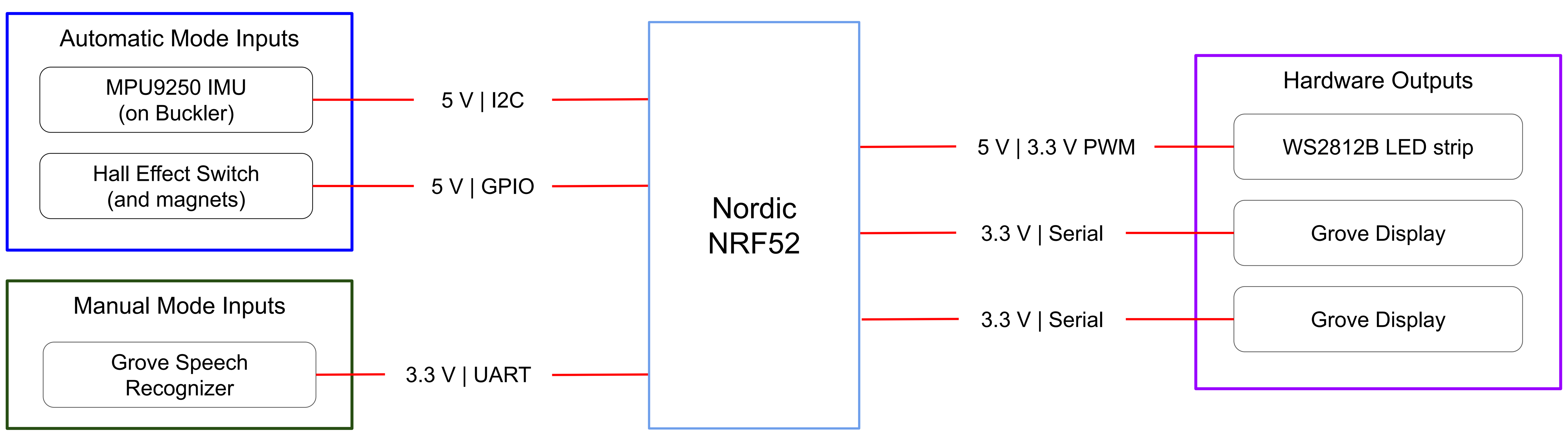
- Sensors for measuring bike kinematics including its speed and roll
- Voice recognition module and kinematics sensing automatically actuate signals
- LEDs attached to bike signal turns and braking
- Displays to show trip data such as current speed, total distance traveled, and temperature





