FOI-R--0460--SE
April 2002
1650-1942

Scientific report

Bengt Boberg
Sven-Lennart Wirkander

Robust Navigation Using GPS and INS: Comparing the Kalman Estimator and the Particle Estimator.

System Technology Division
SE-172 90 STOCKHOLM
Sweden

Swedish Defence Research Agency
System Technology Division
SE-172 90 STOCKHOLM
Sweden

FOI-R--0460--SE
April 2002
1650-1942

Scientific report

Bengt Boberg
Sven-Lennart Wirkander

Robust Navigation Using GPS and INS: Comparing the Kalman Estimator and the Particle Estimator.

Issuing organization	Report number, ISRN	Report type
Swedish Defence Research Agency System Technology Division SE-172 90 STOCKHOLM Sweden	FOI-R--0460--SE	Scientific report
	Research area code: Electronic Warfare	
	Month year: April 2002	Project no.: E6034
	Customers code: Commissioned Research	
	Sub area code: Electronic Warfare including Electro-magnetic Weapons and Protection	

Author/s (editor/s)	
Bengt Boberg Sven-Lennart Wirkander	Project manager: Bengt Boberg
	Approved by: Monica Dahlén
	Sponsoring agency:
	Scientifically and technically responsible: Martin Hagström

Report title

Robust Navigation Using GPS and INS: Comparing the Kalman Estimator and the Particle Estimator.

Abstract

The robustness of navigation systems based on a Global Satellite Navigation System (GNSS) is crucial and can be achieved in many ways, e.g. by using smart antennas (adaptive beamforming antennas or switched beam antennas) or complementing the GNSS with a jamming resistent sensor system.
In order to investigate this latter method, a comparison has been performed between a relatively new type of state estimator, called the Particle method based on a linear Kalman Estimator (KE) when both are applied to the problem of combining information from a GNSS with an Inertial Navigation System (INS). This report is a detailed description of this comparison.
The comparison of the two estimators regards essentially their robustness towards different types of unmodeled errors in the three acceleration measurements. These errors consist of different combinations of white noise components and constant components (biases).
KE uses a continuous linear error model. The task for the KE is to estimate the errors of the INS solution by using the difference between external measurements of velocity and position (from e.g. GPS) and the velocity and position as calculated by the INS.
As the PE does not require linear system equations, it uses a nonlinear full state discrete model. The same external measurements are used as for the KE.english

Keywords

Particle Estimator, Particle Filter, Navigation Equations, GNSS, GPS, Jamming, Inertial Navigation

Further bibliographic information	Language
	English
ISSN: 1650-1942	Pages: 63
Distribution: By sendlist	Price Acc. to pricelist Security classification Unclassified

Utgivare	Rapportnummer, ISRN	Klassificering
Totalförsvarets forskningsinstitut Avdelningen för Systemteknik SE-172 90 STOCKHOLM Sweden	FOI-R--0460--SE	Vetenskaplig rapport
	Forskningsområde: Telekrig	
	Månad, år: April 2002	Projektnummer: E6034
	Verksamhetsgren: Uppdragsfinansierad verksamhet	
	Delområde: Telekrigföring med EM-vapen och skydd	

Författare/redaktör	
Bengt Boberg Sven-Lennart Wirkander	Projektledare: Bengt Boberg
	Godkänd av: Monica Dahlén
	Uppdragsgivare/kundbeteckning: FM
	Tekniskt och/eller vetenskapligt ansvarig: Martin Hagström

