ISSN 1 746-7233, England, UK
World Journal of Modelling and Simulation
Vol. 8 (2012) No. 3, pp. 172-184

High fidelity modelling and simulation of inertial sensors commonly used by autonomous mobile robots *

Phillip Jefferson Durst† , Christopher Goodin

US Army Engineer Research and Development Center, 3909 Halls Ferry Road, Vicksburg 39180, America

(*Received March 3 2011, Accepted April 26 2012*)

Abstract. Autonomous mobile robot behaviors are notoriously difficult to predict using simulations. Robot behaviors are driven by the robot's sensor systems, and sensor outputs need to be produced in simulations at near-truth levels. However, in order to achieve real-time or faster performance, most simulation environments use empircal, effects-based sensor models, and these simplified models do not provide sensor outputs at the level of fidelity necessary to develop and test autonomy algorithms. Simulated sensor outputs must accurately reflect not only the sensor's intrinsic noise and signal processing effects but also the sensor's interaction with its external environment. To address this need for physics-based sensor models, models of the Global Positioning System (GPS), micro electro-mechanical (MEMS) accelerometer, and MEMS gyroscope sensors were developed using a first-principles, physics-based methodology. These sensor models run in near real-time on high performance computering (HPC) assets. The sensors were simulated for a test site located in Vicksburg, MS, and the simulated sensor outputs were compared to truth data collected using a Garmin hand held GPS unit and low cost 6 DOF commercial inertial measurement unit (IMU) consisting of three orthogonal MEMS accelerometers and 3 orthogonal MEMS gyroscopes. The sensor models were also used as part of a simulated autonomous unmanned ground vehicle reconnaissance mission, and results from this simulation are presented.

Keywords: physics-based modelling and simulation, sensor modelling, autonomous navigation, high performance computing (HPC), Global Positioning System (GPS), inertial measurement unit (IMU)

1 Introduction

Due to its complex nature, the problem of autonomous navigation for unmanned ground vehicles (UGVs) remains largely unsolved. For a UGV to maneuver without supervision, it must be able to perceive and localize itself within its environment. UGV perception and positioning are usually accomplished through a combination of electro-optical (EO) sensors (LADAR, cameras, etc.) and inertial sensors (GPS, accelerometers, etc.). However, because of the multitude of interactions that can occur between these sensors and the surrounding environment, perception and localization prove to be difficult tasks. Autonomous navigation systems (ANS) must be prepared to overcome such issues as ground occlusion by foliage, reflective surfaces, GPS dropout, wheel slippage in soft soil, and countless other obstacles and problems. For a simulation environment to be a viable developmental tool for mobile robot autonomy algorithms, it must accurately recreate these sensor-environment and vehicle-terrain interactions.

A simulation test bed that could recreate the impact of the environment on UGV perception and localization in a deterministic, physics-based way would lead to advancements in autonomous navigation. Recognizing this need, the US Army Engineering Research and Development Center (ERDC) has undertaken the development of a high fidelity simulation tool for evaluating ANS, called the Vehicle Autonomous Navigation Environment Computational Test Bed (VANE CTB)[12]. The VANE CTB incorporates modelling and

* Permission to publish granted by Director, Geotechnical and Structures Laboratory.

† Corresponding author. Tel.: 601-634-2410, Fax: 601-634-3068. *E-mail address*: phillip.j.durst@usace.army.mil.

simulation of the environment, energy flow through the environment (via ray casting), vehicle physics, vehicle-terrain interaction, and numerous sensors involved in perception and localization[9]. The focus of this paper is the inertial sensor models contained within the VANE CTB.

A critical aspect of any ANS is localization. While exteroceptive sensors, such as Ladar, Sonar, or millimeter wave Radar, are used to discover the environment's geometry, the ANS must know the position and orientation of these sensors to accurately create an internal model of the world[3, 15]. In order to plan a path and move through the environment, the ANS must also know its position within the world[10]. Therefore, high fidelity models for several common inertial sensors have been developed for use in the VANE CTB. These models include a GPS and micro electro-mechanical system (MEMS) accelerometer and gyroscopes. The sensor models take into account both external environmental effects and internal sensor processes. Section 2 describes the models in detail.

Section 3 presents some of the validation work done for the developed sensor models. Stationary GPS and IMU data were collected and compared to simulated data. Additionally, the GPS sensor model was simulated on a 60-foot diameter circular track running eight loops at a constant velocity and along a 60 foot straight test track running back-and-forth for ten passes at a constant velocity. A Garmin 18x-5Hz[7] was used for the test tracks, and for the stationary data collections a Garmin GPSmap 76CS[7] was used. IMU data was collected using a PhidgetSpatial 3/3/3 9 degree of freedom inertial measurement unit (IMU)[22]. A detailed comparison of the simulated to actual GPS and IMU data is presented.

Section 4 presents some results from a VANE CTB simulation. During the simulation, an autonomous UGV was tasked with the reconnaissance of a building of interest. During its mission, the UGV used a combination of GPS and IMU data for localization and LIDAR point cloud registration. The UGV mission was simulated twice, once with the presented GPS and IMU models and once with simplified, low fidelity models. Differences between the two simulation runs are presented which highlight the need for high fidelity sensor models.

2 Sensor models

2.1 GPS

Without a doubt, the most popular sensor used by autonomous mobile robots for localization is the GPS. GPS errors arise from many sources, such as signal delays caused by the conductive and neutral atmosphere. Because GPS is so complex and effected by so many outside sources, most simulations simply simulate the receiver's position by corrupting the robot's true position with white noise[8, 14, 24]. This type of model is not desirable, however, because it does not account for the two error sources that impact autonomous navigation the most: dilution of precision (DOP) and multi-path effects. DOP errors arise when satellites move in and out view of the receiver or when the satellites visible are closely grouped together, and DOP is the cause of the commonly observed receiver 'jumping' problem. The most accurate GPS model found within the literature to date is that of Balaguer and Carpin [2], which does use the number of satellites visible to the receiver to determine the the amount of error applied to the returned position. This is a better, but not ideal, solution. In order to reduce the computational burden of the model, it does not directly calculate receiver position using receiver-satellite ranges and therefore does not recreate DOP or multipath effects.

