The background features a dark blue grid pattern. A white line graph with circular markers is overlaid on the grid, showing a fluctuating trend that generally increases from left to right. The title text is centered and rendered in a white, sans-serif font.

Smart Monitoring of Environmental Dynamics Using Smart Cars and Drones

Supervised by Dr. Yingying Chen

Team Members: Chen Hao Liu, Ryan Wang, Jeffrey Cheng, Justin
Wain, Masha Musthafa, David Yi



Jeffrey Cheng - Rising sophomore at Rutgers

Ryan Wang - Rising sophomore at University of Texas at Dallas

ChenHao Liu - Rising sophomore at George Washington University



Justin Wain - Rising sophomore at TCNJ

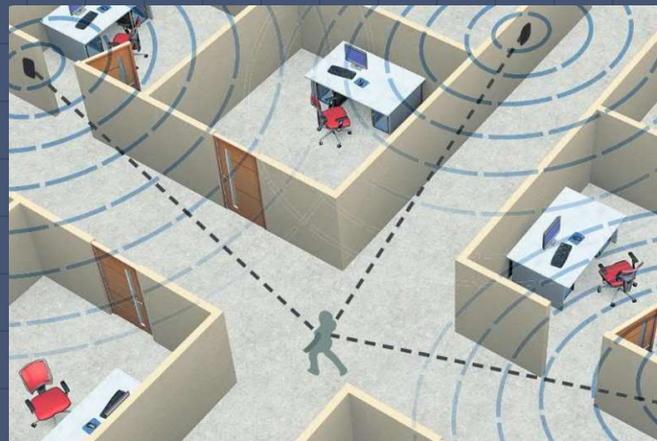


David Yi - High School Intern

Masha Musthafa - High School Intern

Project Motivations

- Movement is generally tracked using GPS
- Acquisition of a user's position and movement patterns is prevalent in mobile applications (fitness tracking, navigation, driving monitoring)
- Use of sensors on mobile device allows accurate estimation of position where GPS signals are weak, such as indoor or dense urban environments
- Our Goal: To achieve a sufficiently accurate method of localization using inertial sensors

