

**2017 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY
SYMPOSIUM
AUTONOMOUS GROUND SYSTEMS (AGS) TECHNICAL SESSION
AUGUST 8-10, 2017 - NOVI, MICHIGAN**

**ANVEL HIL/SIL ADDITIONS FOR NAVIGATION ALGORITHM
DEVELOPMENT**

Brently Nelson
Mechanical Engineering
Auburn University
Auburn, AL

David Bevly
Mechanical Engineering
Auburn University
Auburn, AL

Jeff Ratowski
Bernard Theisen
U.S. Army TARDEC
Warren, MI

ABSTRACT

This work presents the development of a high fidelity Simulation In the Loop/Hardware In the Loop simulation environment using add-ons to Autonomous Navigation Virtual Environment Laboratory (ANVEL) and a navigation unit developed by Auburn University's GPS and Vehicle Dynamics Lab (GAVLAB) in support of the United States Army's Autonomous Ground Resupply Science Technology Objective. The developed add-ons include a real time interface for ANVEL, Inertial Measurement Unit module, Wheel Speed Sensor module, and a GPS module that allows simulated signals or generated Radio Frequency signals. The developed add-ons allow for faster development of navigation algorithms and controllers due to a readily available, highly accurate truth from ANVEL and can be configured to introduce realistic errors from sensors, hardware, and GPS signals such that algorithm and controller robustness can be easily examined.

INTRODUCTION

The Autonomous Ground Resupply (AGR) Science Technology Objective (STO) will develop and demonstrate an improved ground supply distribution system across multiple levels of strategic and tactical sustainment operations. The effort will equip existing military ground vehicles with scalable robotic technology through the integration of modular kits, common interfaces, and a common architecture to improve inter-node supply movement. Further, the system will modernize and optimize the operations within the supply nodes to improve accountability and throughput.

A critical product of the AGR STO is an integrated suite of physics-based simulation tools to develop and refine concepts, test vehicle prototypes, evaluate design changes, determine technology performance, and predict outcomes in a wide variety of terrain, weather, and environmental conditions. High fidelity simulation environments are widely used in industry for rapid development of navigation and control algorithms. Hardware is commonly added to the simulation environments to form a Hardware In the Loop (HIL) simulation. HIL testing shortens development time, cuts cost, and reduces risk from in-vehicle testing. The work in this paper presents

the development of a high fidelity simulation environment with optional HIL testing modules for the development and evaluation of navigation and control algorithms for Unmanned Ground Vehicles (UGV).

SYSTEM ARCHITECTURE

The system architecture for this work is two fold. First, users can choose to run the system in a software only mode, known as Simulation In the Loop (SIL). In SIL mode, all signals from the various sensors are simulated, including Global Positioning System (GPS), Inertial Measurement Unit (IMU), and Wheel Speed Sensors (WSS). In the second mode of operation, users can incorporate HIL in order to add errors and behaviors inherent to hardware implementation. The hardware additions to the system include a Spectracom real-time GPS Radio Frequency (RF) signal generator, analog encoder pulses generated using a Digital to Analog Converter (DAC), and a serial interface for IMU measurements to be fed into a navigation unit. The navigation unit employed was developed by Auburn University's GAVLAB and contains a Honeywell IMU, KVH gyro, Novatel OEM 628 GPS receiver, and an onboard computer implementing an Extended Kalman Filter (EKF) to compute a navigation solution. The overall system architecture with hardware additions included can be seen in Figure 1.

The vehicle simulation is run on Autonomous Navigation Virtual Environment Laboratory (ANVEL), a rapid prototyping tool developed for U.S. Army Engineer Research and Development Center (ERDC), which specializes in UGV technologies and is used as a virtual proving ground for the development of new UGV technologies. The Simulation Personal Computer (SimPC) is only used for the ANVEL simulation and the ANVEL interface, which allows the system to import and export measurements to and from the simulation environment. The SimPC is connected via User Datagram Protocol (UDP) interface with the "addition" PC, which houses the IMU, WSS, and GPS measurement and visibility modules; however, these modules could be implemented on the SimPC if desired. The developed modules are designed to provide simulated sensor output that replicates actual sensor output as closely as possible. If the user desires the HIL system implementation to incorporate measurement errors, the Spectracom simulator, encoder pulse generation, IMU serial interface, and navigation unit may be implemented as seen in Figure 1. The software was also developed such that the navigation unit may be placed on a rate table to incorporate real IMU measurements, but this implementation is outside the scope of this paper. The development of each of the previously mentioned modules, as well as the ANVEL interface, is discussed in the following sections.