The GPS model developed in this work uses a geo-specific, psuedorange method that calculates the receiver position using the distances between the receiver and the GPS space vehicles (SVs). The model operates in three steps. First, it finds the true ranges, R, between the receiver and the SVs via bi-directional ray tracing. The initial SV positions are specific to the site at which the receiver is being simulated. At the start of the simulation, the SV positions within the scene are determined using ephemeris data taken taken from the Scripps Orbit and Permanent Array Center (SOPAC) archive[25]. At each time step during the simulation, the SV positions are updated using Kepler's equations[16] and the ephemeris (orbit parameter) data contained in the SV broadcasted RINEX file[23]. Once the SV positions have been determined, ray tracing is used to check whether or not the SV is visible to the receiver. Ray tracing is also used, along with material reflectance attribution within the scene, to check for multipath effects. To determine multipath errors, the ray tracer finds

every available path between the receiver and an SV. The first return multipath range for an SV is used when the time difference between the first and second returns is less than the cut-off time for the modeled receiver (typically 150ns or 45-meter path difference).

After finding the visible SVs and their distances from the receiver, the model takes the true range (or chosen multipath range) and alters it using several error functions. The effects of the Earth's atmosphere are modeled using three different error functions. Ionosphere range errors are modeled using a Montenbruck mapping function[17]

$$dR_{\text{iono}} = \frac{2.037}{\sqrt{\sin^2 \epsilon + 0.076} + \sin \epsilon},$$

where ϵ is the viewing angle between the receiver and the SV and the constants are derived from the standard range error at zenith. Likewise, the effects of the upper Troposphere are modeled using a Neill mapping function[20]

$$dR_{\text{trop,wet}} = \frac{1 + \frac{a}{1+\frac{b}{1+c}}}{\sin \epsilon + \frac{a}{\sin \epsilon + \frac{b}{\sin \epsilon + c}}},$$

where ϵ is the viewing angle between the receiver and the SV and a, b, and c are constants, the detailed derivation of which can be found in [19]. The range errors caused by the lower atmosphere, or 'hydrostatic Troposphere', are modeled using the equations derived by Park and Ha in [13]

$$dR_{\text{trop,dry}} = \frac{2.2779P}{1 - 0.00266 \cos 2\theta - 0.0028h},$$

where P is the air pressure at the receiver, θ is the receiver's latitude, and h is the receiver's height above ellipsoidal.

In addition to atmospheric effects, the model also accounts for range errors due to SV broadcasted ephemeris errors. As the SVs orbit the Earth, errors in their internal calculations can degrade the SV position reported to the GPS receiver. Errors in the reported SV position in turn create errors in the SV-receiver range. These range errors are modeled using an equation derived from the work of Hwang and Parkinson [11, 21].

$$dR_{\text{SV ephemeris}} = \sqrt{(1 + \frac{1}{2}\sin\theta)(10^{\frac{2R_{ref}}{R_0}}) + c^2(10^{-}18) + 0.918},$$

where θ is the SV's latitude, R_{ref} is a reference orbit radius of 26550000 meters, R_0 is the SV orbit radius, and c is the speed of light in a vacuum.

Once all of the error effects have been accounted for, a new distance between the receiver and SV is calculated

$$R' = R + dR_{\text{iono}} + dR_{\text{trop,wet}} + dR_{\text{trop,dry}} + dR_{\text{SV ephemeris}}.$$

The new ranges, R', are then fed into a generic trilateration algorithm, which computes the receiver's position. A detailed discussion on how GPS receivers find position using SV-receiver ranges can be found in [21]. By using this method, the receiver's returned position will suffer from DOP errors, and these errors will effect the returned position in a deterministic way. To test the offset from true receiver position caused by each error function, the GPS model was simulated at several points along a circle. The receiver's position was calculated for each individual error function, and Fig. 1 shows the effects of each error source on the GPS receiver output position. In Fig. 1, output GPS receiver position for each individual error source for a set of points along a circle. Ephemeris data for seven SVs were used for this test, and position was reported in meters in a local East/North/Up coordiante frame. As would be expected for a true GPS receiver, the greatest offsets are due to the Ionosphere and DOP effects and the Troposphere induced errors are minimal.

Tab. 1 compares the offset for each error source to the expected values found in [21].

