NASA/TM–2015-218803

A Short Tutorial on Inertial Navigation System and Global Positioning System Integration

Kyle M. Smalling
Northrop Grumman, Hampton, Virginia

Kenneth W. Eure
Langley Research Center, Hampton, Virginia

September 2015

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM–2015-218803

A Short Tutorial on Inertial Navigation System and Global Positioning System Integration

Kyle M. Smalling
Northrop Grumman, Hampton, Virginia

Kenneth W. Eure
Langley Research Center, Hampton, Virginia

September 2015

A Short Tutorial on Inertial Navigation System and Global Positioning System Integration

Kyle Smalling, Northrop Grumman

Kenneth W. Eure, NASA Langley Research Center

Abstract

The purpose of this document is to describe a simple method of integrating Inertial Navigation System (INS) information with Global Positioning System (GPS) information for an improved estimate of vehicle attitude and position. A simple two dimensional (2D) case is considered. The attitude estimates are derived from sensor data and used in the estimation of vehicle position and velocity through dead reckoning within the INS. The INS estimates are updated with GPS estimates using a Kalman filter. This tutorial is intended for the novice user with a focus on bringing the reader from raw sensor measurements to an integrated position and attitude estimate. An application is given using a remotely controlled ground vehicle operating in assumed 2D environment. The theory is developed first followed by an illustrative example.

1. Introduction

With advancements in Micro-Electro-Mechanical Systems (MEMS) the estimate of vehicle position and attitude has become affordable and increasingly accurate as sensors have become smaller and cheaper. The integration of sensor outputs to achieve reasonable estimates remains an ongoing area of research. Integrating INS with GPS is most commonly done with tight coupling, which integrates sensor data at the measurement level, or loose coupling, which uses outputs of INS and GPS without knowledge of internal variables. The approach presented here is loose coupling. There is an abundance of literature pertaining to strapdown inertial navigation systems and integration with GPS [1]. However, much of this literature is difficult to follow for the uninitiated. In order to greatly simplify complexity, the work presented here will be in two dimensions. Additional simplifications include a flat, nonrotating earth and short vehicle trajectories. These assumptions allow one to assume constant gravity, elevation, and magnetic field, all on a two-dimensional flat earth.

The remainder of this paper is contained in the following 4 sections. Section 2 covers the mathematical development for attitude estimation around a single axis and INS-GPS fusion for derivation of a platform's 3 degrees-of-freedom (3DOF): $(\mathbf{x}_{north}, \mathbf{x}_{east}, \psi)$. Next, Section 3 describes the experimental setup used to collect data for algorithm evaluation. Lastly, Sections 4 and 5 respectively present results and discuss conclusions.

2. Mathematical development

Reference frames play a large role in navigation. Before vehicle position and attitude can be estimated, one must know what the estimates are relative to. Here, we use two reference frames: the

vehicle body frame (*x,y,z*), and the navigation frame which is fixed to the earth (North, East, Down), (NED). In the body frame, we have the origin as the rotational center of the vehicle, *x*-axis out the front, and *y*-axis out the right side. The frame of the vehicle is assumed to always be coplanar with the north and east vectors; the z axis is normal to this plan pointing down. Since the distances traveled by the vehicle are very small in relation to the earth, we will assume a flat earth with perpendicular lines of latitude and longitude. Also, we will assume there are no variations in the local gravitational and magnetic fields. These assumptions together with loose coupling of the GPS-INS greatly simplify the mathematical development. Also, the development presented here is for a vehicle that is assumed to be traveling in 2D. This is a reasonable assumption since the experimental vehicle trajectory was over a localized flat surface.

Estimation of vehicle position and attitude will be done in the navigation frame. It is assumed here that the senor frame and body frame are co-located and aligned. In order to be used for state estimation, the acceleration measurements onboard the vehicle must be converted from the measured body frame to the navigation frame. This may be accomplished by the simple matrix operations shown below in *Algorithm 1* [2].
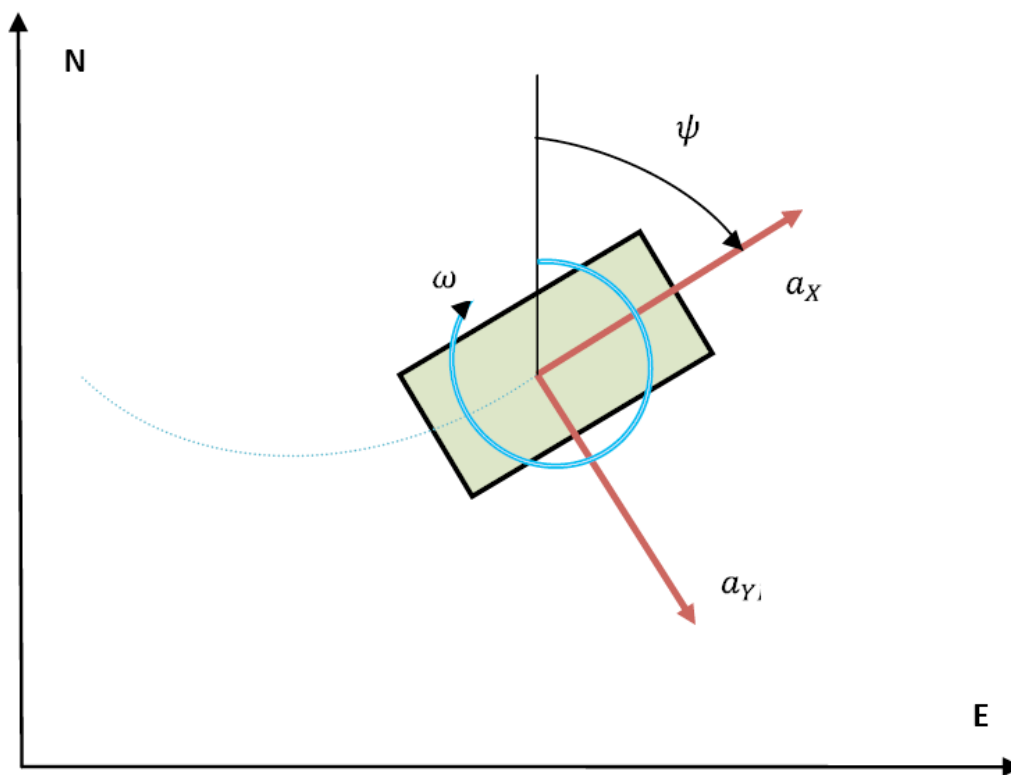
Algorithm 1

Figure 1 Diagram of two-dimensional vehicle body frame and navigation frame. Body frame is front of vehicle and right side (x, y) and navigation frame is North and East (N, E).

$$|m| = \sqrt{m_x(k)^2 + m_y(k)^2} \qquad |^1$$

$$m_x(k) = \frac{m_x(k)}{|m|} \qquad m_y(k) = \frac{m_y(k)}{|m|}$$

$$\psi(k) = \text{atan2}\left(\frac{-m_y(k)}{m_x(k)}\right)$$

$$\mathbf{C_b^n}(k) = \begin{bmatrix} \cos\psi(k) & -\sin\psi(k) \\ \sin\psi(k) & \cos\psi(k) \end{bmatrix}$$

$$\mathbf{f^b}(k) = \begin{bmatrix} f_x^b(k) \\ f_y^b(k) \end{bmatrix}$$

$$\mathbf{f^n}(k) = \begin{bmatrix} f_{north}^n(k) \\ f_{east}^n(k) \end{bmatrix} = \mathbf{C_b^n}(k) \cdot \mathbf{f^b}(k)$$

Figure 1 shows the vehicle body frame *x* and *y* and the navigation frame of north and east. The angular displacement is the vehicle yaw, ψ, and the angular rate is ω, following the right hand rule with *z* pointing down. In this algorithm the outputs from the two axis magnetometer, $m_x(k)$ and $m_y(k)$, are normalized by division of the magnetic vector magnitude, $|m|$. The yaw angle is found using the MATLAB atan2 function which preserves the correct quadrant. Care should be taken that a negative sign is placed on the magnetic *y* measure to match yaw as defined in the figure. The Directional Cosine Matrix (*DCM*), $\mathbf{C_b^n}$, may then be formed using ψ. The *DCM* may then be used to compute the accelerations in the navigation frame. Throughout this algorithm, *k* denotes the integer time step, indicating that this algorithm is carried out every time step. The computed accelerations in the navigation frame, $\mathbf{f^n}$, may then be used to update the state vector in the Kalman filter, later described in *Algorithm 3*. Also, the yaw angle is combined with the gyro measures in the complementary filter to form an improved estimate of the vehicle attitude in *Algorithm 2*.