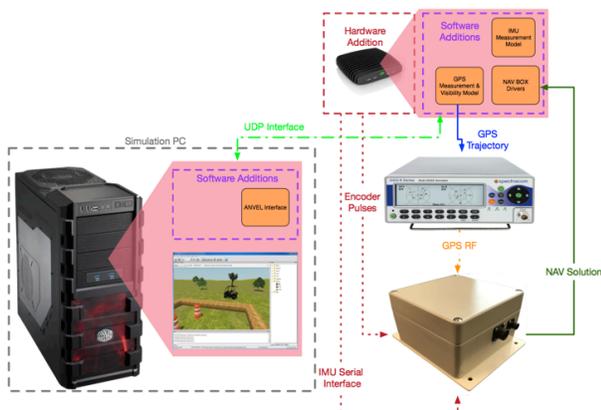


Figure 1: Overall System Architecture

ANVEL INTERFACE

The first step in implementing the HIL/SIL with ANVEL is developing an interface between ANVEL and the rest of the system. Communications are handled via a plugin for ANVEL which allows the user to extend ANVEL past the standard simulation environment and customize based upon needs. Users can add new terrain, sensors, buildings, or entirely new behaviors. The plugin developed for this work gathers vehicle state information and sensor

measurements from ANVEL in real time and makes them available for use by developers as required. For this work, vehicle state information and simulated, ideal sensor output from ANVEL are published by the SimPC as UDP packets. These packets are read by a Robot Operating System (ROS) node on the addition PC and published to a ROS database via standard and custom ROS messages. Employing the ROS framework allows researchers to easily manipulate the desired messages through its publish-subscribe system. Another plugin is employed to pass the navigation solution from the navigation unit back into ANVEL in order to test control algorithms. A ROS node subscribes to the desired messages (position, acceleration, heading, etc.) and writes these measurements to ANVEL, which may then use the measurements as input to a control algorithm. Control algorithms are out of the scope of this paper.

GPS MEASUREMENT AND VISIBILITY MODULE

The GPS measurement and visibility module is designed such that the output from the module is representative of the Novatel receiver used in hardware implementation. The GPS module calculates satellite trajectories using Receiver Independent Exchange (RINEX) downloaded from the Crustal Dynamics Data Information System (CDDIS) Global Navigation Satellite System (GNSS) archive [1]. The GPS module uses the calculated satellite trajectories and known object locations (buildings, trees, etc.) to employ ray tracing to determine which satellites are in view at any time, removing satellites blocked by environmental features. For ray tracing and satellite availability, ANVEL is provided with a three dimensional (3D) map of the vehicle's environment generated prior to simulation [2]. A visualization of the ray tracing capabilities can be seen in Figure 2.

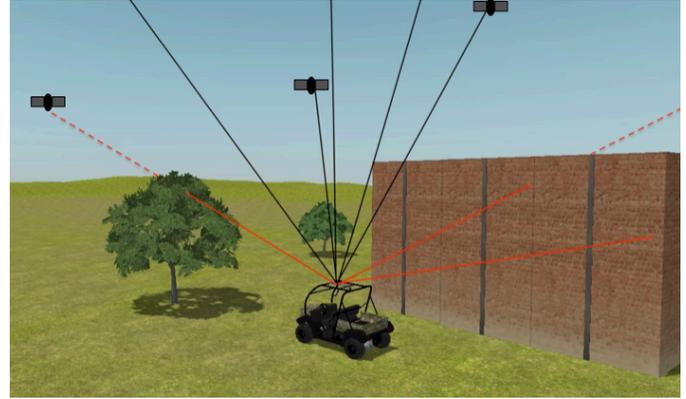


Figure 2: Ray Tracing Visualization

If the vector between the vehicle and satellite intersects an obstacle, that satellite is removed from the solution. Using the known locations of the satellite and true position of the vehicle, known from ANVEL, the module generates pseudorange, carrier phase, and Doppler measurements and publishes these to the ROS database along with the GPS position solution.