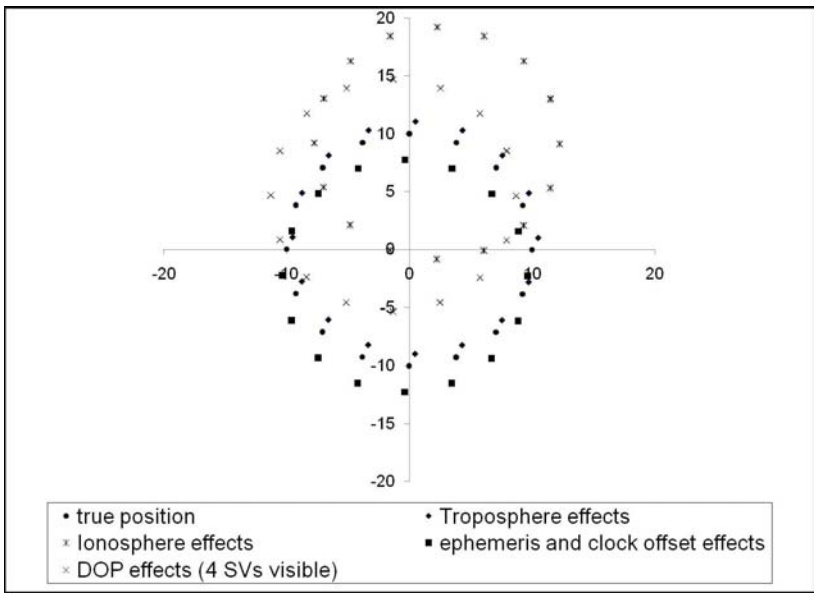

Fig. 1. The effects of each error source on the GPS receiver output position

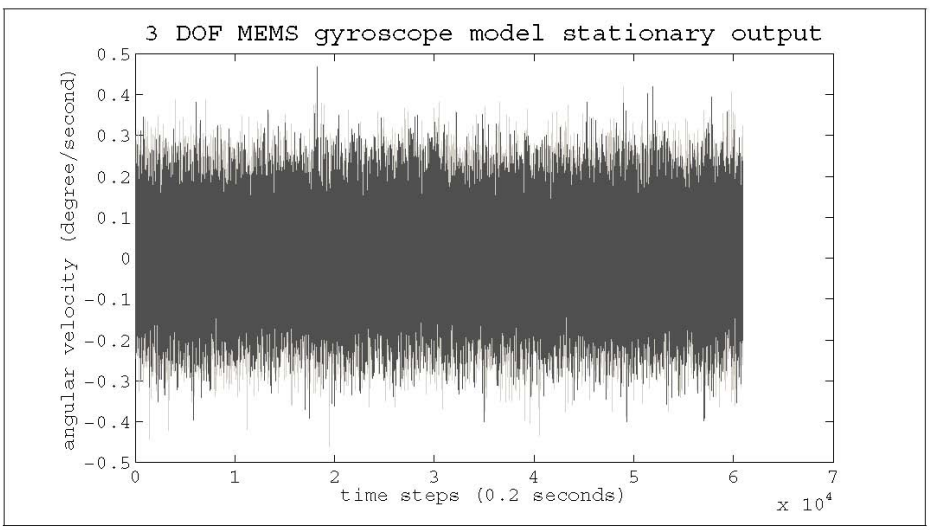

Fig. 2. Simulated MEMS gyroscope sensor outputs for a stationary input

Table 1. Mean receiver offset due to each GPS-SV range error source

Error Source	Expected offset (m)	Model offset (m)
Troposphere	±1.0	0.57
Ionosphere	±5.0	4.73
SV on board errors	±4.5	1.14
Dilution of Precision	varies	2.43

The various error functions can be turned on or off to simulate different modes of GPS operation. An off-the-shelf GPS will suffer from all the above mentioned sources of error. For a dual band GPS, the Ionosphere error is removed. Similarly, for a differential GPS (DGPS), only the Troposphere errors will have an effect on the receiver-SV range. The novelty of the model lies in its use of bi-directional ray tracing within the simulation scene and material reflectance properties. It is the first GPS simulation to explicity and deterministically calculate DOP and multi-path effects, which are the two most important GPS error sources from the viewpoint of an autonomous mobile robot. The model is also the first GPS sensor model to simulate, in a physics-based and fully accurate fashion, the different modes of operation for GPS receivers. The impact these effects have an simulated robots will be discussed further in Section 4.

2.2 Mems accelerometer and gyroscope

The MEMS sensor models are based on a spring and proof mass MEMS design such as [6] for the accelerometer and [5] for the gyroscope. Both the accelerometer and gyroscope models take as input the sensor's true acceleration and angle rates of change, respectively. These true values are then corrupted using both deterministic and white noise error functions. First, the true accelerations/angular velocities are corrupted by thermal noise. The accelerometer thermal noise is calculating using the equation of [6]:

$$\sigma_{aT} = \sqrt{\frac{4k_B T\omega_0}{mQ}},$$

with k_B being Boltzmann's constant, T being the temperature of the sensor, ω_0 the fundamental angular frequency of the spring-mass system, m being the mass of the proof mass, and Q being the sensor's quality factor. Similarly, the gyroscope thermal noise is calculated using the work of [1]:

$$\sigma_{gT} = \frac{Q}{k} * \sqrt{4k_B TRB},$$

with k_B being Boltzmann's constant, T being the temperature of the sensor, Q the sensor's quality factor, k the spring constant, R the equivalent damping resistance, and B the noise bandwidth. Both equations are highly notionalized, and a wide range of sensors can be simulated simply by adjusting the equation parameters k, Q, ω_0, R, and B. In addition to thermal noise, a quantization error term is added to the MEMS sensor outputs using an Allan variance method described in [27]:

$$\sigma_{quant} = 0.01745\sqrt{\frac{3Q^2}{t}},$$

where Q is the sensor's quality factor and t is the total run time of the sensor, in hours. During each time step of the simulation, normal distributions with a means of σ_T and σ_{quant} are generated for each MEMS sensor, and a random value is sampled from these distribution and added to the senor's output.

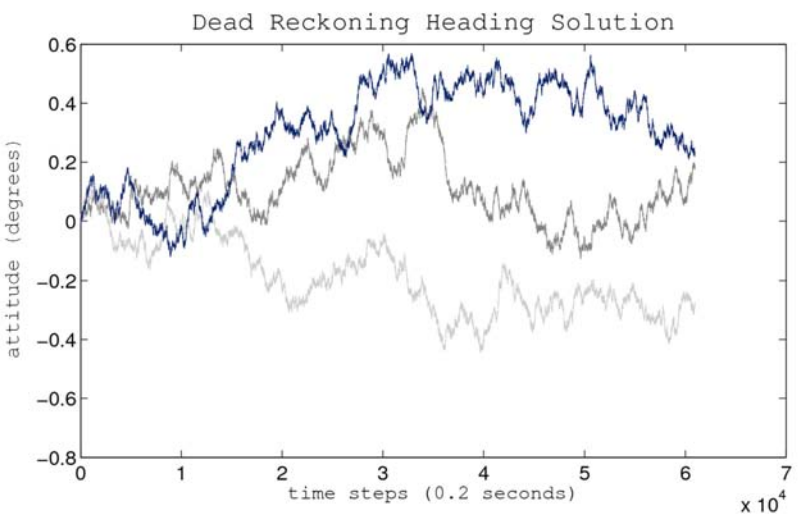

Fig. 3. The dead reckoning attitude solution for the test run

After deterministic noise is added to the accelerations/angular velocities, bias and scale errors are generated. Because the error in the sensor output is dependent on the error in every previous sensor output, bias and scale errors must be modeled using a recursive process[26]. Therefore, a first order Gauss-Markov process was

chosen to generate white noise for the MEMS sensor models. The general model can be found in [18], and is as follow

$$\sigma(t) = \left(\frac{1}{\tau}\right)\sigma(t-1) + \sigma_0,$$

where τ is the correlation time in the noise and σ_0 is the mean offset from zero, and $\sigma(0) = \sigma_0$. Like the error models employed for systematic noise, the nature of the GM process is such that numerous off-the-shelf IMU sensors can be tested by simply varying the correlation time and zero offset constants. As with each other source of noise, at each time step, normal distributions with a means of $\sigma(t-1)$ and σ_0 are created and randomly sampled from in order to calculate $\sigma(t)$. Fig. 2 shows the output for a stationary 3 DOF MEMS gyroscope over a 20 minute test run, and Fig. 3 shows the dead reckoning attitude solution for the test run. Output dead reckoning heading solution for three orthogonal MEMS gyroscopes plotted versus time using the simulated stationary sensor output shown in Fig. 2. Each line corresponds to the angular heading calculated using the sensor's output angular velocity, in degrees, for one gyroscope. The total drift rate for the gyroscpes is approximately 3 degrees/hour, which is in agreement with [4]. The values for noise functions were notional and taken to be similar to those found in [1, 4].