There are numerous schemes to estimate vehicle attitude based on sensor data. One common technique is the implementation of a Bayesian estimator such as an Extended Kalman Filter (EKF), an Unscented Kalman Filter (UKF), or a particle filter (PF). Because of its simplicity, stability, and accuracy, a special form of the complementary filter developed by Mahony [3] will be used. This algorithm is computationally efficient and well suited for real time processing using limited computational resources; it is shown below in *Algorithm 2*.

Algorithm 2

$$\boldsymbol{\omega}^b(k) = \begin{bmatrix} 0 & -\omega(k) \\ \omega(k) & 0 \end{bmatrix}$$

$$\boldsymbol{\pi}_a(k) = \frac{1}{2} \cdot \left(\hat{\mathbf{C}}_\mathbf{b}^{\mathbf{n}^T}(k) \cdot \mathbf{C_b^n}(k) - \mathbf{C_b^n}^T(k) \cdot \hat{\mathbf{C}}_\mathbf{b}^\mathbf{n}(k) \right)$$

$$\boldsymbol{\Omega}(k) = \boldsymbol{\omega}^b(k) + K_p \cdot \boldsymbol{\pi}_a(k)$$

---

[1] Normalization of the magnetic measurements is not necessary; however it may improve noise immunity.

$$\hat{\mathbf{C}}_b^b(k, k+1) = \mathbf{I} - \frac{\mathbf{\Omega}(k) \cdot \sin(|\mathbf{\Omega}(k)| \cdot T_s)}{|\mathbf{\Omega}(k)|} + \frac{\mathbf{\Omega}(k)^2 \cdot (1 - \cos(|\mathbf{\Omega}(k)| \cdot T_s))}{|\mathbf{\Omega}(k)|^2}$$

$$\hat{\mathbf{C}}_b^n(k+1) = \hat{\mathbf{C}}_b^n(k) \cdot \hat{\mathbf{C}}_b^b(k, k+1)^T$$

$$\psi(k) = \text{atan2}(\hat{\mathbf{C}}_{b_{2,1}}^n(k+1), \hat{\mathbf{C}}_{b_{1,1}}^n(k+1))$$

$$\mathbf{f^n}(k) = \begin{bmatrix} f_{north}^n(k) \\ f_{east}^n(k) \end{bmatrix} = \hat{\mathbf{C}}_b^n(k+1) \cdot \mathbf{f^b}(k)$$

In *Algorithm 2,* the matrix $\boldsymbol{\omega}^b$ contains the output of the z-axis gyro sensor in the body frame. The matrix $\mathbf{C}_b^n$ is that formed in *Algorithm 1*. The matrix $\hat{\mathbf{C}}_b^n$ is an estimate of the true Directional Cosine Matrix formed by combining $\mathbf{C}_b^n$ of *Algorithm 1* with the gyro reading at time step *k*. T_s is the sample rate. From $\hat{\mathbf{C}}_b^n$ the yaw is computed every time step. The subscripts on $\hat{\mathbf{C}}_{b_{2,1}}^n$ and $\hat{\mathbf{C}}_{b_{1,1}}^n$ represent the second row, first column and the first row, first column values respectively. The equation for $\hat{\mathbf{C}}_b^b(k, k+1)$ forms the time step update. If a simpler form of integration is desired, one can use Euler integration with little loss of accuracy. A more accurate estimation of the acceleration in the navigation frame is obtained by using *Algorithm 2* compared to *Algorithm 1* since both magnetometer and gyro inputs are considered. The gain, K_p, is set to one or greater and determines the relative contributions of the gyro and *DCM* to the attitude solution. In summary the inputs to *Algorithm 2* are the gyro readings and *DCM* from *Algorithm 1*. The outputs are the attitude and an improved estimate of the *DCM*. The matrix $\hat{\mathbf{C}}_b^n$ is used to transform the accelerations, which are measured in the body frame, to the navigation frame. Once in the navigation frame, the accelerations may be used as inputs into *Algorithm 3* to help estimate vehicle displacement and velocity.

The acceleration expressed in the navigation frame may be combined with GPS measurements of position and velocity to perform state estimation of the vehicle in the navigation frame. This is done with a Kalman filter. In order to construct the Kalman filter, the vehicle dynamic equations are written in state space form. The state vector consist of the displacement, velocity, and acceleration in each component direction; that is north and east.

$$x(k) = x(k-1) + \dot{x}(k-1) \cdot T_s + \frac{1}{2} \cdot \ddot{x}(k-1) \cdot T_s^2$$

$$\dot{x}(k) = \dot{x}(k-1) + \ddot{x}(k-1) \cdot T_s$$

$$\ddot{x}(k) = \ddot{x}(k-1) \qquad |^2$$

In matrix form we have

$$\begin{bmatrix} x(k) \\ \dot{x}(k) \\ \ddot{x}(k) \end{bmatrix} = \begin{bmatrix} 1 & T_s & 1/2 \cdot T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x(k-1) \\ \dot{x}(k-1) \\ \ddot{x}(k-1) \end{bmatrix}$$

The above equation may be written in compact notation.

$$\mathbf{x}(k) = \mathbf{\Phi} \cdot \mathbf{x}(k-1)$$

---

[2] The acceleration is assumed not to change over one time step.

Here **x** is the state vector of displacement, velocity, and acceleration.

In state space form, the Kalman filter may be expressed in orthogonal north and east directions as in *Algorithm 3*. Here, two channels are used to independently evaluate the state estimates, north and east.

Algorithm 3

Prediction:

$$\mathbf{x}_{\mathrm{north}}^{-}(k) = \mathbf{\Phi} \cdot \mathbf{x}_{\mathrm{north}}(k-1) \qquad\qquad \mathbf{x}_{east}^{-}(k) = \mathbf{\Phi} \cdot \mathbf{x}_{east}(k-1)$$

$$\mathbf{P}_{\mathrm{north}}^{-} = \mathbf{\Phi} \cdot \mathbf{P}_{\mathrm{north}}(k-1) \cdot \mathbf{\Phi}^{\mathrm{T}} + \mathbf{Q} \qquad\qquad \mathbf{P}_{east}^{-} = \mathbf{\Phi} \cdot \mathbf{P}_{east}(k-1) \cdot \mathbf{\Phi}^{\mathrm{T}} + \mathbf{Q}$$

Update:

$$\Delta\boldsymbol{\varepsilon}_{north}(k) = \begin{bmatrix} \mathrm{LatGps}(k) \\ \mathrm{VelGps}_{north}(k) \\ \mathrm{f}_{\mathrm{north}}^{\mathrm{n}}(k) \end{bmatrix} - \mathbf{x}_{\mathrm{north}}^{-}(k) \qquad \Delta\boldsymbol{\varepsilon}_{east}(k) = \begin{bmatrix} \mathrm{LongGps}(k) \\ \mathrm{VelGps}_{east}(k) \\ \mathrm{f}_{east}^{\mathrm{n}}(k) \end{bmatrix} - \mathbf{x}_{east}^{-}(k)$$

$$\mathbf{K}_{\mathrm{north}} = \mathbf{P}_{\mathrm{north}}^{-} \cdot (\mathbf{P}_{\mathrm{north}}^{-} + \mathbf{R})^{-1} \qquad\qquad \mathbf{K}_{east} = \mathbf{P}_{east}^{-} \cdot (\mathbf{P}_{east}^{-} + \mathbf{R})^{-1}$$

$$\mathbf{x}_{\mathrm{north}}(k) = \mathbf{x}_{\mathrm{north}}^{-}(k) + \mathbf{K}_{\mathrm{north}} \cdot \Delta\boldsymbol{\varepsilon}_{north}(k) \qquad \mathbf{x}_{east}(k) = \mathbf{x}_{east}^{-}(k) + \mathbf{K}_{east} \cdot \Delta\boldsymbol{\varepsilon}_{east}(k)$$

$$\mathbf{P}_{\mathrm{north}}(k) = (\mathbf{I} - \mathbf{K}_{\mathrm{north}}) \cdot \mathbf{P}_{\mathrm{north}}^{-} \qquad\qquad \mathbf{P}_{east}(k) = (\mathbf{I} - \mathbf{K}_{east}) \cdot \mathbf{P}_{east}^{-}$$

In *Algorithm 3*, the state to be estimated is **x**, and the measurements LatGps and LongGps are the latitude and longitude in meters displaced from an arbitrary starting point. These values are derived from the GPS readings[3]. Likewise VelGps represents the respective velocities as measured from GPS. The matrices **Q** and **R** are the process and measurement noise covariance, respectively. In practice **Q** and **R** are tuned for desired performance. *Algorithm 3* is an iterative process; it is executed every time a new GPS reading is acquired. It consist of two steps: a prediction step (indicated by a minus superscript), which computes an estimate of the state **x** before the latest GPS measurements are used, and an update step which uses the latest GPS and acceleration measurements. An error term, $\Delta\boldsymbol{\varepsilon}$, is formed between the measured and the predicted state. The Kalman gain, **K**, informs the algorithm how much weight to place on the error update. This weighted error term is used to update the state estimation. Since this process is repeated every time a new GPS reading arrives, the algorithm tracks the evolution of the state through time.

