GPS + Inertial Sensor Fusion

Functional Description & Complete System Block Diagram

By:

Aleksey Lykov,
William Tarpley, &
Anton Volkov

Project Advisors:

Dr. In Soo Ahn
Dr. Yufeng Lu

Date:

September 24, 2013

Table of Contents

I. Introduction

An inertial navigation system (INS) or inertial measurement unit (IMU) is a form of dead reckoning navigation system that uses a combination of accelerometer and gyroscope sensors working in concert to detect displacement relative to a starting point. The system measures both linear accelerations given by its three-axis accelerometer and angular velocity changes from the gyroscope measurements. World referenced-frame acceleration data can then be integrated to calculate the velocity and position of the sensors over time, but because the INS can only measure motion in relation to a starting location, the initial position must be supplied by some outside system (in the case of this project, a Global Position Sensor). Additionally, to compensate for the drift in the inertial navigation system caused by various defects in inertial sensors, this outside reference (GPS) must be polled occasionally to correct for the position error.

Using a Raspberry Pi microcomputer as the base system and an MPU 9150 IMU, an inertial navigation system will be developed. Kalman Filtering and GPS will be used to complete a "strapdown solution" – a closed-loop system which can self-correct for error [1].

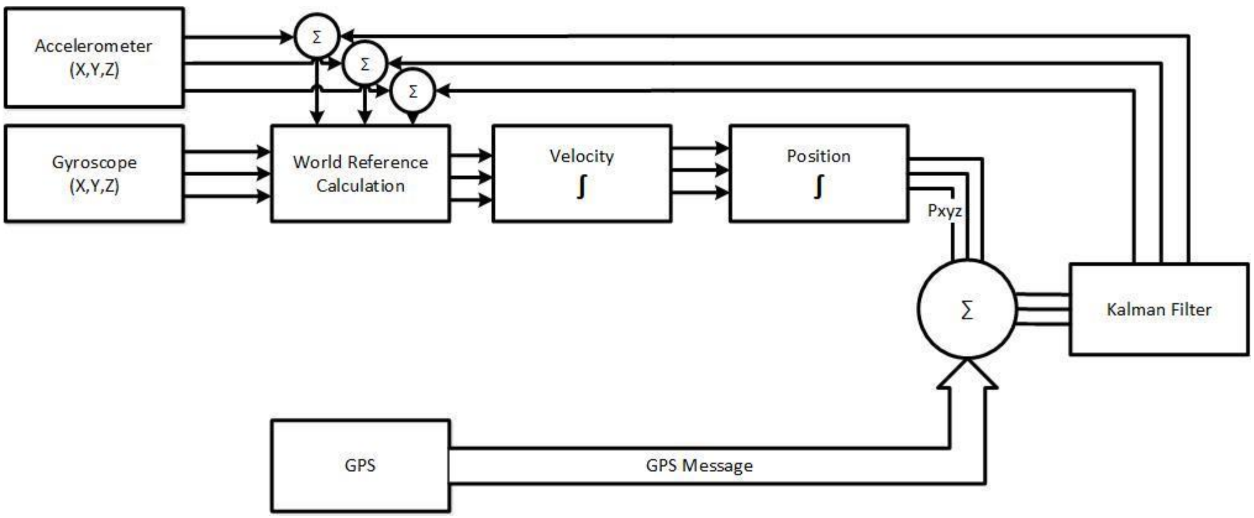
Junior Project Achievements:

- Raspberry Pi set up and operating system installed
- Cross-compilation in C and C++ set up between Windows and the Raspberry Pi's "Raspbian" Debian Linux-based OS
- File sharing via Samba between Windows and the Raspberry Pi
- Data acquisition with real-time nanosecond-resolution (less than nanosecond-accuracy) timestamps for IMU data acquisition at around 250 samples per second