3 Model validation

Data were collected over the course of several days at an outdoor test site located in the Le Tourneau community of Vicksburg, Mississippi in November of 2010. Meteorological conditions collected during testing were fairly constant, with the temperatures varying between 50.4°F and 59.6°F, humidity between 73% and 87%, and surface pressure roughly constant at 1022 hPa. A Garmin GPSmap 76CS was used to collect stationary GPS data for 20 minutes at two different locations. Stationary IMU data were also collected in tadem with the GPS data using a PhidgetSpatial 3/3/3. Using a Garmin 18x-5Hz GPS receiver, data were also collected for a constant velocity traversal of a 30 foot radius circular test track and a 60 foot linear test track.

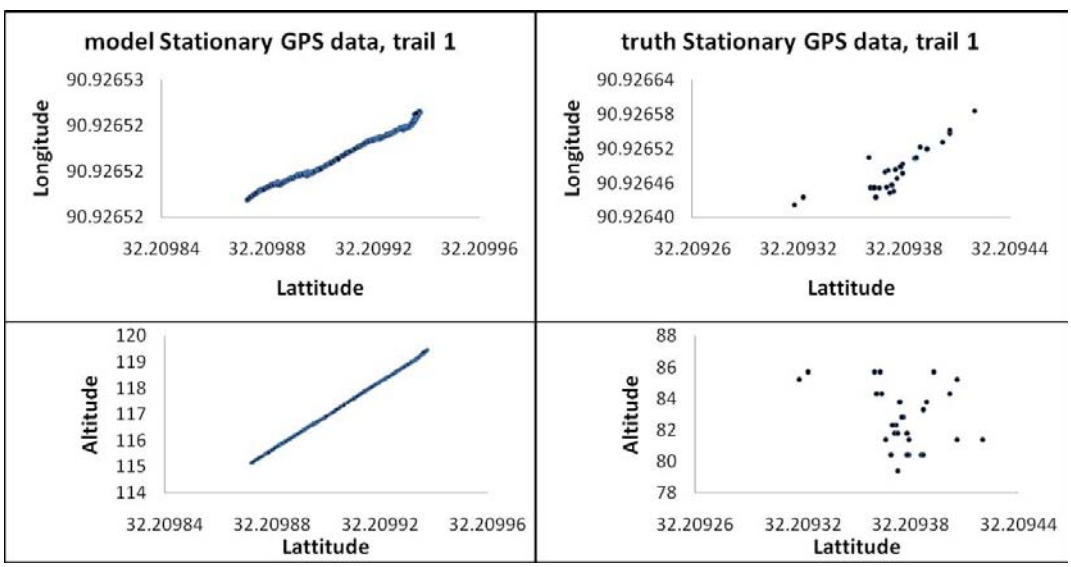

Fig. 4. The simulated and ground truth GPS position for the first stationary data collection

For the GPS model, a single channel commercial receiver was simulated, so all error function were present. Because of the choice of an outdoor test site, multipath effects were not taken into account. Average atmospheric weather conditions were used for the model runs, with the height of the 200 hPa geopontential set at 12,000 meters and the vertical total electron content (VTEC) set at 12. During field testing, an average of 7 SVs were visible. For model runs, SV ephemeris data for 7 SVs were taken from the SOPAC archives for

November 17, 2010 at 1300 hours. For the IMU model settings, a full characterization of the Phidget sensor was not performed simply because it is such a low quality system (drift rate roughly 3°/minute). Rather, generalized inputs representing a low quality system with comparable drift rates were chosen. The quality factor was set at 0.15, the correlation time to 10 seconds, the white noise process to 0.30 degrees/second for the gyroscope and 0.15 m/s^2 for the accelerometer, and the zero offsets were taken from the field collected data.

Table 2. Simulated and actual GPS receiver position standard deviations for 30 minute stationary data collections

	Truth		Simulation	
	STDEV(total)	STDEV(lat,long)	STDEV(total)	STDEV(lat,long)
Trail 1	1.943	4.465e-5	1.68	2.40e-5
Trail 2	7.729	2.2389e-5	3.661	1.0746e-4
Circle track	2.67	4.5e-3	0.85	7.83e-5
Linear track	2.65	6.5e-5	0.85	4.02e-5

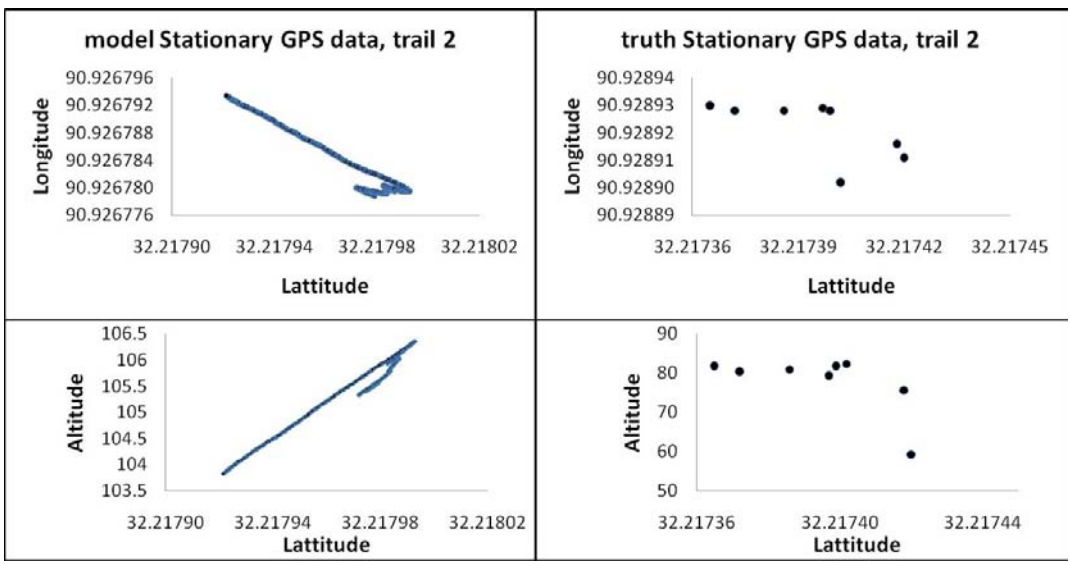

Fig. 5. The simulated and ground truth GPS position for the second stationary data collection