3. Experimental Setup

 Algorithms 1 to *3* were used to post-process data obtained from a small radio controlled (R/C) car shown in Figure 2. The car was driven in loops around a local area and data from the INS and GPS were recorded. The purpose of this experiment was to demonstrate the use of the algorithms and explore implementation issues for practical vehicle attitude and position estimation.

---

[3] Because of the local flat earth approximation, the conversion from degrees of Latitude and Longitude is scalar. For our location in Hampton, VA we have 110,980 meters per degree Latitude and 88,892 meters per degree Longitude.
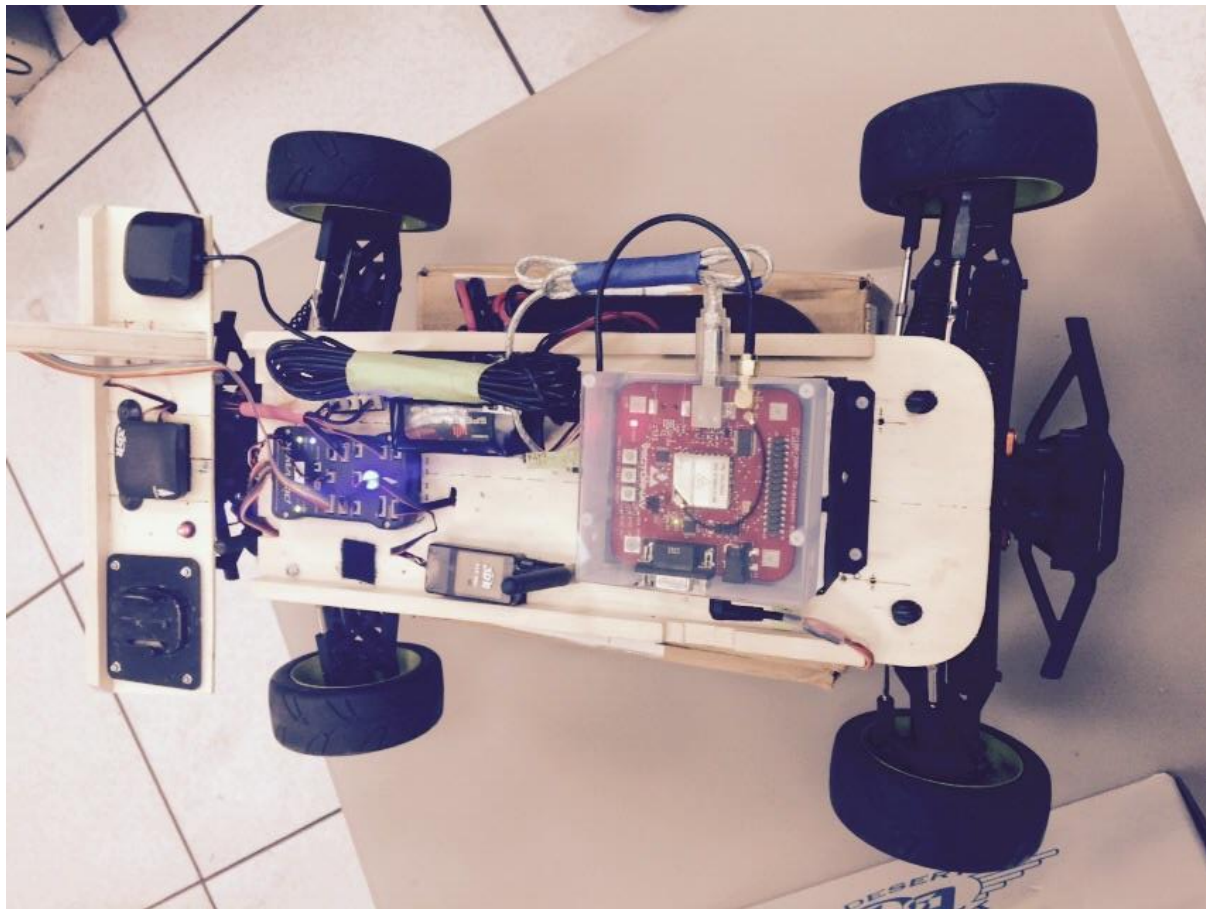
Figure 2. Radio controlled car used to gather data for navigation algorithms. Red circuit card is the VN200.

The R/C car was an HPI Trophy Truggy Flux (part #:107018). A modification made to the vehicle was the replacement of the factory plastic body and wing with pieces of plywood in order to have an easy surface to mount the electronics. Suspension modifications were made to accommodate the extra weight from additional batteries and electronics. The factory 2-channel 2.4GHz radio was replaced with a 7-channel Spektrum DX-7s; this allowed additional channels to be used in conjunction with the autopilot on board, which was not used for this experiment. The factory off-road tires and rims were replaced with slightly smaller but much smoother and more rigid street tires in order to increase traction on the paved surfaces. With all batteries and modifications, the R/C car weighed 12.45 lbs., which is approximately 2 lbs. heavier than stock.

For this experiment a BeagleBoard Xm and a VN-200 GPS/INS were used to collect data. The VN-200 was configured to output a binary message stream which was saved to a USB flash drive for post-processing in MATLAB. To reduce this noise in the accelerometer data, a layer of Dubro 1/4 inch latex foam rubber was mounted in between the VN-200 and BeagleBoard. Additional noise reduction was realized by using a noise reduction filter within the VN-200 architecture.

4. Experimental Results

The data recorded from the radio controlled car was post-processed to obtain estimates of vehicle attitude and position. Estimates were compared with those given by the VN200. Figure 3 shows the attitude of the car during a segment of the test. Since the test car was driven on a flat track, only yaw is shown. Pitch and roll are assumed to be negligible.
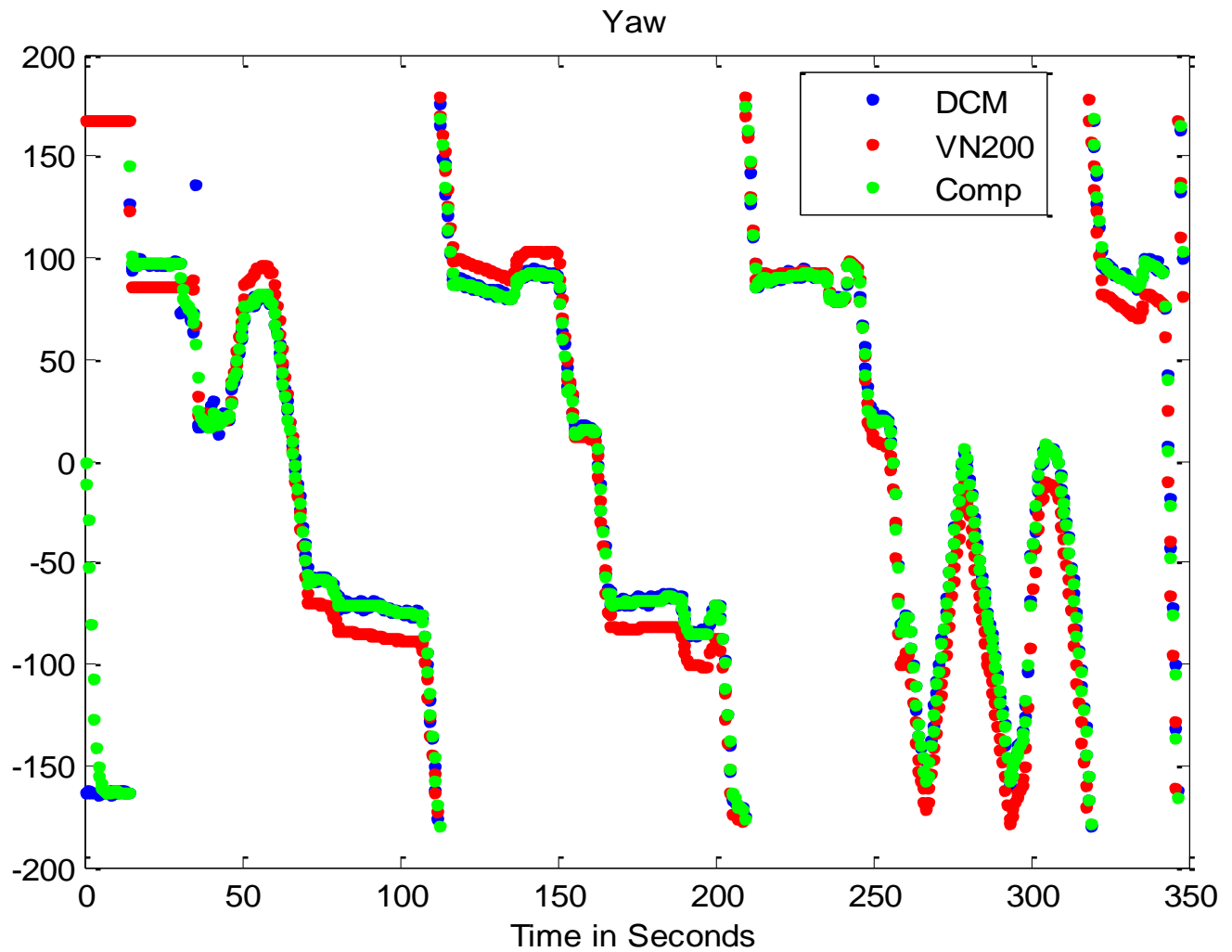