Rapportens titel

Robust navigering med GPS och TN: Jämförelse av Kalman- och partikelestimator

Sammanfattning

Störfasthet är en kritisk egenskap hos sådana navigeringssystem som utnyttjar satellitnavigering (GNSS). Den kan åstadkommas på olika sätt, t ex genom elektrisk styrning av antennerna eller genom att komplettera GNSS med ett icke störningskänsligt sensorsystem.
För att undersöka den senare av dessa metoder har en jämförelse gjorts mellan en förhållandevis ny typ av tillståndsestimator, den s k partikelestimatorn (PE), och Kalmanestimatorn (KE), då båda appliceras på problemet att optimalt kombinera information från ett GNSS och ett tröghetnavigeringssystem (INS). Föreliggande rapport innehåller en detaljerad beskrivning av denna jämförelse.
Jämförelsen mellan de två estimatorerna avser väsentligen deras robusthet mot olika slags icke-modellerade störningar hos de tre accelerationsmätningarna. Dessa fel består av kombinationer av vitt brus och bias.
KE använder en underliggande kontinuerlig, linearisera felmodell. Dess uppgift är att estimera INS-felen genom att använda skillnaden mellan externa hastighets- och positionsmätningar och de hastigheter och positioner som INS räknar fram.
Eftersom PE inte kräver linjära systemekvationer, använder den en olinjär diskret modell med fulla tillstånd samt samma externa mätningar som KE.english

Nyckelord

Partikelestimator, Partikelfilter, Navigeringsekvationer, GNSS, GPS, Störning, Tröghetsnavigering

Övriga bibliografiska uppgifter	Språk
	Engelska

ISSN	Antal sidor
1650-1942	63
Distribution: Enligt missiv	Pris: Enligt prislista Sekretess: Öppen

Contents

1. Introduction

1.1 Background

Jamming of navigation systems is a current threat since Global Navigation Satellite Systems (GNSS) became a key component in modern navigation systems. The use of the United States' Global Positioning System (GPS) is expanding, and recently the European Union has approved funding of the Galileo project, which is the European equivalent to GPS.

Generally, new demands are being solved by using more advanced navigation systems. The demands from the US defence, the American program Common Guidance Inertial Measurement Unit (CG-IMU), and also from car manufacturers, to design small, cheap and well performing integrated sensor systems based on GPS and Micro Electro Mechanical System (MEMS) sensors, are important driving forces for the development of future navigation systems.

For the development of personal positioning technologies, the most important force is the United States Federal Communication Commission Enhanced 911 service (FCC-E911 service). FCC-E911 sets explicitly defined requirements on position accuracy of emergency calls made from mobile phones: 67 % CEP 50 m (Circular Error Probable, i.e. 67 % of the measurements within a circle with radius 50 m) and 95 % CEP 150 m. GPS is the only globally applicable technology that meets the FCC-E911 requirement. The European Union is promoting, but so far not requiring, an equivalent E112 service.

The robustness of navigation systems is crucial and can be achieved in many ways. One way is to protect the GPS receiver by using different software and hardware solutions. Another is to support the GPS system with external complementary measurements. Two efficient methods have shown to be, see [9]:

- supporting the GPS receiver with a complementary jamming robust sensor system, e.g. integrating GPS and Inertial Measurement Unit (IMU). This is treated in [6] and [7].

- protecting the GPS receiver by using smart antennas (adaptive beamforming antennas and switched beam antennas), thereby achieving spatial null steering or spatial beam forming, [5].

There are of course also other interesting sensors to support the navigation system during GPS outages and intentional jamming. An interesting example of sensor combination in order to achieve a robust navigation system is the use of an INS and a radar altimeter in combination with a height database to determine the position by correlating height profiles.

However, the terrain navigation system has poor performance at high altitudes due to the radar maximun mesuring range and/or lack of information in the terrain due to flatness, eg. sea surface. In [8] the problem of integrated aircraft navigation is treated and specifically how to integrate inertial navigation with terrain aided positioning.

1.2 Aim of work

This report describes how a relatively new type of state estimator, called the Particle Estimator (PE), has been evaluated and compared with the well established method based on a linear Kalman Estimator (KE). The GPS receiver position and velocity are used as measurements to the estimators.

The estimators use different forms of the navigation equations which conventionally, and also here, have been expressed in the so called n-frame (local geographic), [10]. For completeness, and as a preparation for future work, we also present these equations in a couple of different frames, namely the e-frame (earth centered/earth fixed, ECEF) and the i-system (inertial). All these frames are derived as special cases of a completely general frame, called the a-frame.

The task for the KE is to estimate the errors of an Inertial Navigation System (INS) by using the difference between the GPS measurements of velocity and position and the integrated INS velocity and position. For this purpose, the KE uses a linearization of the INS system as its system model. The error estimates are used as corrections to the values calculated by the INS, in order to get an optimal full state Kalman estimate. The PE, on the other hand, uses the GPS values as measurements that are directly applied to a full state model of the system to be tracked, thereby producing directly an estimate of the position and velocities that are closer to the truth than are the INS values themselves.