Atmospheric errors due to the Ionosphere and Troposphere are also added to the simulated GPS signals for a more realistic representation. Ionospheric and Tropospheric models are shown in Equations (1), (4), and (5) respectively [3].

$$I_{\rho} = \frac{40.3 * TEC}{f^2} \quad (1)$$

Where I_{ρ} is the Ionospheric group delay, subscript ρ denotes pseudorange, TEC is Total Electron Content, and f is the signal frequency, 1575.42 MHz for L1. TEC data is downloaded from [4]. An Obliquity Factor (OF) is used to account for the fact that signals from satellites at lower elevation levels must travel further through the atmosphere

$$OF(\zeta) = \left[1 - \left(\frac{R_E \sin \zeta}{R_E + h_I} \right)^2 \right]^{-1/2} \quad (2)$$

yielding

$$I(\zeta) = I_z * OF(\zeta). \quad (3)$$

$I(\zeta)$ is Ionospheric delay as a function of zenith delay, I_z , and OF. R_E is the average radius of the Earth and h_I is the mean ionospheric height.

The Troposphere may be modeled using the Saastamoinen Model. The model is split into dry and wet delays, due to the dry gases and water vapor present in the atmosphere as shown in Equations (4) and (5).

$$\tilde{T}_{z,d} = 0.002277(1 + 0.0026 \cos 2\phi + 0.00028H)P_0 \quad (4)$$

$$\tilde{T}_{z,w} = 0.002277 \left(\frac{1255}{T_0} + 0.05 \right) e_0 \quad (5)$$

Equations (4) and (5) represent the zenith dry and wet delays, $\tilde{T}_{z,d}$ and $\tilde{T}_{z,w}$, respectively, where T_0 is the temperature in Kelvin, P_0 is the total pressure, e_0 is the partial pressure due to water vapor, ϕ is latitude, and H is the orthometric height of the antenna. If the described environmental variables are known, Equations (4) and (5) may be used to calculate the dry and wet delays, respectively. Otherwise, Tropospheric data in the form of total zenith delay may be downloaded from the CDDIS [5] and is a combination of the dry and wet delays. The mapping function for the Tropospheric delay is modeled as

$$m(el) = \frac{1}{\sqrt{1 - (\cos el / 1.001)^2}} \quad (6)$$

where el is elevation angle.

IMU AND ENCODER MODULE

IMU and WSS modules are developed for both the SIL and HIL implementations of the system.

SIL IMU

For the SIL implementation, the IMU module is configured to output sensor measurements similar to those the navigation algorithm would receive from the on-board IMU. This includes errors typical of the sensors used in HIL implementation such as noise, bias, and drift. Simulations of the inertial sensors are developed from the simple IMU models below [6].

$$g_r = r + c_r + b_r + w_{gyro} \quad (7)$$

$$a_x = \ddot{x} + g \sin \phi + c_{\ddot{x}} + b_{\ddot{x}} + w_{accel} \quad (8)$$

The models represent the measured output of the yaw gyro (g_r) and lateral accelerometer (a_x) as a combination of the true output (r and \ddot{x}) plus a turn on bias or constant offset (c), a moving or walking bias (b), and wide band sensor noise (w_{gyro} and w_{accel}). The lateral acceleration equation includes a term to account for the effects of gravity (g) when the vehicle experiences roll (ϕ). Roll and velocity are assumed to be zero for the static data sets analyzed when characterizing sensors. The wide band sensor noise (w) is assumed to be normally distributed with a zero mean and sampled covariance.