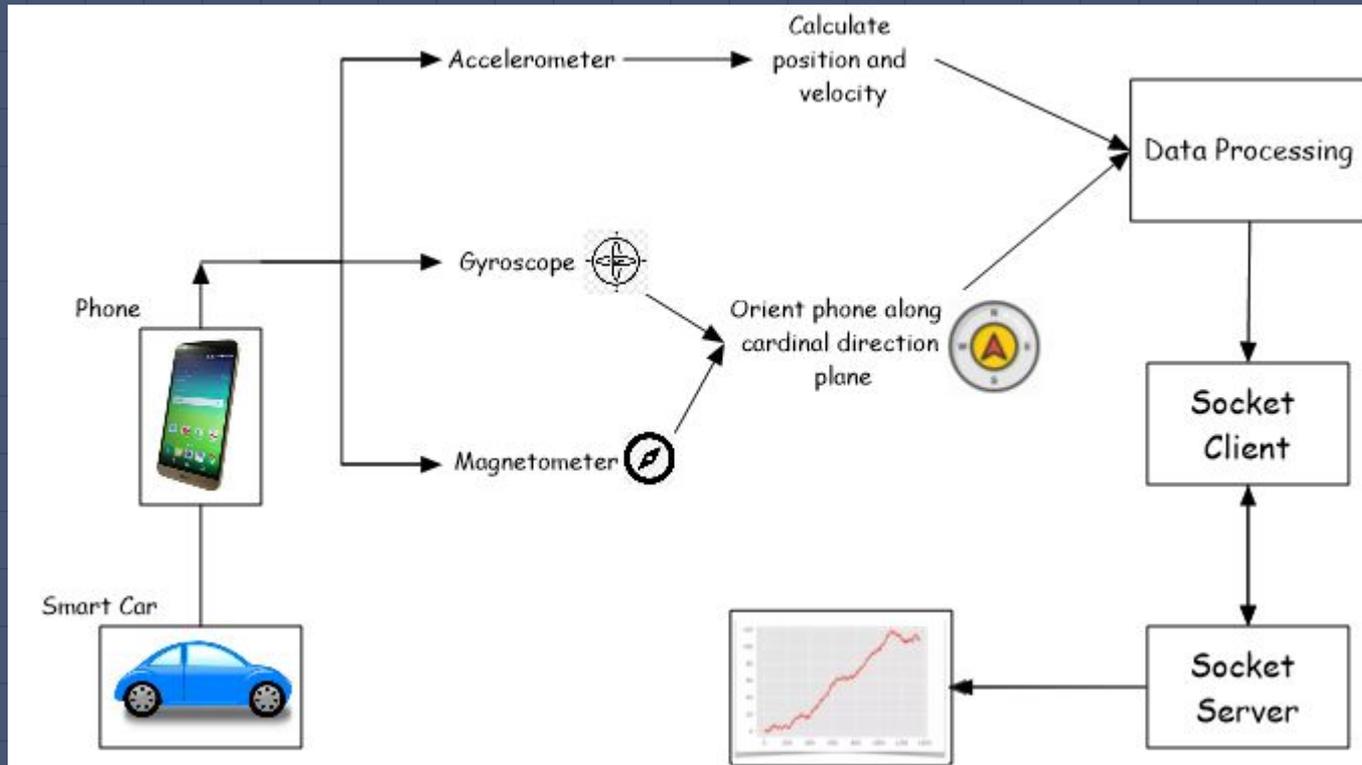


Why Use Smartphones?

- Devices are widely accessible and offer much utility
- Convenient to program with
- Portable and easy to use
- Eliminates need to install additional sensory equipment



System Overview



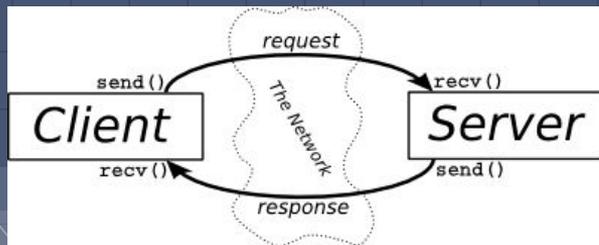
Smart Car Development/Android Studio

- Built the Arduino Smart Car Robot to simulate a moving vehicle
 - Features include moving in all directions, line tracking, and obstacle avoidance
- Used Android Studio to create our mobile application



Real Time Plotting/Data collection

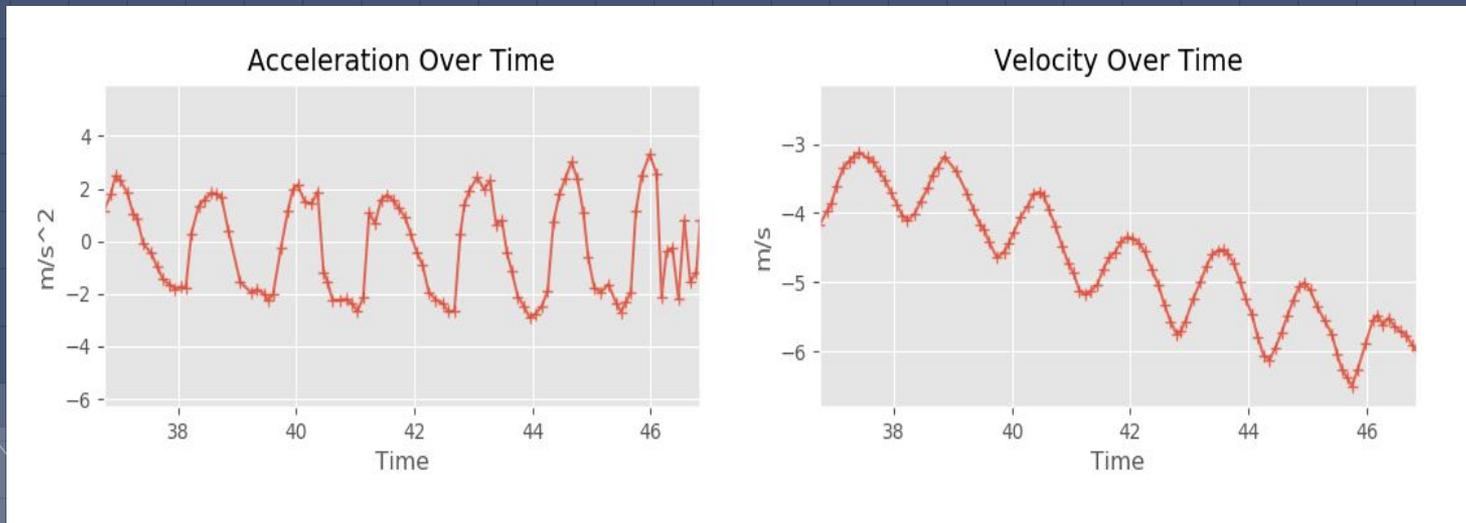
- Wireless connection between phone and server
- Laptop serves as the *socket server*
- Mobile device (*socket client*) transmits data over wifi
- Matplotlib (Python plotting library) algorithm graphs the data in real time



matplotlib

Problems Encountered

- Drift: Accumulation of error over time in velocity/displacement estimation due to inherent inaccuracies of the sensors
- Acceleration noise included in the integration for velocity, a small initial error increases error in velocity linearly over time