Figure 3. Yaw angle in degrees. Blue is from Directional Cosine Matrix, red is from VN200, and green is from complementary filter.
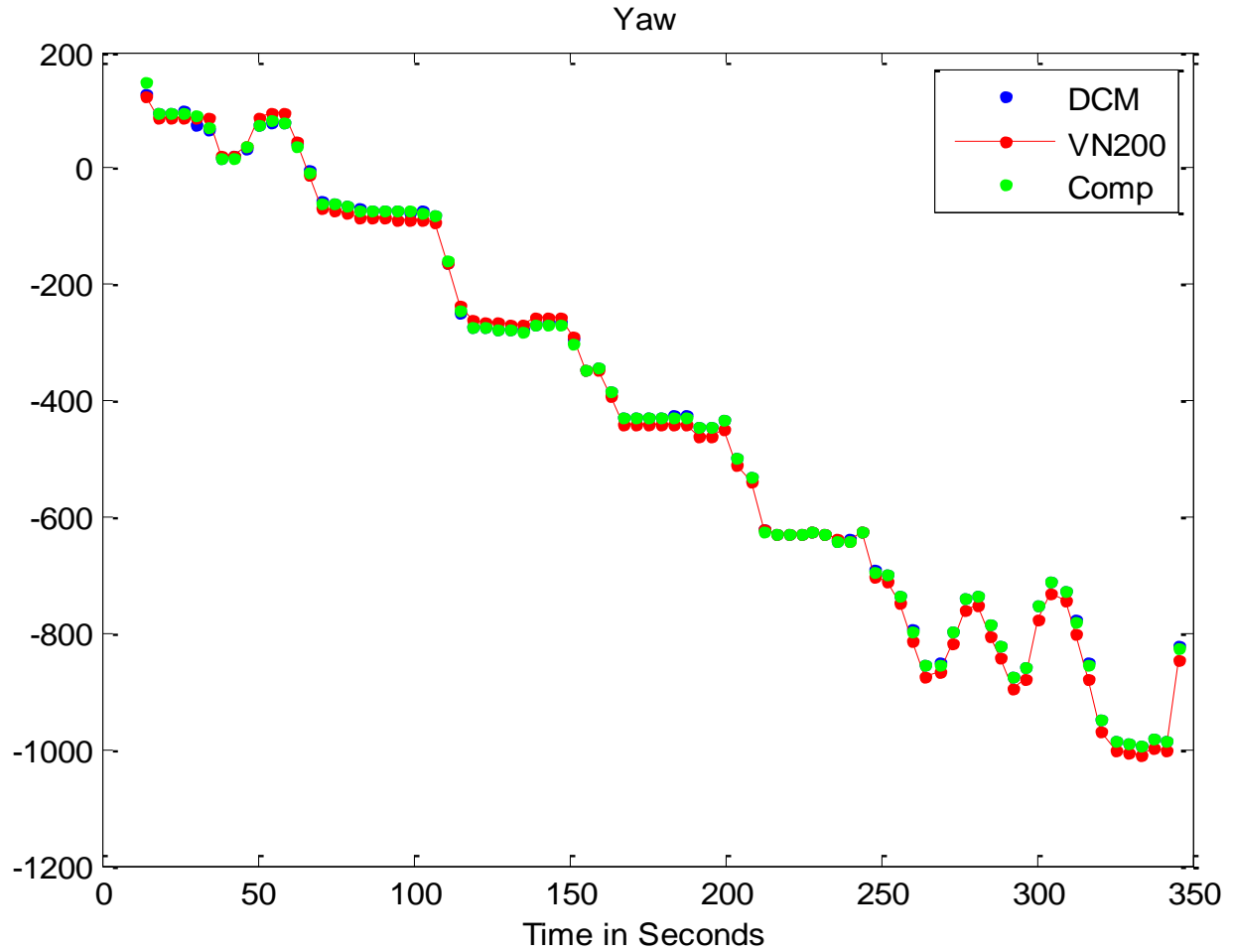
Figure 4. Yaw angle in degrees using unwrap

As can be seen in Figure 3 and Figure 4, the estimates obtained from the DCM using the magnetometer only and those of the complementary filter closely follow the yaw computed by the VN200. Part of the difference may be attributed to the fact that the VN200 uses true north as the yaw reference while the estimates of the DCM and complementary filter use magnetic north. This would account for a difference of approximately $10^\circ 42'$ for the test track location. A zoomed in section is shown in Figure 5. Here, one can see by comparing the DCM solution with the complementary filter that the complementary filter is combining the magnetometer and gyro measures to produce a smooth attitude estimate.
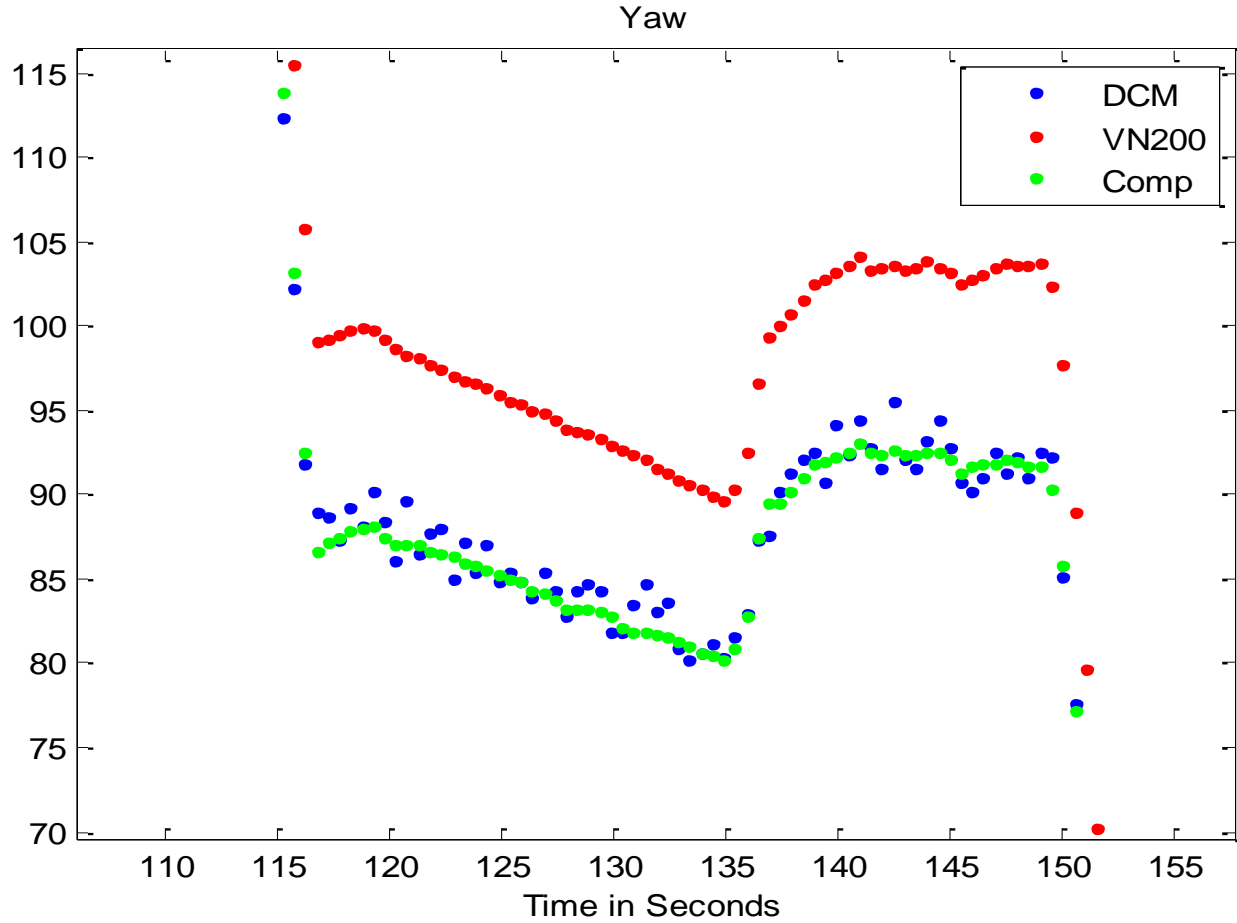
Figure 5. Zoomed in section of yaw estimation showing smoothing action of complementary filter (green line).