$$E[w^2] = \sigma^2 f_s \quad (9)$$

The moving bias term (b_r), or sensor drift, is modeled as a first order Markov process and is outlined in Equations (10), (11), and (12). The Markov process takes into account the sampling frequency (f_s) of the sensor and has the following statistics

$$E[b] = 0 \text{ and } E[b^2] = \sigma_{bias}^2 \quad (10)$$

$$\dot{b} = -\frac{1}{\tau} b + w_{bias} \quad (11)$$

where

$$w_{bias} = \sqrt{\frac{2f_s\sigma_{bias}^2}{\tau}} v. \quad (12)$$

The noise, v , that drives the bias, is normally distributed with zero mean and a sampled covariance of one as shown in Equation (13).

$$v \sim N[0,1] \quad (13)$$

This results in a variance for the bias, which is given in below.

$$Q_{bias} = E[w_{bias}^2] = \frac{2f_s\sigma_{bias}^2}{\tau} \quad (14)$$

During initial algorithm development and concept verification, the sensor characteristics from the data sheet are used to model the sensors. However, for a more accurate simulation, the specific sensors used in the HIL implementation are characterized, and their statistics are used in simulation. For the purpose of this work, Allan Variances were conducted on static sensor data to determine the true characteristics of the IMU used in the navigation unit and other IMU's that may be implemented, shown in Figures 3 and 4. Using the experimentally determined IMU characteristics ensures that the simulated sensors closely resemble the actual sensors in performance.

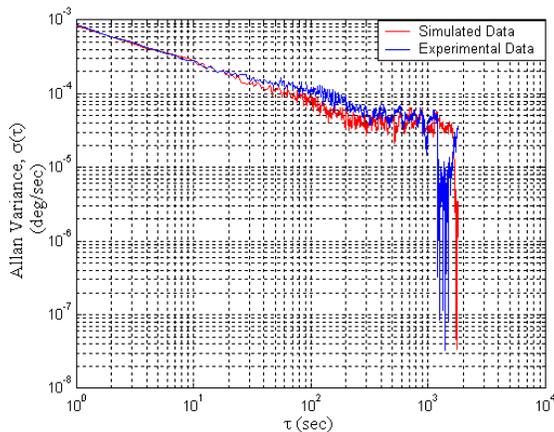


Figure 3: Allan Variance Comparison of Actual Data and Experimental Data for KVH Gyro

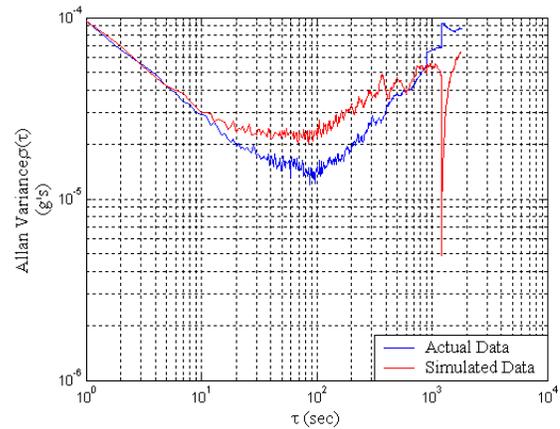


Figure 4: Allan Variance Comparison of Actual Data and Simulated Data for Microelectromechanical systems (MEMS) IMU

The close match of the experimental data to the simulated data indicates that the simple sensor models employed are sufficient to characterize the sensor wide-band noise and walking bias. These sensor models, along with the “perfect” inertial data from ANVEL, are used to generate realistic inertial measurements to be used by the navigation algorithm.

SIL WSS

Given encoder resolution, vehicle position, and wheel speed information, encoder counts are also produced for the SIL implementation. To characterize the WSS, many test runs were conducted using WSS on the GAVLAB's test vehicle around the National Center for Asphalt Technology (NCAT) test track. The NCAT track is a 1.7 mile oval track with known bank angle, and serves as a benign test environment. The GAVLAB test vehicle is a 2003 Infiniti G35 outfitted with Septentrio 3 antenna RTK GPS, the previously described navigation unit, WSS, and a Honeywell eTALIN IMU used for truthing.

Quantization error suffered by wheel encoders is well documented [7,8]; however, during testing it was observed that the WSS error also grew cubically with increasing speed.