The KE is continuous and assumes white Gaussian noise and is based on linearized navigation error dynamics expressed in the n-frame. The PE, on the other hand, is discrete and also assumes Gaussian noise distribution,

In the original and well cited paper [1] and also in [11], the PE is applied to a nonlinear tracking problem. A framework for positioning, navigation and tracking using PE:s (sequential Monte Carlo methods) is developed in [2]. Also [8] mentioned above uses the PE.

The PE does neither require white Gaussian noise distribution nor linear equations. This is a reason why the full states can be estimated directly, without use of the linearized error equations as in the KE case. The computational load for the PE increases with the complexity of the problem, e.g. the number of states. However, considering the current development in the computing field, the PE seems to be a promising method. For example, a combination of both PE and KE can be applied, as described in [3].

1.3 Results

The two estimators were compared, essentially regarding their robustness against different types of errors in the three acceleration measurements. At all simulations, these errors contained a white noise component. If, in addition, constant components (biases) were added to some of them, the Kalman Estimator gave different results:

- When none of the measurement errors contained any bias, the estimates were correct within its 3σ-limit. They were also substantially smoother than the GPS measurements, thus showing a low frequency filtering effect.
- Also when the measurement of the first acceleration component contained a bias, the estimates were smooth, but the estimate of the velocity component corresponding to the biased measurement were itself biased. This bias could be decreased by increasing the elements of the process noise matrix in the estimator.
- When all three acceleration measurement components were biased, all three velocity component estimates became also biased. This effect could also be reduced by an increase of the process noise matrix elements (note, however, that such an increase always makes the estimator more sensitive to noise).

The PE estimates were, on the contrary, independent of biases in the acceleration measurements, indicating that the external measurements of the velocities and positions from the GPS had a dominating influence, since these measurements themselves contain no bias. This is also confirmed by the fact that they had essentially the same noise characteristics as the estimates.

1.4 Suggestions for Future Work

- The Particle Estimator used in this work uses the simplest form of importance sampling distribution, namely the state prior distribution, that does not depend on the measurements. Investigate what type of more general importance sampling distribution that could be used in order to improve the Particle Estimates in our application. For a treatment of the case with such a general sampling, see [13].
- Analyze the PE performance theoretically using the Cramer-Rao bound, [8].
- Investigate the possibility of using so called Rao-Blackwellization, which is a method of exploiting the existing linearities of the system as much as possible. The KE is used for the linear parts, and the PE only for the remaining nonlinear ones. This method is described in [12].
- A natural continuation of the PE/KE-comparison could be to replace the KE with a discrete version of the Extended Kalman Filter (EKF), which is a nonlinear generalization of the KE, utilizing linearization of the system equations around the estimate. This would possibly be a more fair comparison as regards the Kalman method. The use of non-linear estimation methods is still more motivated by the need to investigate the use of smart antennas (adaptive beamforming antennas or switched beam antennas), because this necessitates the use of GNSS raw observables (pseudo range and phase), giving rise to further non-linearities.
- Use the Earth Centered Earth Fixed (ECEF) reference frame instead of the navigation frame for the navigation equations (derived in this report).

2. Navigation Equations

2.1 Reference Frames.

The estimators use different forms of the navigation equations which conventionally, and also here, have been expressed in the so called n-frame (local geographic), [10]. For completeness, and as a preparation for future work, we also present these equations in a couple of different frames, namely the e-frame (earth centered/earth fixed, ECEF) and the i-system (inertial). All these frames are derived as special cases of a completely general frame, called the a-frame.

2.2 Navigation equations in a fixed inertial i-frame.

Inertial navigation is basically integration of inertially sensed acceleration with respect to time. In an inertial frame the differential equations needed to be solved are

$$\begin{aligned} \frac{d}{dt}\dot{x}^i &= \ddot{x}^i = g^i\left(x^i\right) + f^i, \\ \frac{d}{dt}x^i &= \dot{x}^i, \end{aligned} \tag{2.1}$$

where x^i is the position vector coordinates in the i-frame, $g^i\left(x^i\right)$ is the acceleration components due to the gravitational field and f^i the signals sensed by the accelerometers.