A DGPS survey of the test site was not performed due to time and cost constraints, so the GPS model could not be checked for absolute positioning accuracy. Instead, the standard deviation from mean position and the drift for the model and ground truth GPS receivers were compared. Likewise, the position drift between passes for the circle and linear test tracks were compared. While this type of validation did not allow an exact one-to-one comparison of model and true receiver offsets, it does provide a good method for comparing the relative errors and error accumulation over time, which are more important. Fig. 4 shows the simulated and ground truth GPS position for the first stationary data collection. In Fig. 4, GPS position for the Garmin GPSmap76CS and presented model, in decimal latitude and longitude, for the first 20 minute stationary data collection. The direction and magnitude of the drift in returned position are almost identical between the real and simulated GPS receivers. The higher variability in the altitude of the true receiver is due to multipath effects, which were not considered for validation test runs. Fig. 5 shows the data from the second and GPS position for the Garmin GPSmap76CS and presented model for the second 20 minute stationary data collection. As with the first test, the direction and magnitude of the drift in returned position are in good agreement. As these figures show, the mean receiver position in the model is offset from truth, but the drift in the returned positions are highly comparable. Likewise, the largest source of position error is in altitude for both the sim-

ulated and actual GPS positions. Table 2 lists the simulated and actual receiver position standard deviations, which are in good agreement.

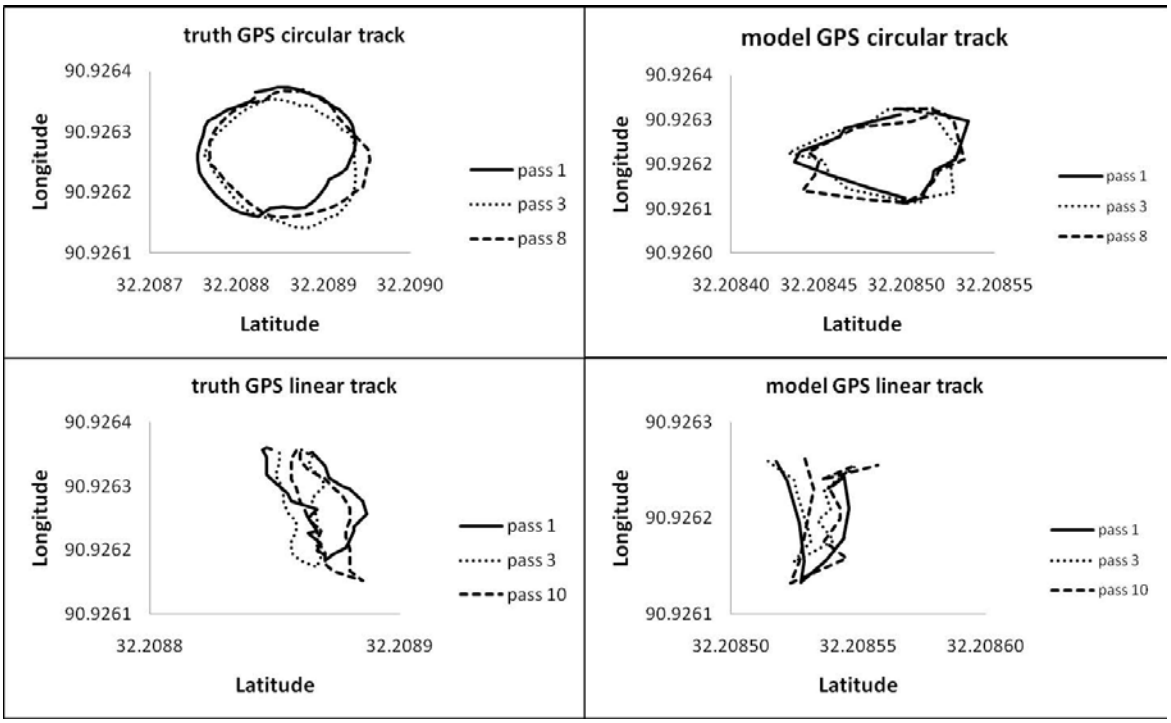

Fig. 6. The truth and model GPS data for the two scenarios

In addition to validation against stationary data, the GPS model was also tested against two test tracks. GPS data were collected for eight passes at constant velocity along a 30 foot radius circle and ten passes along a 60 foot linear test track using a Garmin 18x-5Hz. Fig. 6 shows the truth and model GPS data for the two scenarios. In Fig. 6, GPS data for the Garmin 18x-5Hz and presented model for the circular and linear test tracks. The model is in very good agreement with the true sensor data. In each case, the standard deviation from the mean position and the total overall drift between passes are in very good agreement between the simulated and actual GPS sensors, and these values are listed in Tab. 2. As expected, the greatest source of error in both the simulated and actual GPS positions was in the receiver altitude

Validation testing of the IMU sensor models was done using stationary data only. As with the GPS, the model was validation primarly to test the offset and drift errors and the error accumulated in the dead reckoning solution over time. The Phidget IMU was run at a frequency of 125 Hz during validation testing, and was mounted on a level tripod. Accelerometer and gyroscope data were collected for two hours. Because of the notionalized model settings used and the random nature of IMU noise, the true and simulated outputs are not expected to match. However, the magnitude of the offset from zero and overall drift should be in very close agreement. Because of the low quality of the sensor, the drift in the dead reckoning position and heading was on the order of kilometers after the two hour data collect. Therefore, only the first ten minutes of data are presented and were used for validation testing.

To mimic the poor quality of the tested sensor, the IMU model parameters were set as follows: a quality factor of 0.2, a correlation time of 10 seconds, and a mean white noise process coefficient of 0.12 for the accelerometer and 0.15 for the gyroscope. The null offset was taken to be the mean value measured during field testing for each of the six sensors, and ten minutes of data at 125 Hz were simulated. Fig. 7 shows the dead reckoning attitude and position computed using the true and simulated IMU data. In Fig. 7, computed dead reckoning attitude and positions plotted versus time for the ten minute staitonary IMU data collection. The top two plots show the 3/3/3 Phidget IMU sensor's angular headings computed using the sensor's output angular velocities and the x, y, z position of the sensor computed using the sensor's output accelerations. Similarly, the bottom two plots show the three computed angular headings and x, y, z position for the simulated sensor

with a stationary input. The 3/3/3 Phidget IMU (top) and simulated MEMS accelerometers and gyroscopes (bottom) are in very good agreement. The returned positions and headings match remarkably well, as do the total offset and drift. The agreement between the sensor and the model is very good, especially given that no full characterization of the Phidget was performed.