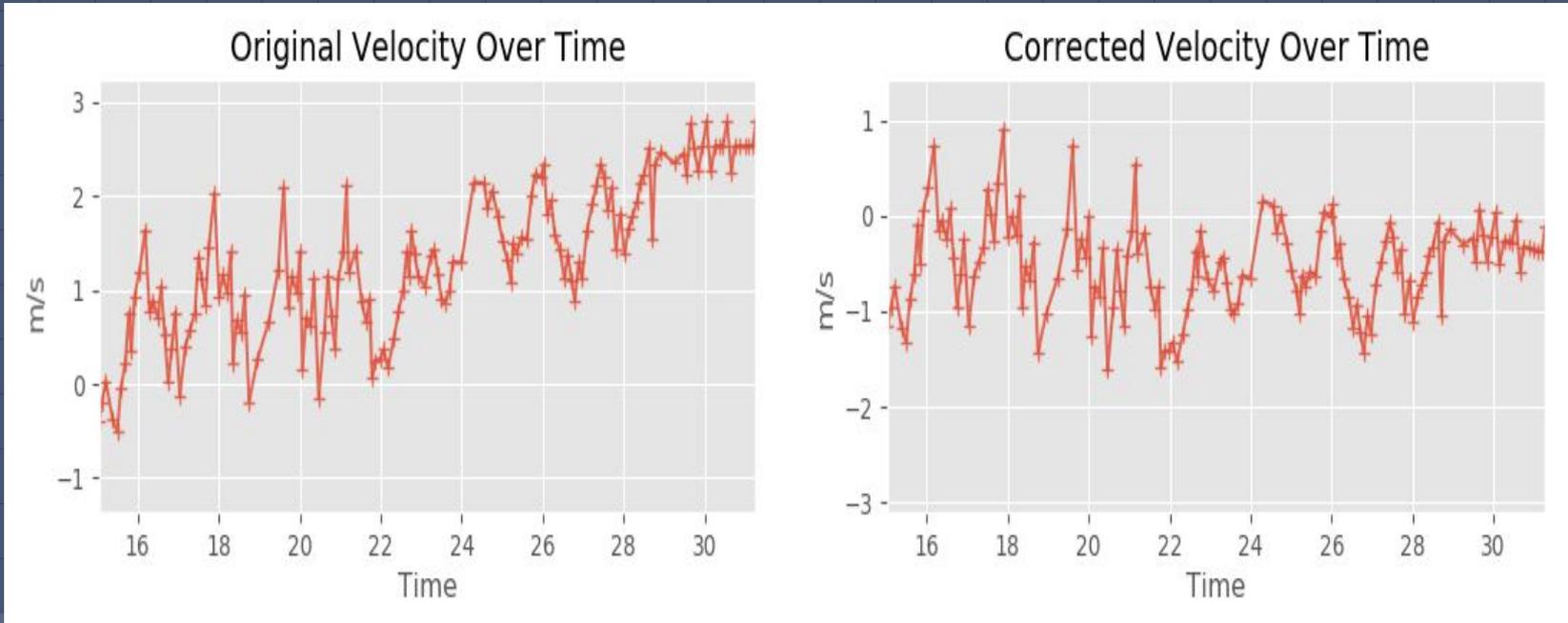


Drift Correction Methods

Inaccuracies in accelerometer data necessitate correction methods to increase precision of derived values:

- Low Pass Filter: Smooths out acceleration data by filtering out frequencies above a set cutoff frequency
 - Kalman Filter: Recursive algorithm for predicting velocity and displacement using acceleration data
 - Drift Correction: An algorithm that uses a series of reference points to calculate the continuously changing error in acceleration
- 

Correction Results



Conclusion and Future Work

- Completed a comprehensive system for velocity/displacement estimation
- More testing is needed to properly calibrate system for accurate data estimations
- Efficiency could be improved through better threading management, allowing higher sampling rate
- Upgrade system to accommodate for the 3D dynamics of a flying drone

Thank You!



Another Thank You

From all of us in this group, we would like to thank Professor Chen for everything she has done for our research group. We would also like to thank Richard Wang, Jian Chen, Yang Bai, and Xin Yang for being our mentors this summer!