2.3 Navigation equations in an arbitrary concentric rotating a-frame.

Let the a- and the i-frame be concentric. The a-frame rotates with respect to the i-frame with the angular rate components in the a-frame ω_{ia}^a (the superscript a denotes the used reference frame). The transformation of a position vector from the a-frame to the i-frame is given by

$$x^i = C_a^i x^a, \tag{2.2}$$

using the transformation matrix C_a^i, whose the corresponding time derivative is given by

$$\dot{C}_a^i = C_a^i \Omega_{ia}^a, \tag{2.3}$$

where Ω_{ia}^a is the skew-symmetric matrix with elements from $\omega_{ia}^a = (\omega_1, \omega_2, \omega_3)^T$ according to

$$\Omega_{ia}^a = [\omega_{ia}^a \times] = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{2.4}$$

For the general case the second time derivative is also needed:

$$\ddot{C}_a^i = C_a^i \dot{\Omega}_{ia}^a + C_a^i \Omega_{ia}^a \Omega_{ia}^a. \tag{2.5}$$

Differentiating (2.2) twice with respect to time and using (2.3) and (2.5) yield

$$\begin{aligned}\frac{d}{dt}x^i &= \dot{C}^i_a x^a + C^i_a \dot{x}^a,\\ \frac{d}{dt}\dot{x}^i &= \ddot{C}^i_a x^a + 2\dot{C}^i_a \dot{x}^a + C^i_a \ddot{x}^a\\ &= C^i_a \left(\dot{\Omega}^a_{ia} + \Omega^a_{ia}\Omega^a_{ia}\right) x^a + 2C^i_a \Omega^a_{ia}\dot{x}^a + C^i_a \ddot{x}^a. \end{aligned} \tag{2.6}$$

Solving for $\ddot{x}^a$ and combining with (2.1) gives the navigation equations coordinatized (mechanized) in an arbitrary a-frame

$$\begin{aligned}\frac{d}{dt}\dot{x}^a &= -2\Omega^a_{ia}\dot{x}^a - \left(\dot{\Omega}^a_{ia} + \Omega^a_{ia}\Omega^a_{ia}\right) x^a + f^a + g^a\left(x^a\right), \\ \frac{d}{dt}x^a &= \dot{x}^a, \end{aligned} \tag{2.7}$$

where f^a and $g^a\left(x^a\right)$ are the specific force and gravity vector components in the a system, i.e., $f^a = C^a_i f^i$ and $g^a\left(x^a\right) = C^a_i g^i\left(C^i_a x^a\right)$. In this strapdown inertial navigation system the accelerometer signals, f^b, are the specific force components in the vehicle's own system (b). In order to be used in (2.7), they need to be transformed as

$$f^a = C^a_b f^b, \tag{2.8}$$

where the transformation matrix is given by the differential equation

$$\dot{C}^a_b = C^a_b \Omega^b_{ab}. \tag{2.9}$$

The elements in the skew-symmetric matrix $\Omega^b_{ab} = \left[\omega^b_{ab}\times\right]$ are based on the inertially sensed angular rotation, expressed in the b system, ω^b_{ib}. ω^b_{ab} can be split in two parts as

$$\begin{aligned}\omega^b_{ab} &= \omega^b_{ib} - \omega^b_{ia}\\ &= \omega^b_{ib} - C^b_a \omega^a_{ia}, \end{aligned} \tag{2.10}$$

where ω^a_{ia} is a constant if a is the e-frame (depends on earth rate alone) and depends on the navigation solution if a is the n-frame (earth rate and transport rate).

2.3.1 Navigation equations in the e-frame. The following derivation is based on (2.7), where all indices a are substituted by e. Then, assuming that the angular velocity of the earth relative to inertial space, Ω^e_{ie}, is constant, i.e. $\dot{\Omega}^e_{ie} = 0$

$$\begin{aligned}\frac{d}{dt}\dot{x}^e &= -2\Omega^e_{ie}\dot{x}^e - \Omega^e_{ie}\Omega^e_{ie}x^e + f^e + g^e\left(x^e\right), \\ \frac{d}{dt}x^e &= \dot{x}^e. \end{aligned} \tag{2.11}$$