The attitude information is used by the inertial navigation system as described in *Algorithm 3* to compute estimates of vehicle position. Vehicle position was computed in two independent channels using a Kalman filter to integrate the dynamic model with GPS readings to obtain a smooth position estimate. The plots shown in Figure 6 and Figure 7 are for the eastward velocity and position and those of Figure 8 and Figure 9 are northward.
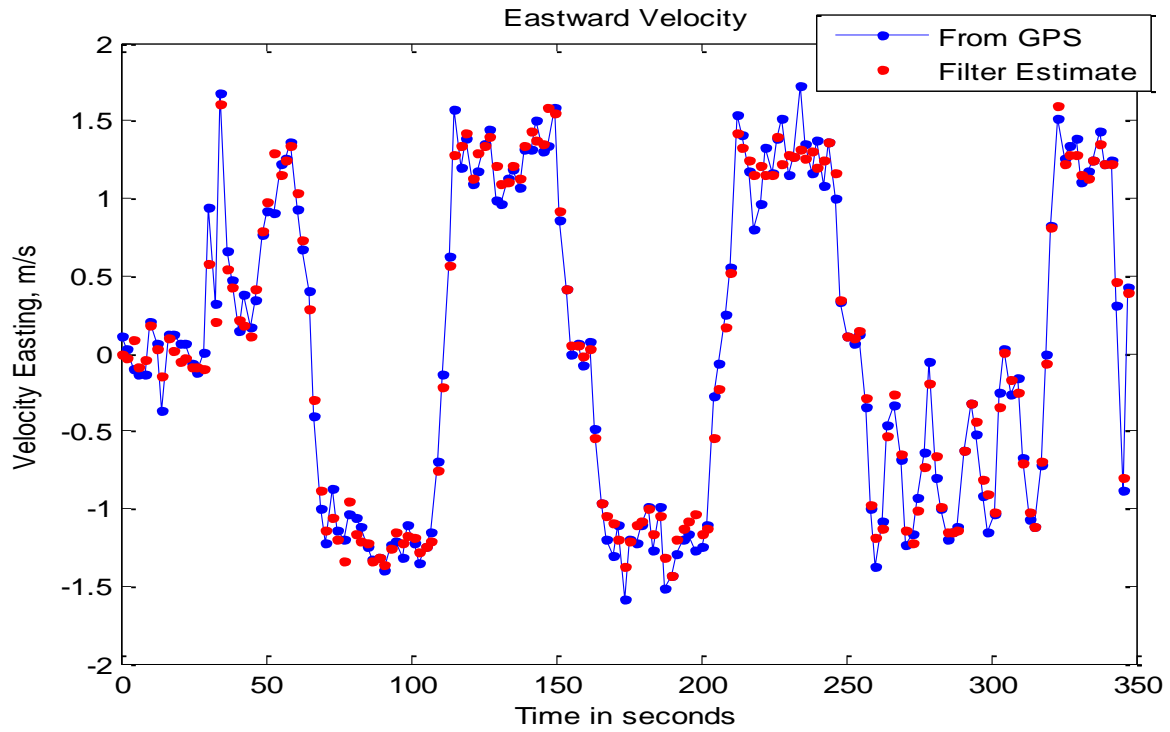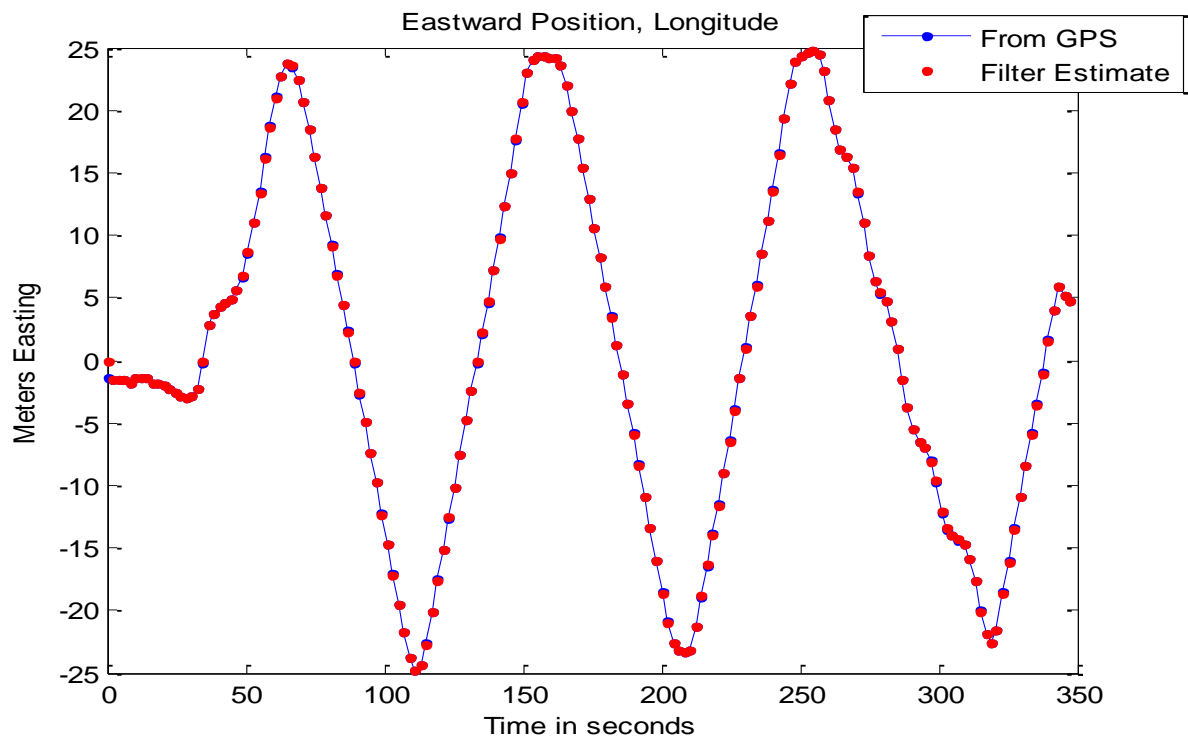
Figure 6. Eastward Velocity
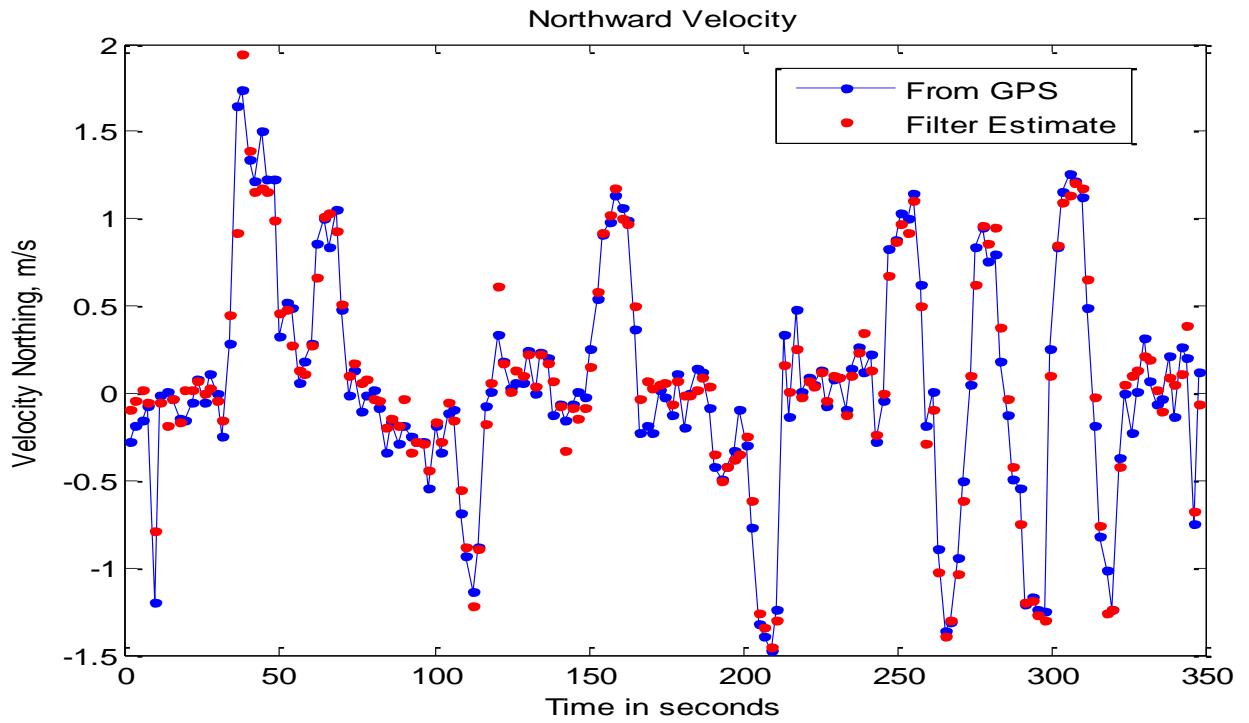