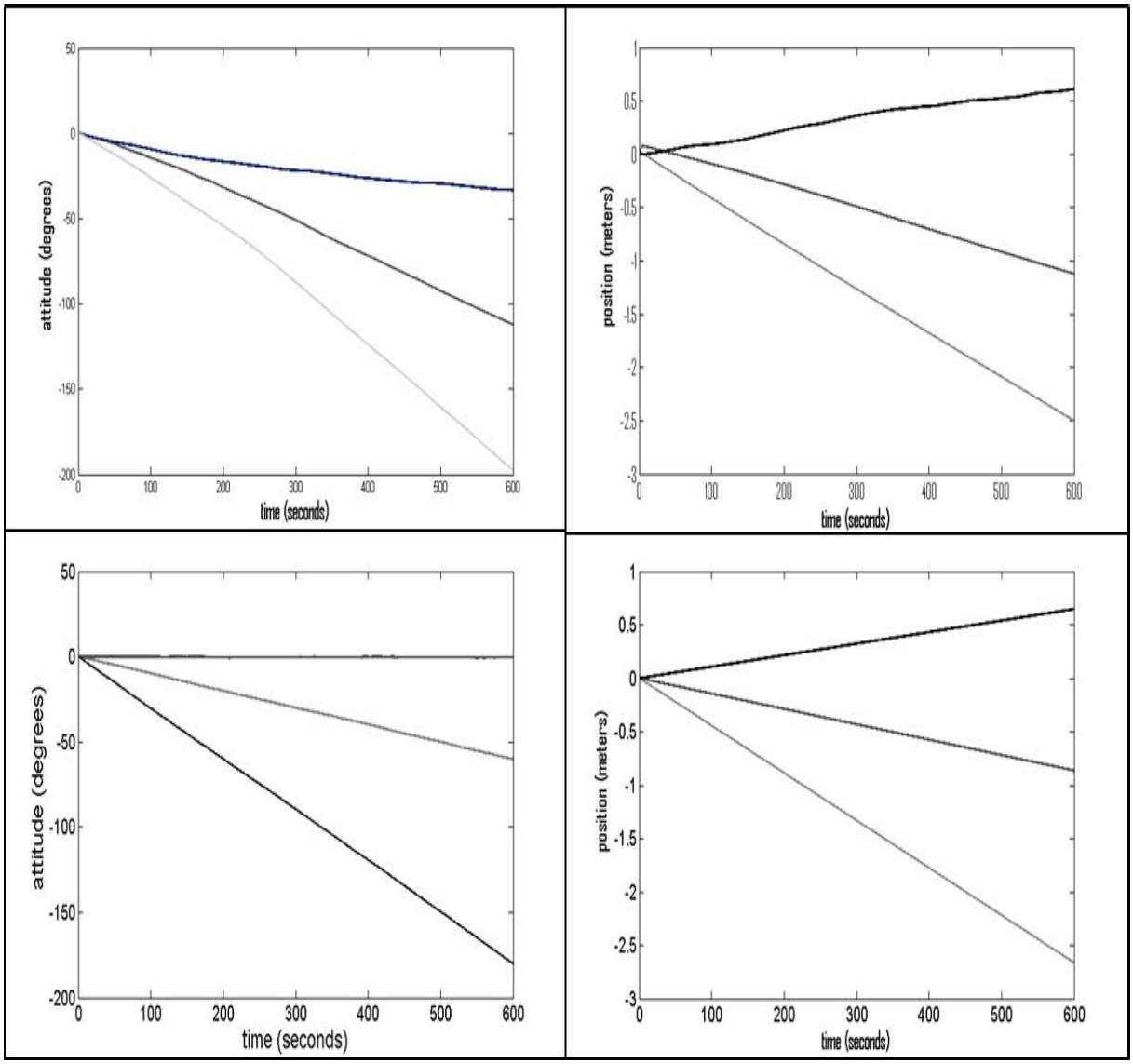

Fig. 7. The dead reckoning attitude and position computed using the true and simulated IMU data

4 Use of the models in an autonomous ugv simulation

The presented models were used as part of a larger VANE CTB simulation. An unmanned ground vehicle (UGV) was simulated performing an autonomous reconnaissance mission in a complex urban environment. Using an a priori map with marked waypoints, the UGV was to navigate from the forward operating base (FOB) to three mission waypoints. When it arrived at these waypoints, the UGV was to stop and collect high resolution images of a building of interest. The autonomy system onboard the UGV made use of LIDAR, GPS, and IMU sensors to complete its mission. Figure 8 is an overhead view of the simulation scene and the UGV mission planning information. Waypoints given to the UGV autonomy system are shown as arrows, with waypoints 7, 9, and 11 being the mission waypoints. The total scene size was one square kilometer.