The accelerometer signals are transformed from the body frame b to the earth frame e as

$$f^e = C^e_b f^b, \tag{2.12}$$

where C^e_b is determined by the differential equation

$$\dot{C}^e_b = C^e_b \Omega^b_{eb}. \tag{2.13}$$

The elements of the matrix $\Omega^e_{eb} = [\omega^e_{eb}\times]$ are based on the measured angular rates ω^b_{ib} as

$$\omega^b_{eb} = \omega^b_{ib} - C^b_e \omega^e_{ie}. \tag{2.14}$$

2.4 Navigation Equations in the e-frame in component form.

With the definitions

$$x^e = \begin{pmatrix} x_1^e & x_2^e & x_3^e \end{pmatrix}^T, \tag{2.15}$$
$$\dot{x}^e = \begin{pmatrix} v_1^e & v_2^e & v_3^e \end{pmatrix}^T, \tag{2.16}$$
$$f^e = \begin{pmatrix} f_1^e & f_2^e & f_3^e \end{pmatrix}^T, \tag{2.17}$$

and the expressions

$$\omega_{ie}^e = \begin{pmatrix} 0 & 0 & \omega_e \end{pmatrix}^T \quad \text{or} \quad \Omega_{ie}^e = \begin{pmatrix} 0 & -\omega_e & 0 \\ \omega_e & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.18}$$

and

$$g^e\left(x^e\right) = -\frac{kM \begin{pmatrix} x_1^e & x_2^e & x_3^e \end{pmatrix}^T}{\left(\left(x_1^e\right)^2 + \left(x_2^e\right)^2 + \left(x_3^e\right)^2\right)^{3/2}} \tag{2.19}$$

for the earth's rotation rate and gravitation, respectively, (2.11) can be written in component form as

$$\begin{aligned}
\dot{v}_1^e &= 2\omega_e v_2^e + \omega_e^2 x_1^e + f_1^e - \frac{GMx_1^e}{\left(\left(x_1^e\right)^2 + \left(x_2^e\right)^2 + \left(x_3^e\right)^2\right)^{3/2}}, \qquad (2.20)\\
\dot{v}_2^e &= -2\omega_e v_1^e + \omega_e^2 x_2^e + f_2^e - \frac{GMx_2^e}{\left(\left(x_1^e\right)^2 + \left(x_2^e\right)^2 + \left(x_3^e\right)^2\right)^{3/2}},\\
\dot{v}_3^e &= f_3^e - \frac{kGx_3^e}{\left(\left(x_1^e\right)^2 + \left(x_2^e\right)^2 + \left(x_3^e\right)^2\right)^{3/2}},\\
\dot{x}_1^e &= v_1^e,\\
\dot{x}_2^e &= v_2^e,\\
\dot{x}_3^e &= v_3^e.
\end{aligned}$$

where G is the constant of gravitaion and M is the mass of earth.

2.5 Navigation equations in the n-frame.

The n-frame serves to define local directions for the velocity vector determined in the e-frame. The components in the n system for the movement relative to the earth is expressed as a transformation of the same vector's components in the e system according to

$$v^n = C_e^n \dot{x}^e. \tag{2.21}$$

Note that, in general, $v^n \neq \dot{x}^n$, where x^n is defined as

$$x^n = C_e^n x^e, \tag{2.22}$$

Indeed, differentiating this expression gives

$$\begin{aligned} \dot{x}^n &= C_e^n \dot{x}^e + \dot{C}_e^n x^e \\ &= v^n + C_e^n \Omega_{ne}^e x^e \\ &= v^n + C_e^n \Omega_{ne}^e C_n^e x^n \\ &= v^n + \Omega_{ne}^n x^n \\ &= v^n - \Omega_{en}^n x^n. \end{aligned} \tag{2.23}$$