Figure 7. Eastward Position
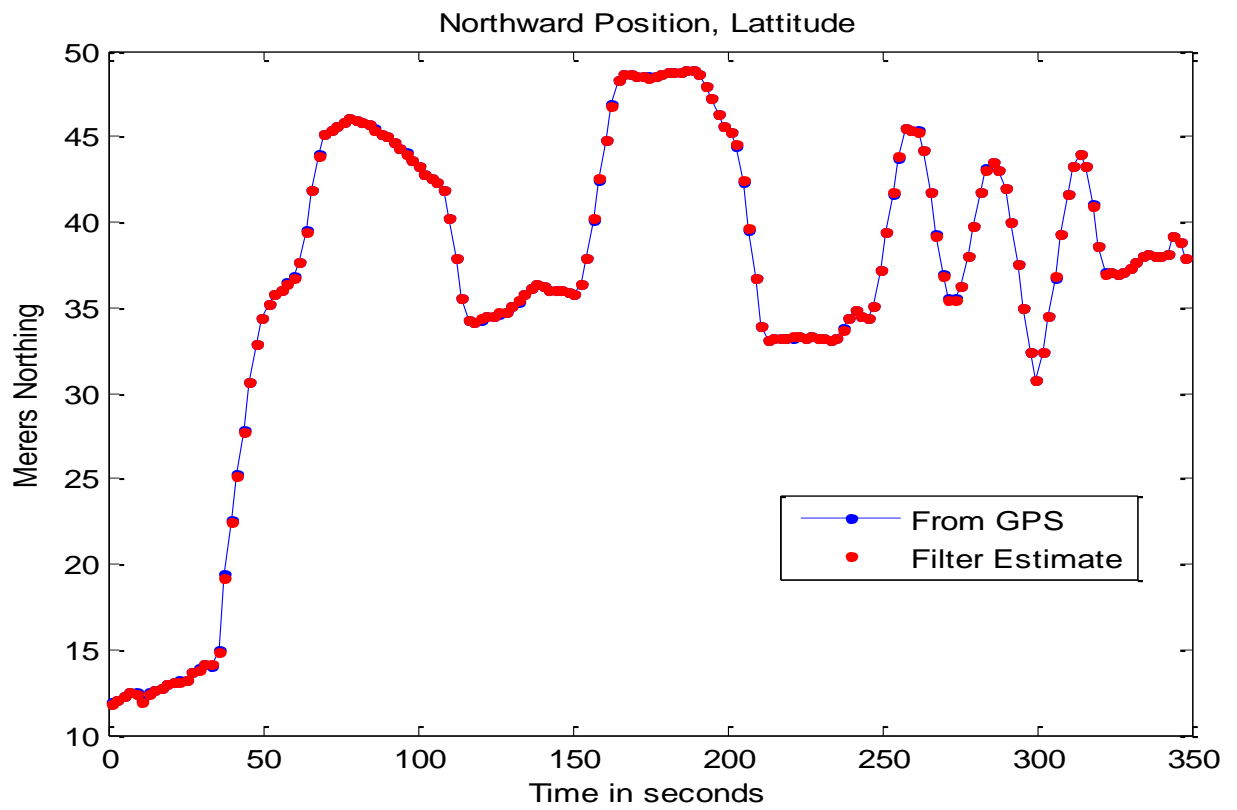
Figure 8. Northward Velocity

Figure 9. Northward Position.

Figures 6-9 show the velocity and position tracking of the Kalman filter over a large portion of the test path. As can be seen, close tracking is maintained throughout. In order to see an advantage gained by incorporating INS with GPS, a zoomed in section of the vehicle location is shown in Figure 10 and Figure 11.
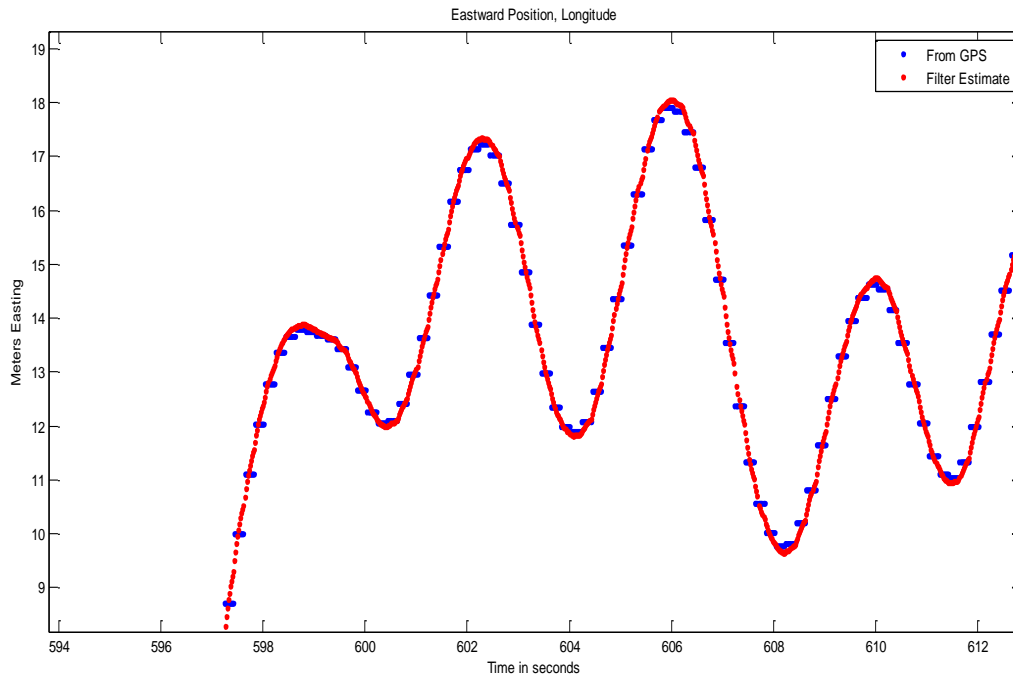

Figure 10. Figure illustrating improvement of Eastward position estimates, red, by Kalman filter.
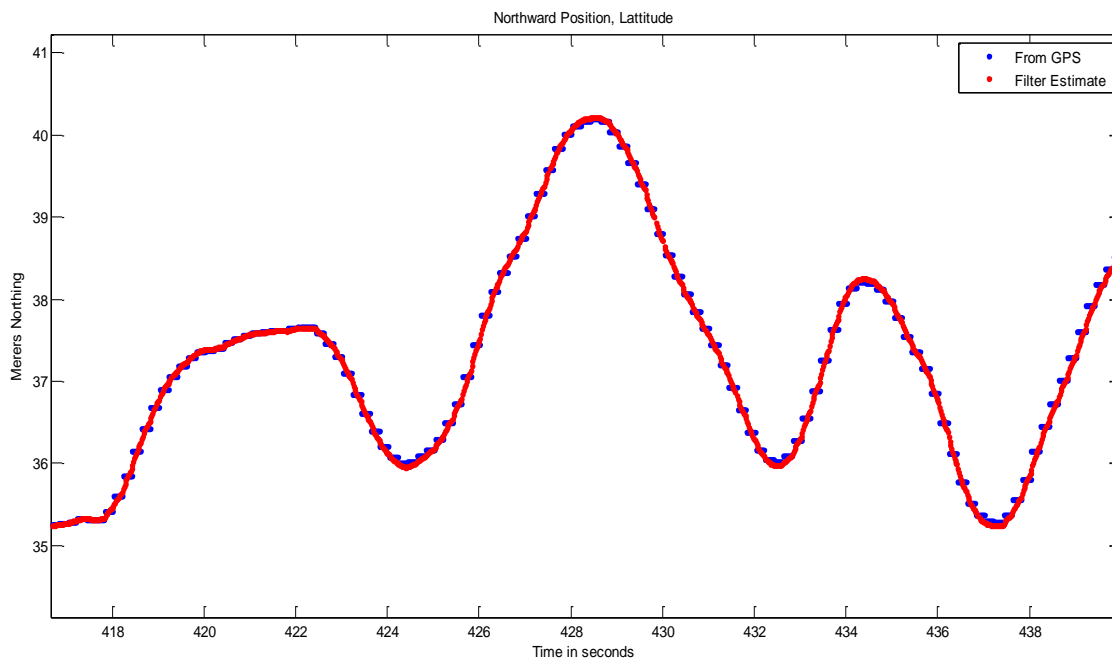

Figure 11. Figure illustrating improvement of Northward position estimates, red, by Kalman filter.