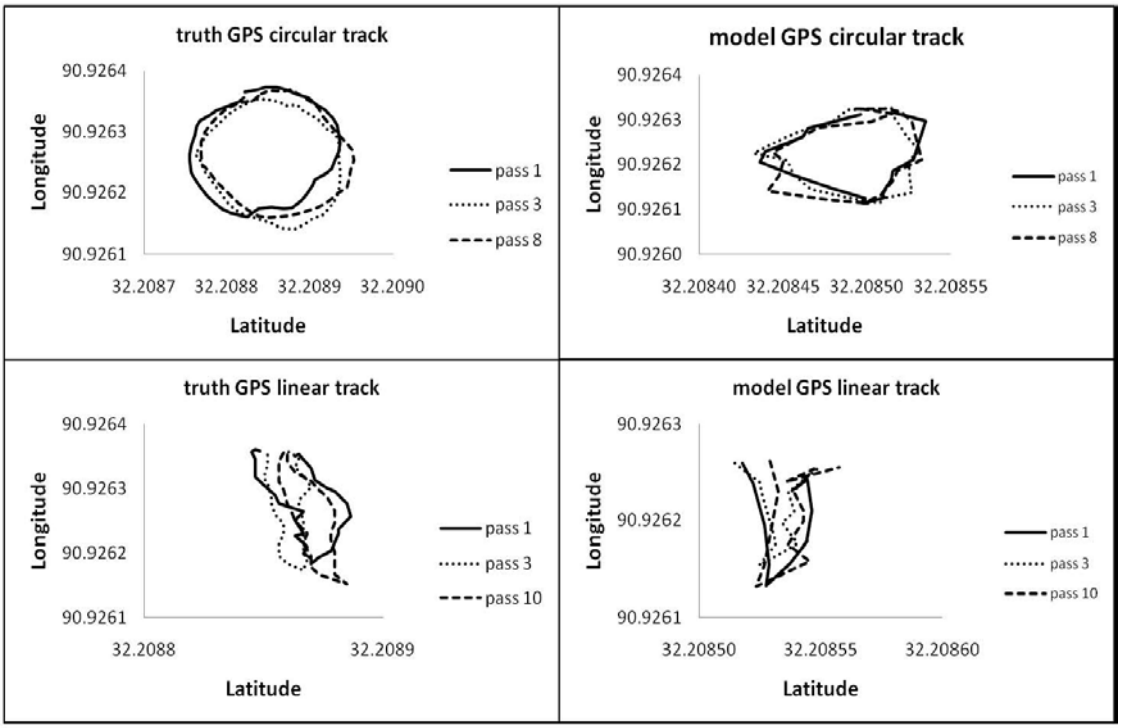

Fig. 8. Scene for the VANE CTB mission simulation

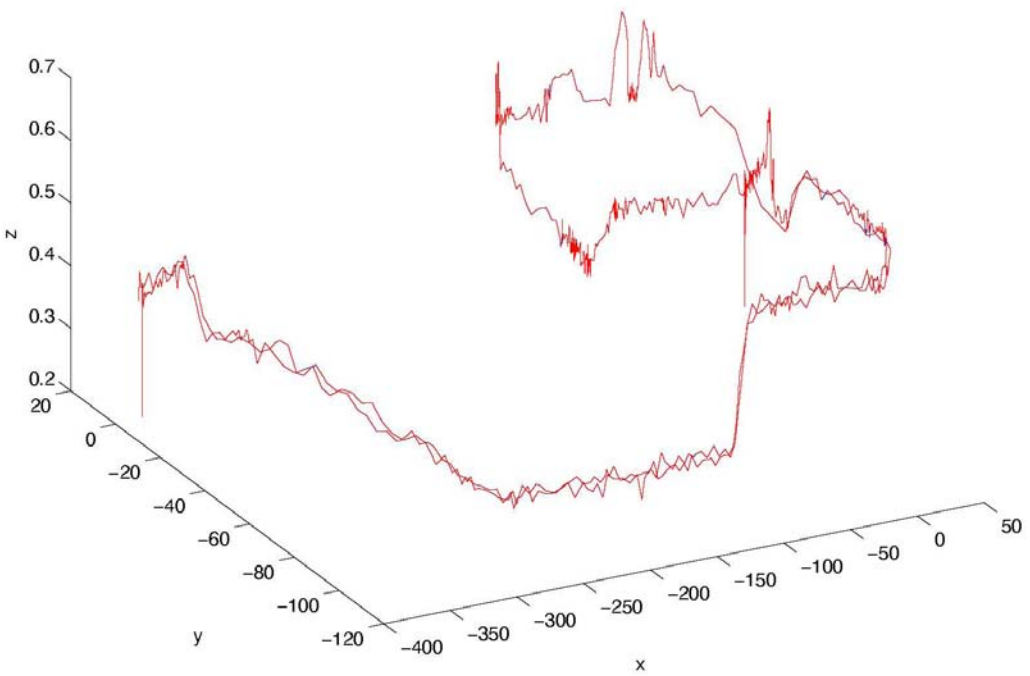

Fig. 9. The output of the UGV position during the simulation

An initial run of the simulation was made using low fidelity localization sensor models. For this run, the GPS was assumed to be a DGPS, and the returned receiver position was simply the true position plus a random error of about 30 to 50 centimeters. The IMU system was assumed to have a "perfect" calibration, with no bias or scale effects and a correlation time of 1500 seconds. Fig. 9 is the output of the UGV position during the simulation. In Fig. 9, true UGV position (red line) and GPS/IMU output position (blue line) generated for the UGV performing the mission shown in Figure 8 using the low fidelity GPS and IMU sensor models. A

DGPS was simulated by offsetting the true GPS position by a random, centimeter scale error at each time step. The true and simulated positions are so nearly identical that the two lines overlap and are indistinguishable. During this initial test, the UGV was able to complete its mission successfully.

Later simulation runs made use of the presented GPS and IMU models. The GPS model was run as a DGPS, and the IMU was assumed to have a very high degree of accuracy, on par with [4]. The mission was simulated several times, and the UGV was never able to successfully complete its mission. The autonomy system onboard the UGV could not compensate for variation in position due to DOP and multipath effects. Fig. 10 shows the path the UGV traveled during the simulation with low fidelity models and the output high fidelity GPS model positions. True UGV position (red line) and GPS/IMU output position (blue line) generated for the UGV performing the mission shown in Fig. 8 using the presented high fidelity GPS and IMU models. As expected, with multipath and DOP errors taken into account, the returned GPS position was highly variable, particularly in the z (altitude) direction. This variation in altitude caused significant errors in LIDAR point cloud registration, and the UGV could not complete its mission. As Fig. 10 shows, the greatest variation in GPS position was in the z direction, which is true for real GPS receivers as was seen in Section 3. The rapid jumps in position caused problems not only with UGV localization but also with LIDAR point cloud registration.

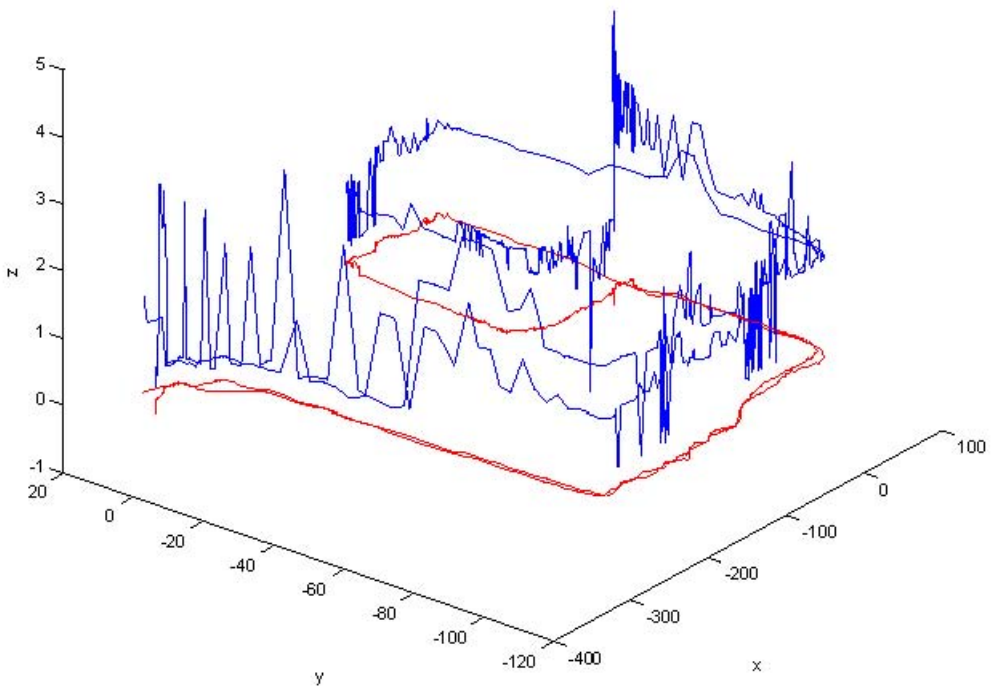

Fig. 10. The path the UGV traveled during the simulation with low fidelity models and the output high fidelity GPS model positions