Substituting $\dot{x}^e = C_n^e v^n$ in the navigation equations coordinatized in the e-frame, (2.11), and using (2.23) together with

$$\frac{d}{dt}\left(C_n^e v^n\right) = C_n^e \frac{d}{dt}\left(v^n\right) + \frac{d}{dt}\left(C_n^e\right) v^n = C_n^e \frac{d}{dt}\left(v^n\right) + C_n^e \Omega_{en}^n v^n = C_n^e \left(\frac{d}{dt} v^n + \Omega_{en}^n v^n\right), \tag{2.24}$$

gives

$$\begin{aligned} C_n^e \left(\frac{d}{dt} v^n + \Omega_{en}^n v^n\right) &= -2\Omega_{ie}^e C_n^e v^n - \Omega_{ie}^e \Omega_{ie}^e x^e + f^e + g^e\left(x^e\right), \\ \frac{d}{dt} v^n &= -\Omega_{en}^n v^n - C_e^n 2\Omega_{ie}^e C_n^e v^n - C_e^n \Omega_{ie}^e \Omega_{ie}^e x^e + C_e^n f^e + C_e^n g^e\left(x^e\right), \\ \frac{d}{dt} v^n &= -\left(\Omega_{en}^n + 2\Omega_{ie}^n\right) v^n - C_e^n \Omega_{ie}^e \Omega_{ie}^e x^e + f^n + g^n\left(x^n\right), \end{aligned} \tag{2.25}$$

where $g^n\left(x^n\right) = C_e^n g^e\left(C_n^e x^n\right)$.

It is common to introduce a gravity vector $\bar{g}^n\left(x^n\right)$ which consists of the centrifugal acceleration and the gravitational vector as

$$\bar{g}^n\left(x^n\right) = -C_e^n \Omega_{ie}^e \Omega_{ie}^e C_n^e x^n + g^n\left(x^n\right), \tag{2.26}$$

leading to the following differential equation for the velocities:

$$\frac{d}{dt} v^n = -\left(\Omega_{en}^n + 2\Omega_{ie}^n\right) v^n + f^n + \bar{g}^n. \tag{2.27}$$

The accelerometer signals are transformed from the body frame b to the earth frame n as

$$f^n = C_b^n f^b, \tag{2.28}$$

where C_b^n is determined by the following differential equation:

$$\dot{C}_b^n = C_b^n \Omega_{nb}^b. \tag{2.29}$$

The elements of the matrix $\Omega_{nb}^n = [\omega_{nb}^n \times]$ are based on the measured angular rates ω_{ib}^b according to

$$\omega_{nb}^b = \omega_{ib}^b - C_n^b \omega_{in}^n. \tag{2.30}$$

2.6 Navigation Equations in the n-frame in component form

In order to write (2.27) in component form, we introduce the components of v^n as

$$v^n = \begin{pmatrix} v_N & v_E & v_D \end{pmatrix}^T. \tag{2.31}$$

Now, the latitude (L), longitude (λ), and the height above the earth (h) can be defined through the differential relations

$$\begin{aligned} \dot{L} &= \frac{v_N}{R+h}, \\ \dot{\lambda} &= \frac{v_E}{(R+h)\cos L}, \\ \dot{h} &= -v_D. \end{aligned} \tag{2.32}$$

The angular rate components used in (2.27) are easily realized to be

$$\begin{aligned} \omega_{ie}^n &= \begin{pmatrix} \omega_e \cos L & 0 & -\omega_e \sin L \end{pmatrix}^T, \\ \omega_{en}^n &= \begin{pmatrix} \dot{\lambda}\cos L & -\dot{L} & -\dot{\lambda}\sin L \end{pmatrix}^T \end{aligned} \tag{2.33}$$

in vector form, or in matrix form:

$$\begin{aligned} \Omega_{ie}^n &= \begin{pmatrix} 0 & \omega_e \sin L & 0 \\ -\omega_e \sin L & 0 & -\omega_e \cos L \\ 0 & \omega_e \cos L & 0 \end{pmatrix}, \\ \Omega_{en}^n &= \begin{pmatrix} 0 & \dot{\lambda}\sin L & -\dot{L} \\ -\dot{\lambda}\sin L & 0 & -\dot{\lambda}\cos L \\ \dot{L} & \dot{\lambda}\cos L & 0 \end{pmatrix}. \end{aligned} \tag{2.34}$$

Also, define the components of the specific forces and the generalized gravity according to

$$\begin{aligned} f^n &= \begin{pmatrix} f_N & f_E & f_D \end{pmatrix}^T, \\ \bar{g}^n &= \begin{pmatrix} \bar{g}_N & \bar{g}_E & \bar{g}_D \end{pmatrix}^T. \end{aligned} \tag{2.35}$$