Figure 10 and Figure 11 illustrate the ability of the Kalman filter to use the dynamic model to fill in position measurements between GPS updates.

5. Conclusions

This tutorial describes a simple method for the integration of INS and GPS under a restrictive set of assumptions. The assumptions of a flat nonrotating earth with constant fields brought much simplification to the navigation problem. The purpose of this document is to offer a simple introduction into the complex field of strapdown inertial navigation systems and GPS/INS integration. The basic principles covered here may be extended to more complex navigation and filtering problems.

Appendix A

In this appendix the Kalman filter used in *Algorithm 3* will be derived from Bayes' theorem [4]. Consider the state space equation show below.

$$x_k = Ax_{k-1} + q_{k-1}$$

$$y_k = Hx_k + r_k$$

Here x_k is the *n*-dimension state at time step *k*, *A* is the state transition matrix, y_k is the vector of sensor measurements and *H* describes the connection between the states and measurements. The vectors q_{k-1} and r_k are independent zero mean Gaussian process and measurement noise vectors respectively. Since the system described above is linear and excited by Gaussian noise, the state and measurement vectors must also be Gaussian. The probability density function, *pdf*, for a multivariate Gaussian distribution is shown below where we use the notation $N(\,random\ variable \mid mean, covarience\,)$.

$$N(\,x \mid m, P\,) = \frac{1}{(2\pi)^{n/2}|P|^{1/2}} exp\left(-\frac{1}{2}(x-m)^T P^{-1}(x-m)\right)$$

In the above $|P|$ is the determinant of the covariance matrix *P* and *m* is the mean state vector. The Kalman filter has two steps: prediction and update. In the prediction an estimate of the state at time step *k* is computed based on past measurements $y_{1:k-1}$ before measurement y_k is taken. In the update the measurement y_k is used to update the state estimation. The prediction and update steps occur at every time step making the Kalman filter a recursive algorithm. The recursive formulas are derived below.

For the prediction at time step *k*, consider the joint *pdf* shown below conditioned on past measurements $y_{1:k-1}$; here the equality follows from conditional probability.

$$p(\,x_k, x_{k-1} \mid y_{1:k-1}\,) = p(\,x_k \mid x_{k-1}, y_{1:k-1}\,)p(\,x_{k-1} \mid y_{1:k-1}\,)$$

Because the first term on the right hand side has no dependence on the measurements *y*, we may write the above equation as shown below.

$$p(\,x_k, x_{k-1} \mid y_{1:k-1}\,) = p(\,x_k \mid x_{k-1}\,)p(\,x_{k-1} \mid y_{1:k-1}\,)$$

Both sides may be integrated over x_{k-1} to obtain the Chapman-Kolmogorov equation.

$$\int_{-\infty}^{\infty} p(x_k, x_{k-1} \mid y_{1:k-1})dx_{k-1} = \int_{-\infty}^{\infty} p(x_k \mid x_{k-1})p(x_{k-1} \mid y_{1:k-1}) \, dx_{k-1}$$

The joint distribution now becomes the marginal distribution conditioned on past measurements $y_{1:k-1}$. One can solve this equation to obtain the desired *pdf* for the predicted state mean m_k^- and covariance P_k^-.

$$p(x_k \mid y_{1:k-1}) = \int_{-\infty}^{\infty} p(x_k \mid x_{k-1})p(x_{k-1} \mid y_{1:k-1}) \, dx_{k-1}$$

The *pdf*s in the above equation are Gaussian and may be expressed as shown below.

$$N(x_k \mid m_k^-, P_k^-) = \int_{-\infty}^{\infty} N(x_k \mid Ax_{k-1}, Q)N(x_{k-1} \mid m_{k-1}, P_{k-1})dx_{k-1}$$

Here Q is the process noise convenience matrix. Given the past values of m_{k-1} and P_{k-1} from the previous time step the above equation may be solved for the current predicted mean and covariance values m_k^- and P_k^-. These values are estimates based on measurements up to and including *k-1*, but do not include the measurement y_k. The solution is shown below[4].

$$m_k^- = Am_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

The above equations are predictions of the mean of the state x_k and covariance at time step k, but without using the measurement y_k. This is to say all measurements before k were used to give a best estimate of the state and covariance based on past measurements, state estimations and knowledge of the state and measurement distributions. The next step is to update this estimate using the information obtained in measurement y_k.

Bayes' rule is used in the update step. Recall Bayes' as shown below for random variables *A* and *B*.

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)}$$

In general Bayes' rule can also be expressed with an additional conditional variable *C* as shown below.

$$p(A \mid B, C) = \frac{p(B \mid A, C)p(A \mid C)}{p(B \mid C)}$$

For our derivation, we can use Bayes' rule as shown below. Here we have an estimation of x_k that includes y_k for the update step.

$$p(x_k \mid y_k, y_{1:k-1}) = \frac{p(y_k \mid x_k, y_{1:k-1})p(x_k \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})}$$

---

[4] For the Gaussian linear system case the Chapman-Kolmogorov equation may be explicitly solved. Alternatively note that the expected value E$\{x_k\}$ = E$\{Ax_{k-1} + q_{k-1}\}$ = Am_{k-1}. Also the covariance cov$\{Ax_{k-1} + q_{k-1}\}$ = $E\{(Ax_{k-1} + q_{k-1} - Am_{k-1})(Ax_{k-1} + q_{k-1} - Am_{k-1})^T\} = AE\{(x_{k-1}x_{k-1}^T - m_{k-1}m_{k-1}^T)\}A^T + Q = AP_{k-1}A^T + Q$ where *E* is the expectation operator and the time dependence of *Q* has been suppressed.

Since our derivation involves Gaussian distributions, we are only interested in computing the mean and covariance of the state update. To this end we will only examine the exponent of the distribution; which contains the mean and covariance, x_k and P_k respectively. We can express this as the proportion shown below after recognizing that $p(y_k \mid x_k, y_{1:k-1}) = p(y_k \mid x_k)$ since there is no dependency on $y_{1:k-1}$ in our state space model. Also we have written $y_k, y_{1:k-1}$ in compact notation as $y_{1:k}$.

$$p(x_k \mid y_{1:k}) \propto p(y_k \mid x_k)p(x_k \mid y_{1:k-1})$$

The *pdf*s in the right hand side of this equation are known. The first term is simply the measurement equation *pdf* and the second is that determined in the previous prediction step. We may now express the update step in terms of the respective Gaussian distributions.

$$N(x_k \mid m_k, P_k) \propto N(y_k \mid Hx_k, R)N(x_k \mid m_k^-, P_k^-)$$

Here R is the measurement noise covariance matrix and m_k^-, P_k^- are the estimates computed in the prediction step. The goal is to find m_k, P_k which are the updated mean and covariance based on the current measurement y_k. One can do this by completing the square in the product of the two Gaussian *pdf*s given above.

$$N(x_k \mid m_k, P_k) \propto$$

$$\frac{1}{(2\pi)^{n/2}|R|^{1/2}} exp\left(-\frac{1}{2}(y_k - Hx_k)^T R^{-1}(y_k - Hx_k)\right)\frac{1}{(2\pi)^{n/2}|P_k^-|^{1/2}} exp\left(-\frac{1}{2}(x_k - m_k^-)^T (P_k^-)^{-1}(x_k - m_k^-)\right)$$

Consider the exponent only. The expansion is shown below. In this expansion, care must be taken to maintain the order since this is a matrix equation and multiplication does not commute.