Project Implementation

Hardware Architecture



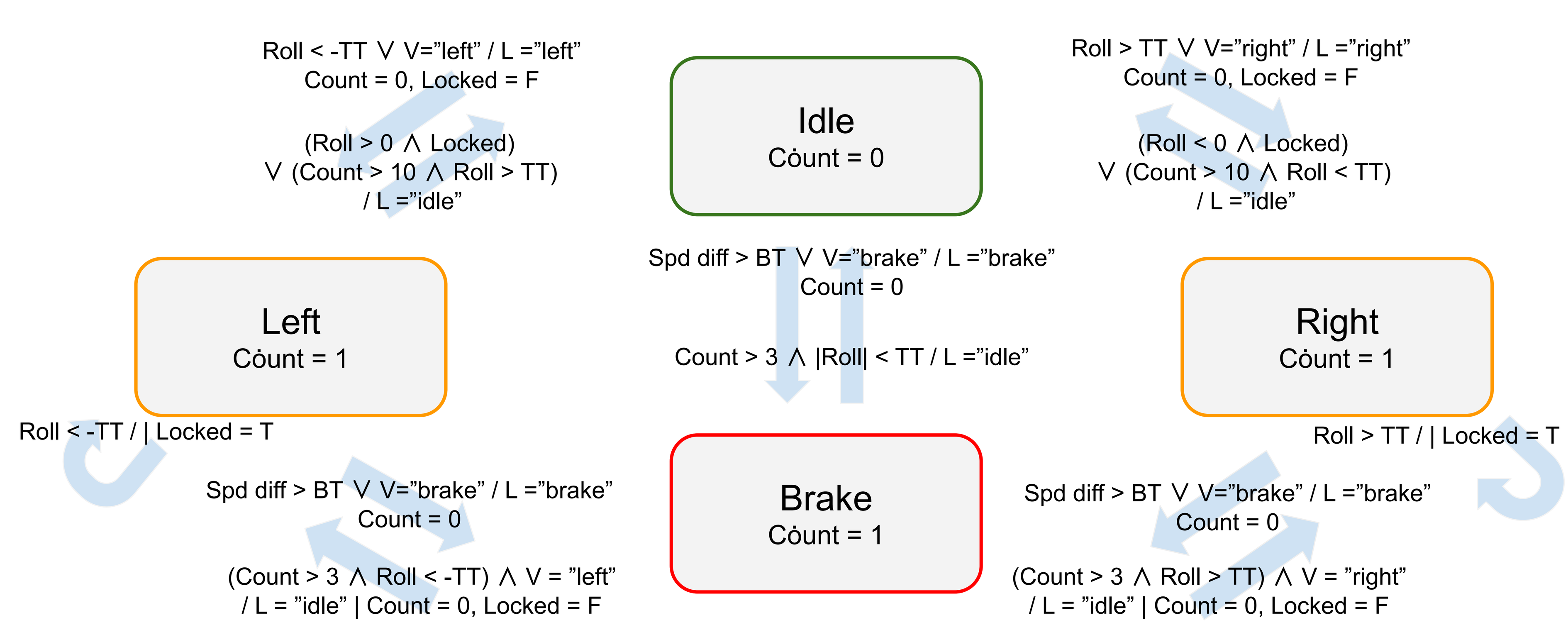
Software FSM

Inputs:
IMU: Roll ∈ R
Voice: V ∈ {"Left", "Right", "Stop", None}
Hall: Revolutions ∈ Z

Outputs:
Lights: L ∈ {"Left", "Right", "Idle", "Brake"}

State Variables:
Speed := Hall / Time difference (1 sec)
Speed Diff := Speed - Speed[-1]
Count: Timer
Turn Lock: Locked ∈ {T, F}

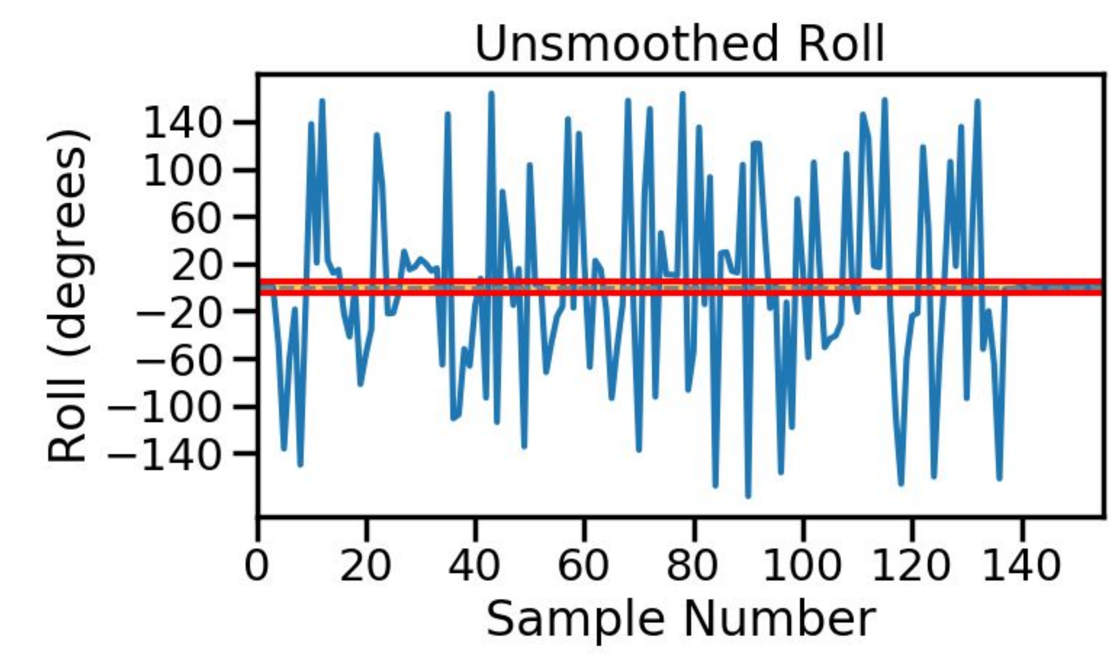
Constants: to be tuned
Turn_threshold (TT)
Brake_threshold (BT)