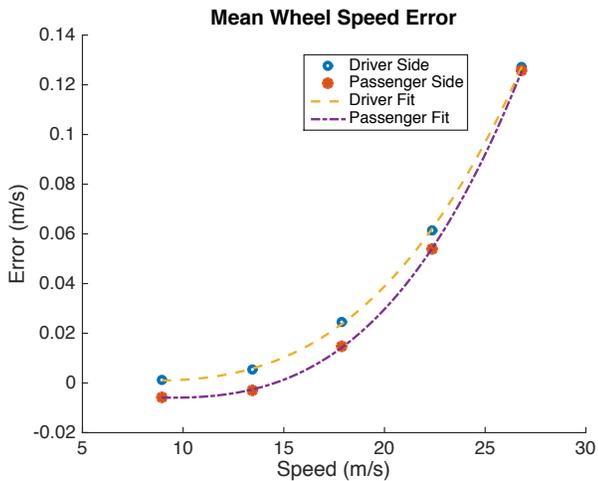


Figure 5. Wheel Speed Error vs. Speed

The wheel speed error shown is calculated as the difference between the reported wheel speed and Septentrio speed in the straight portions of the NCAT test track. The errors are due to changes in the effective wheel radius and the fact the tires must produce more force as the vehicle travels faster to overcome air drag. Wheel speeds are calculated as rotation rate multiplied by the effective wheel radius as seen in Equation (15).

$$\frac{\Delta_{count}}{\Delta_{time}} * \frac{1 \text{ revolution}}{1000 \text{ counts}} * 2\pi r = \text{wheel speed} \quad (15)$$

The effective wheel radius is found by comparing wheel rotation rate to Septentrio speed in the straight-line sections of the NCAT track at low, constant speed in order to minimize the amount of wheel slip in the data. Wheel slip is accounted for in ANVEL by use of a high fidelity tire model. ANVEL has several models built in, and a custom model may be added, but for this work, the built-in Pacejka tire model is chosen. Using the results found from wheel speed characterization, the “perfect” ANVEL wheel speed information is corrupted and then input into the navigation unit.

HIL

Implementing the IMU and WSS modules for the HIL system requires the previously described SIL modules. The key difference is the SIL modules are adapted to output signals representative of the sensors via hardware interface instead of measurements. For the IMU module, a serial interface is developed to mimic output from the IMU within the navigation unit. With the addition of the developed serial interface to the IMU error model, the simulation becomes as close to live testing as possible for the IMU without the use of a rate table. A rate table was not used in this work because of cost concerns; therefore the HIL implementation presented in this work is not true HIL. The usage of the HIL IMU module requires the navigation unit have the ability to switch to an external IMU.

To implement the HIL WSS module, the SIL WSS module is switched to output digital pulses instead of encoder counts. The pulses are sent through a DAC so that an analog pulse signal is generated and sent to the navigation unit, which reads the pulses and converts them to wheel speed as when implemented on the test vehicle.

SPECTRACOM INTEGRATION

In order to create a higher fidelity GPS simulation, GPS RF signals may be introduced to the system through the use of a Spectracom GPS simulator with real-time RF signal generation capabilities as part of HIL mode. GPS satellite trajectories are calculated by the Spectracom using RINEX data downloaded from the CDDIS GNSS archive [1]. Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONASS) almanac data is also downloaded from [9] for use by the Spectracom. The Spectracom is provided the true vehicle position from ANVEL and uses the calculated satellite trajectories to generate RF signals to be used as input for the Novatel OEM 628 receiver in the navigation unit. Ionospheric and Tropospheric errors are simulated by the Spectracom unit as well. The real-time capabilities

of the Spectracom are realized using Real-time Scenario Generation (RSG) commands as specified in the Spectracom GNSS Signal Generator (GSG) Standard Commands for Programmable Instruments (SCPI) handbook [10]. The Spectracom simulator also employs ray tracing and only generates RF signals for each of the satellites in view. Environment models for the purpose of ray tracing by the Spectracom can be easily generated using the SketchUp tool. SketchUp is a 3D modeling software available for free from Trimble Navigation [11]. The use of the Spectracom introduces GPS errors inherent to RF signals such as electrical noise and signal processing of the receiver, which cannot be easily simulated. The addition of the Spectracom module allows researchers to choose between simulated GPS output and output from an actual receiver using realistic RF signals. This choice gives researchers the ability to easily analyze and compare errors resulting from the two different GPS modules.