$$-\frac{1}{2}\left(y_k^T R^{-1}y_k - y_k^T R^{-1}Hx_k - x_k^T H^T R^{-1}y_k + x_k^T H^T R^{-1}Hx_k + x_k^T (P_k^-)^{-1}x_k - x_k^T (P_k^-)^{-1}m_k^-\right.$$
$$\left. - (m_k^-)^T (P_k^-)^{-1}x_k + (m_k^-)^T (P_k^-)^{-1}m_k^-\right)$$

We need to shape the above expression into the form $(x_k - m_k)^T P_k^{-1}(x_k - m_k)$ in order to identify the updated mean m_k and covariance P_k. In order to do this we gather like terms and drop any term that does not contain x_k. The dropped terms can be ignored since they do not contribute to the expression for the Gaussian mean and covariance. Factoring out x_k and recalling that the transpose of a covariance matrix is just the matrix itself we obtain the expression below.

$$-\frac{1}{2}\left\{x_k^T\left(H^T R^{-1}H + (P_k^-)^{-1}\right)x_k - x_k^T\left(H^T R^{-1}y_k + (P_k^-)^{-1}m_k^-\right) - \left(y_k^T R^{-1}H + (m_k^-)^T (P_k^-)^{-1}\right)x_k\right\}$$

In order to make the math more tractable while completing the square we define some variables.

$$V = H^T R^{-1}H + (P_k^-)^{-1}$$

$$b = H^T R^{-1}y_k + (P_k^-)^{-1}m_k^-$$

Using the above, we can write our expression as shown.

$$x_k^T V x_k - x_k^T b - b^T x_k$$

In order to complete the square we can add a term to this expression as long as it does not contain x_k.

$$x_k^T V x_k - x_k^T b - b^T x_k + b^T V^{-1} b$$

Also, recall that $VV^{-1} = I$ can be inserted into any term to help complete the square.

$$x_k^T V x_k - x_k^T VV^{-1} b - b^T VV^{-1} x_k + b^T V^{-1} VV^{-1} b = (x_k - V^{-1}b)^T V (x_k - V^{-1}b)$$

The fact that the above equality holds can be verified by expanding the right side to get the left. We now have the desired form and can identify the updated mean and convenience matrices.

$$m_k = V^{-1}b = \left(H^T R^{-1} H + (P_k^-)^{-1}\right)^{-1}\left(H^T R^{-1} y_k + (P_k^-)^{-1} m_k^-\right)$$

$$P_k = V^{-1} = \left(H^T R^{-1} H + (P_k^-)^{-1}\right)^{-1}$$

To make the computations more efficient, we use the matrix inverse lemma shown below where *A,U,C* and *V* are simply placeholders to demonstrate the lemma.

$$(A + UCV)^{-1} = A^{-1} - A^{-1} U (C^{-1} + VA^{-1}U)^{-1} V A^{-1}$$

$$P_k = \left((P_k^-)^{-1} + H^T R^{-1} H\right)^{-1} = P_k^- - P_k^- H^T (R + HP_k^- H^T)^{-1} HP_k^-$$

$$m_k = \{P_k^- - P_k^- H^T (R + HP_k^- H^T)^{-1} HP_k^-\}\left(H^T R^{-1} y_k + (P_k^-)^{-1} m_k^-\right)$$

$$m_k = P_k^- H^T R^{-1} y_k + m_k^- - P_k^- H^T (R + HP_k^- H^T)^{-1} HP_k^- H^T R^{-1} y_k - P_k^- H^T (R + HP_k^- H^T)^{-1} Hm_k^-$$

We need to simplify the y_k term.

$$P_k^- H^T R^{-1} - P_k^- H^T (R + HP_k^- H^T)^{-1} HP_k^- H^T R^{-1} =$$

$$P_k^- H^T R^{-1} - P_k^- H^T (R + HP_k^- H^T)^{-1}(R + HP_k^- H^T) R^{-1} + P_k^- H^T (R + HP_k^- H^T)^{-1} =$$

$$P_k^- H^T (R + HP_k^- H^T)^{-1}$$

Replacing the y_k coefficient with the above simplified expression one may now write m_k as shown below. The previously derived prediction equations and covariance update are also shown for completeness.

Prediction step:

$$m_k^- = Am_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

Update step:

$$m_k = m_k^- + P_k^- H^T (R + HP_k^- H^T)^{-1}(y_k - Hm_k^-)$$

$$P_k = P_k^- - P_k^- H^T (R + HP_k^- H^T)^{-1} HP_k^-$$

The equations are iterated every time step as in *Algorithm 3*.

Appendix B

The purpose of this section is to provide MATLAB code to implement *Algorithms 2* and *3*. The code snippets are partial and are not intended to be functioning code, but are presented only as an aid to the

reader. Both blocks of code run in a loop that executes every time step. The code for *Algorithm 2* is shown below.

```
mnorm = norm([mx(k) my(k)]); %mx(k) my(k) from magnetometer
mx(k) = mx(k)/mnorm; my(k) = my(k)/mnorm;
psi = atan2(-my(k),mx(k));
DCM = [cos(psi)  -sin(psi);
       sin(psi)   cos(psi)];
YAW(k) = 180/pi*psi;    %YAW from magnetometer
gyro  = [ 0     -wz(k); %wz(k) from z axis gyro
          wz(k)   0  ];
omaga = 0.5*(Rh'*DCM - DCM'*Rh);
OM = gyro + kp*omaga; % kp >= 1, gain constant
nOM = norm(OM); %For exact solution
A = eye(2) - OM.*sin(nOM*Ts)./nOM + (OM^2).*(1-cos(nOM*Ts))/nOM^2;
Rh = Rh*A';
yaw(k) = 180/pi*atan2(Rh(2,1),Rh(1,1)); %yaw, Complementary Filter
```

The above code returns `Rh` which is an improved estimate of the DCM. From `Rh` the yaw angle is calculated.

The code for *Algorithm 3* is shown below. This block is for the longitude channel. The inputs are the time, GPS longitude, GPS velocity, and eastward acceleration in the navigation frame.

```
Ts = Time(k)-Time(k-1); %Time step
A = [1 Ts 0.5*Ts^2; 0 1 Ts; 0 0 1];
x(:,k) = A*x(:,k-1); %Predict
P = A*P*A' + Q;
err(:,k) = [LongGpsE(k); VelNedE(k); accel_nav(2,k)] - x(:,k);
S = P + R;
K = P*(S\I);
x(:,k) = x(:,k) + K*err(:,k); %Update
P = (I - K)*P;
```

The estimated state `x(:,k)` combines both the INS and GPS measurements for improved estimates of position, velocity, and acceleration. The latitude channel follows in like manner.

Bibliography

[1] D. Titterton and J. Weston, Strapdown Inertial Navigation Techology, London, United Kingdom: The Institution of Engineering and Technology, 2004.

[2] [Online]. Available: http://aerostudents.com/files/avionics/InertialNavigationSystems.pdf.

[3] R. Mahony, M. Euston, J. Kim, P. Coote and T. Hamel, "A Non-Linear Observer for Attitude Estimation of a Fixed-Wing Unmanned aerial Vehicle Without GPS Measurements," *Transactions of the Institute of Measurement and Control,* pp. 699-717, 2010.

[4] S. Särkkä, Bayesian Filtering and Smoothing, Cambridge, United Kingdom: Cambridge University Press, 2013.

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-09-2015	Technical Memorandum	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Short Tutorial on Inertial Navigation System and Global Positioning System Integration	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Smalling, Kyle M.; Eure, Kenneth W.	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
	999182.02.60.07.03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
NASA Langley Research Center Hampton, VA 23681-2199	L-20602

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
National Aeronautics and Space Administration Washington, DC 20546-0001	NASA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
	NASA-TM-2015-218803

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 04
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The purpose of this document is to describe a simple method of integrating Inertial Navigation System (INS) information with Global Positioning System (GPS) information for an improved estimate of vehicle attitude and position. A simple two dimensional (2D) case is considered. The attitude estimates are derived from sensor data and used in the estimation of vehicle position and velocity through dead reckoning within the INS. The INS estimates are updated with GPS estimates using a Kalman filter. This tutorial is intended for the novice user with a focus on bringing the reader from raw sensor measurements to an integrated position and attitude estimate. An application is given using a remotely controlled ground vehicle operating in assumed 2D environment. The theory is developed first followed by an illustrative example.

15. SUBJECT TERMS

GPS; Kalman Filtering; Navigation; Tutorial

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	22	19b. TELEPHONE NUMBER *(Include area code)* (757) 864-9658