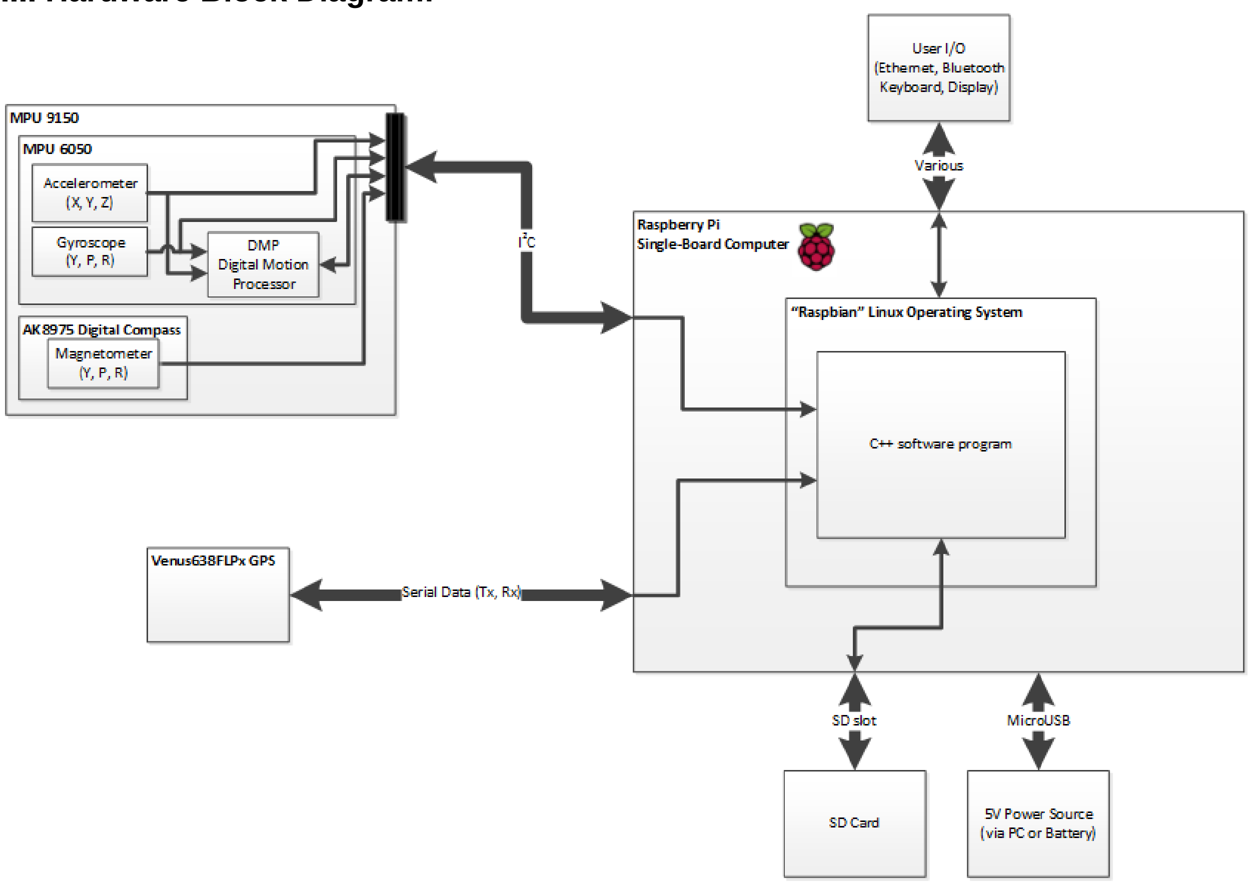
Current Project Goals:

- Design a Kalman filter which will auto-correct acceleration measurements used in position integration with outside displacement feedback
  - Determine the system model and state equations for Kalman filtering
  - Determine equations to relate calculated displacement from inertial measurements to the external position reference (provided by GPS)
- Secure a GPS unit to the Raspberry Pi single-board computer and combine the GPS and inertial data using precise timestamps
- Determine how long to keep past data for efficient memory management
- Implement a real-time Kalman filter onboard to complete the "strapdown" solution (as opposed to post-run Kalman filtering in MATLAB) for real-time inertial navigation, allowing a post-run log file of position data to be plotted on a map
- To enhance usability and marketability of the system, keep costs under $150, design with a "black box" solution in mind, and develop a simple user interface
- Final accuracy goals: report ≤5% error of displacement in X and Y directions after 60 seconds of navigation with or without GPS (for example, if 20m travelled in one minutes, report displacement within 1m of actual at the end of the run).

II. Signal Block Diagram:



III. Hardware Block Diagram:



IV. Functional Description:

The main sensor used for the project is the MPU 9150 Inertial Measurement Unit (IMU). This chip sends out nine axes of data: x-acceleration, y-acceleration, z-acceleration, yaw, pitch, roll, and three axes dedicated to magnetometer data. The three-dimensional acceleration and gyroscope data is used for data acquisition. Additionally, the chip has the capability to output the quaternion as well as real-world adjusted data which combines the data from the gyroscope and accelerometer to get real-time accelerations for yaw, pitch and roll. If needed, the magnetometer can be used to help calibrate the gyroscope to compensate for some of the drift.

A GPS unit will be used to give the program its initial position and to correct for error through the operation of a Kalman Filter.

The heart of the system will be the Raspberry Pi microcomputer. It runs C or C++ code for data acquisition and data processing. The IMU and the GPS are semi-permanently attached to the I/O pins of the Pi, which read in data from both devices after which the data is time stamped and saved to SD card. Eventually, the Pi will also perform real time filtering of the data instead of just data acquisition in order to calculate current system position.

Currently, data analysis is performed separately in MATLAB, manipulating data gathered by the system hardware onto the Pi's SD card.

V. Conclusion:

The completed system will be a functioning "strapdown solution", relying on Kalman filtering to correct body-fixed acceleration measurements to the navigation frame using the attitude computed with angular rates and external GPS position data. The updated acceleration data will be integrated into velocity and then into position data for the purpose of short-term movement tracking in areas where GPS is intermittent or unavailable.

VI. References:

[1] D. H. Titterton, and J. L. Weston. *Strapdown Inertial Navigation Technology*. Stevenage: Institution of Electrical Engineers, 2004.