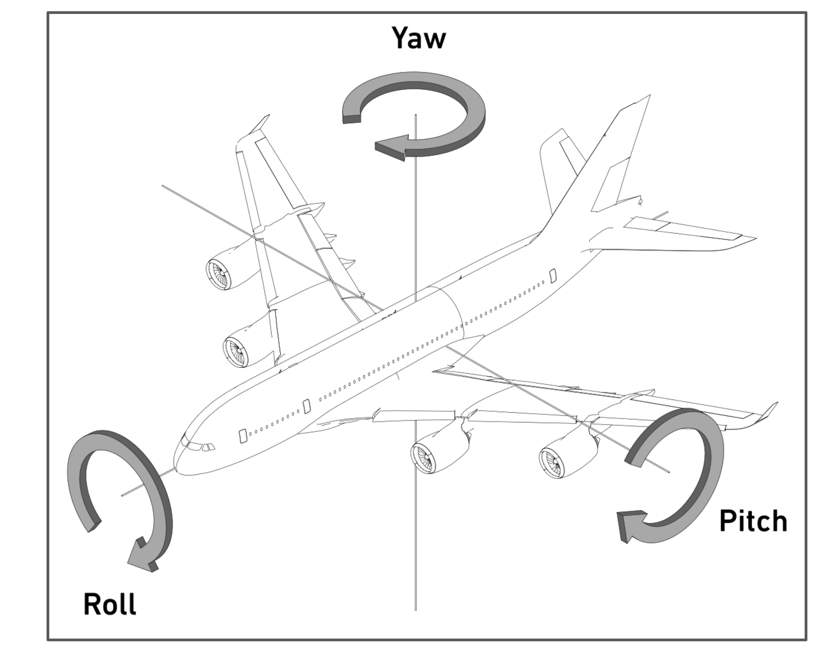


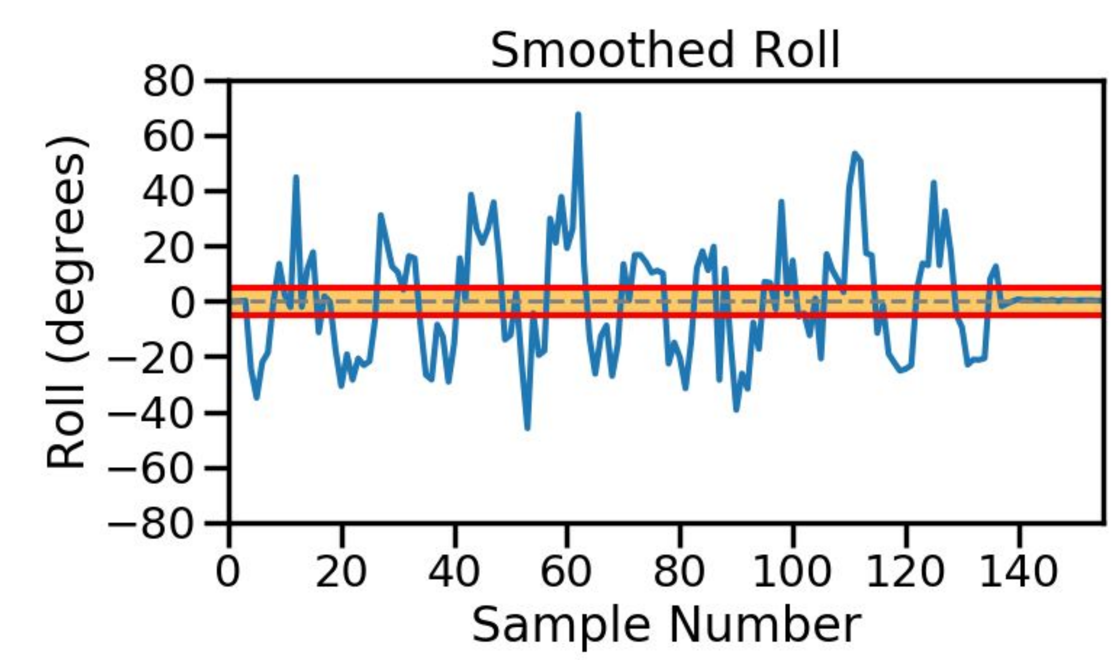
Design Evaluation

Challenge: Find the best signal for automatically detecting a bike's orientation independent of gravity and rider acceleration

Solution: Use the Madgwick AHRS algorithm to find absolute orientation







Turn/Brake Detection Performance	
Kinematics	8/10
Voice Detection	8/10

Voice Commands

- Left/Right – Activate Turn Signals
- Stop – Activate Brake Lights
- Next/Previous – Change Display Mode

Next Steps

- Build or 3D print a sturdier holder for the displays and sensors
- Incorporate audio feedback so the rider can stay focused on the road
- Acquire more robust IMU and voice recognition sensor
- Ultimate goal: develop Google Glass-like heads-up display

Course Connections

Sensors and Actuators

- Affine sensor model and calibration for IMU
- Voice Recognition Sensor
- Hall Effect Sensor

Interrupts & Polling

Communication Protocols

- Grove Display (UART)
- LED Strip (PWM)
- I2C Sensors

Finite State Machines

Linear temporal logic

- G(decelerate ⇒ Xbrake)