The high fidelity sensor models were able to better predict the UGV's mission capabilities. Without the developed models, the shortcomings in the autonomy system's localization abilities would not have been discovered until field testing. The results from these simulations clearly show the need for physics-driven sensor modelling for effective aunotomous navigation system development and debugging.

5 Conclusions and future work

In the field of autonomous navigation, simulations have thus far failed to reach their full potential for enabling rapid and efficient development of autonomous mobile robots. For simulations to become a viable tool for developing autonomous navigation algorithms, simulated sensor outputs must be created using physics-based methods. The sensor models must be able to deterministically recreate the many problems that UGVs

face in the "real world". To meet this need, high fidelity GPS and IMU sensor models were developed. The results of simulations using both the high fidelity models and standard, low fidelity were compared, and the need for high fidelity models was demonstrated.

During the VANE CTB simulation, the UGV was able to complete its mission when the low fidelity GPS model was used. However, simulations using the high fidelity models presented ended in mission failure. The UGV's autonomy system was not prepared to deal with GPS multipath and DOP errors or IMU drift. Without the high fidelity models, in particular the GPS model, the UGV's autonomy system could not have been accurately debugged. The problems seen during the VANE CTB simulation would not have been discovered until field testing.

The work presented in this paper represents an initial investigation into need for high-fidelity modelling and simulation for autonomous navigation algorithm development and testing. Work is underway to test the GPS and IMU models on an algorithm level by observing the impact of high fidelity sensor models on the output of specific algorithms. Work is also underway to expand the inertial sensors to include other types of MEMS, including capacitive and piezo-electric.

References

[1] V. Annovazzi-Lodi, S. Merlo. Mechanical-thermal noise in micromachined gyros. *Microelectronics Journal*, 1999, **30**(12): 1227–1230.

[2] B. Balaguer, S. Carpin. Where am I? A simulated gps sensor for outdoor robotic applications. *Simulation, Modeling, and Programming for Autonomous Robots*, 2008, 222–233.

[3] J. Borenstein, H. Everett, L. Feng. *Navigating mobile robots: systems and techniques*. AK Peters, Ltd. Natick, MA, USA, 1996.

[4] A. Brown. Test results of a gps/inertial navigation system using a low cost mems imu. **in:** *Proceedings of the 11th Annual Saint Petersburg International Conference on Integrated Navigation System, Saint Petersburg, Russia*, 2004.

[5] A. Burg, A. Meruani, et al. MEMS Gyroscopes and Their Applications.

[6] M. Gad-el Hak. *The MEMS handbook*. CRC Pr I Llc, 2002.

[7] Garmin, 2010. http://www.garmin.com/garmin/cms/site/us.

[8] B. Gerkey, R. Vaughan, A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. **in:** *Proceedings of the 11th international conference on advanced robotics*, Portugal, 2003, 317–323.

[9] C. Goodin, R. Kala, et al. Sensor modeling for the virtual autonomous navigation environment. IEEE Sensors, 2009.

[10] C. Hofner, G. Schmidt. Path planning and guidance techniques for an autonomous mobile cleaning robot. *Robotics and Autonomous Systems*, 1995, **14**(2-3): 199–212.

[11] C. Hwang, L. Hwang. Satellite orbit error due to geopotential model error using perturbation theory: applications to ROCSAT-2 and COSMIC missions. *Computers and Geosciences*, **28**: 357–367.

[12] R. Jones, J. Priddy, et al. Virtual autonomous navigation environment (vane). ASCE, 2008.

[13] H. Kim, J. Ha, et al. Comparison of Tropospheric Signal Delay Models for GNSS Error Simulation. *Journal of Astronomy and Space Sciences*, **26**: 21–220.

[14] J. Kim, S. Sukkarieh. Slam aided gps/ins navigation in GPS gps denied and unknown environments. **in:** *Proceedings of the 2004 International Symposium on GNSS/GPS*, 2004, 1–5.

[15] J. Leonard, H. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. **in:** *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, IEEE, 2002, 1442–1447.

[16] M. Mehrtash. GPS Navigation Toolbox GNT08.1.2.

[17] O. Montenbruck, M. Garcia-Fernandez. Ionosphere Path Delay Models for Spaceborne GPS. *Deutsches Zentrum fr Luft-und Raumfahrt*, 2005, DLR-GSOC TN 05-07.

[18] S. Nassar. Improving the inertial navigation system (INS) error model for INS and INS/DGPS applications. *UCGE Reports*, 2003, 20183.

[19] A. Niell. The IMF Mapping Functions. **in:** *GPSMet Workshop*, 2003.

[20] A. E. Niell. Improved atomspheric mapping functions for VLBI and GPS. *Earth Planets Space*, **52**: 699–702.

[21] B. Parkinson, J. Spilker. *The global positioning system: theory and applications*. 1996.

[22] Phidget 9DOF IMU, 2010. http://www.phidgets.com/products.php?category=5&product_id=10%56.

[23] Receiver Independent Exchange Format, ver. 2.10, 2009. http://gps.wva.net/html.common/rinex.html.

[24] J. Sasiadek, Q. Wang. Sensor fusion based on fuzzy kalman filtering for autonomous robot vehicle. **in:** *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 4, IEEE, 1999, 2970–2975.

[25] Scripps Orbit and Permanent Array Center, 2010. http://sopac.ucsd.edu/cgi-bin/dbShowArraySitesMap.cgi.

[26] K. Walchko, P. Mason. Inertial Navigation. **in:** *Proceedings of Florida Conference on Recent Advances in Robotics*, 2002.

[27] X. Zhang, Y. Li, P. Mumford, C. Rizos. Allan Variance Analysis on Error Characters of MEMS Inertial Sensors for an FPGA-based GPS/INS System. *School of Surveying and Spatial Information Systems University of New South Wales, Australia*.