MODULARITY AND REAL ERRORS

The major benefit of the system architecture presented in this work is the modularity. For fast prototyping, researchers may use the system in SIL mode, which does not require any of the hardware specified for HIL mode. This means the researchers do not use the Spectracom simulator, navigation unit, DAC, or interfaces between modules and navigation unit. However, using the HIL mode allows researchers to add more realistic errors and develop algorithms in environments closer to live implementation. Using the two different modes, researchers can “turn on and off” individual error sources, giving the ability to analyze the effects of each individual error source on the developed navigation algorithm and/or controller. The environment is developed with an open and modular architecture to ensure easy additions of other sensor packages. Both navigation and control algorithms may be changed with ease, and other sensors such as a camera or

Light Detection and Ranging (LIDAR) may be added through other plug-ins. As stated, the system can also be configured to place the navigation unit on a rate table. The simulated IMU measurements would be turned off and the IMU would measure the motion of the box, introducing real IMU errors to the simulation

CONCLUSION

The AGR STO and other future robotic efforts will need high fidelity HIL/SIL modeling to be able to develop and improve the automation of ground supply distribution systems. The physics-based simulation tools developed and refined under this effort will have broad applications across future efforts allowing for reduced development and testing time, cost savings and risk reduction.

REFERENCES

1. International GNSS Service (IGS). 1992. Daily GPS Broadcast Ephemeris Files. Data. NASA CDDIS. Available from [<ftp://cddis.gsfc.nasa.gov/gnss/data/daily/>](ftp://cddis.gsfc.nasa.gov/gnss/data/daily/)
2. "Importing Custom Environments." *ANVEL Wiki*. Quantum Signal, LLC, 5 June 2017. Web. 10 June 2017.
3. Misra, Pratap, and Per Enge. "Signal Propagation Modeling Errors." *Global Positioning System: Signals, Measurements, and Performance*. Lincoln, MA: Ganga-Jamuna, 2011. 157-74. Print.
4. International GNSS Service (IGS). 1992. Daily TEC Maps. Product. NASA CDDIS. Available from [<ftp://cddis.gsfc.nasa.gov/gnss/products/ionex/>](ftp://cddis.gsfc.nasa.gov/gnss/products/ionex/)
5. International GNSS Service (IGS). 1992. Daily ZPD Estimates. Product. NASA CDDIS. Available from [<ftp://cddis.gsfc.nasa.gov/gnss/products/troposphere/zpd/>](ftp://cddis.gsfc.nasa.gov/gnss/products/troposphere/zpd/)
6. Flenniken, Warren. *Modeling Inertial Measurement Units and Analyzing The Effect*

- of Their Error In Navigation Applications*. Thesis. Auburn University, 2005. Print.
7. T.A.C. Verschuren. *Extracting More Accurate Position and Velocity Estimations Using Time Stamping*. 2006.
 8. Gubner, John A. "Expectation of a Single Random Variable." *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge: U, 2010. 150-51. Print.
 9. International GNSS Service (IGS). 1992. Daily GLONASS Broadcast Ephemeris Files. Data. NASA CDDIS. Available from <ftp://cddis.gsfc.nasa.gov/glonass/data/daily/>
 10. *GSG-5/6 Series SCPI Handbook*. Rochester, NY: Spectracom, 2014. Electronic. Revision 15.
 11. "3D Modeling for Everyone." *3D Modeling for Everyone | SketchUp*. Trimble, Inc., 2017. Web. 16 May 2017.

Disclaimer: Reference herein to any specific commercial company, product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the Department of the Army (DoA). The opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or the DoA, and shall not be used for advertising or product endorsement purposes.

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. OPSEC # 29254