Then (2.27) can be written

$$\begin{aligned} \dot{v}_N &= f_N + \bar{g}_N - 2\omega_e v_E \sin L + \dot{L} v_D - \dot{\lambda} v_E \sin L, \\ \dot{v}_E &= f_E + \bar{g}_E + 2\omega_e v_N \sin L + 2\omega_e v_D \cos L + \dot{\lambda} v_N \sin L + \dot{\lambda} v_D \cos L, \\ \dot{v}_D &= f_D + \bar{g}_D - 2\omega_e v_E \cos L - \dot{\lambda} v_E \cos L - \dot{L} v_N, \end{aligned} \tag{2.36}$$

or, with the above expressions for $\dot{L}$ and $\dot{\lambda}$,

$$\begin{aligned} \dot{v}_N &= f_N + \bar{g}_N - 2\omega_e v_E \sin L + \frac{v_N v_D}{R+h} - \frac{v_E^2}{R+h}\tan L, \\ \dot{v}_E &= f_E + \bar{g}_E + 2\omega_e \left(v_N \sin L + v_D \cos L\right) + v_E \left(\frac{v_N \tan L + v_D}{R+h}\right), \\ \dot{v}_D &= f_D + \bar{g}_D - 2\omega_e v_E \cos L - \frac{v_E^2 + v_N^2}{R+h}. \end{aligned} \tag{2.37}$$

By introducing $\omega_{nb}^b = \begin{pmatrix} \omega_x & \omega_y & \omega_z \end{pmatrix}^T$ and $\omega_{ib}^b = \begin{pmatrix} p & q & r \end{pmatrix}^T$ and by using $\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n$ the following expression can be derived

$$\omega_x \hat{=} p - c_{11}\left(\omega_e \cos L + \frac{v_E}{R+h}\right) + c_{12}\frac{v_N}{R+h} + c_{13}\left(\sin L + \frac{v_E \tan L}{R+h}\right) \tag{2.38}$$

$$\omega_y \hat{=} q - c_{21}\left(\omega_e \cos L + \frac{v_E}{R+h}\right) + c_{22}\frac{v_N}{R+h} + c_{23}\left(\sin L + \frac{v_E \tan L}{R+h}\right) \tag{2.39}$$

$$\omega_z \hat{=} r - c_{31}\left(\omega_e \cos L + \frac{v_E}{R+h}\right) + c_{32}\frac{v_N}{R+h} + c_{33}\left(\sin L + \frac{v_E \tan L}{R+h}\right) \tag{2.40}$$

If the elements of the transformation matrix C_b^n from the body coordinate system to the n-system are called $c_{ij}, i, j = 1, 2, 3$, then the differential equation for this matrix can, according to (2.29), be written in component form as

$$\dot{c}_{11} = c_{12}\omega_z - c_{13}\omega_y, \dot{c}_{12} = c_{13}\omega_x - c_{11}\omega_z, \dot{c}_{13} = c_{11}\omega_y - c_{12}\omega_x, \quad (2.41)$$

$$\dot{c}_{21} = c_{22}\omega_z - c_{23}\omega_y, \dot{c}_{22} = c_{23}\omega_x - c_{21}\omega_z, \dot{c}_{23} = c_{21}\omega_y - c_{22}\omega_x, \quad (2.42)$$

$$\dot{c}_{31} = c_{32}\omega_z - c_{33}\omega_y, \dot{c}_{32} = c_{33}\omega_x - c_{31}\omega_z, \dot{c}_{33} = c_{31}\omega_y - c_{32}\omega_x. \quad (2.43)$$

3. Error dynamics

3.1 Nonlinear System Error Dynamics

The navigation equations expressed in the general a-frame are differential equations for velocity and positions where the forcing functions are the sensed acceleration components. The error dynamics equations describe how the sensor errors affect the position and velocity errors. These equations can be derived by perturbing the navigation equation (2.7). Subtracting the equation that results from driving (2.7) with the measured accelerations $\tilde{f}^a$ and the angular velocities $\tilde{\Omega}^a_{ia}$ from the one driven by the corresponding true values ($\tilde{f}^a$ and $\tilde{\Omega}^a_{ia}$, respectively) results in

$$\begin{aligned}
\frac{d}{dt}\left(\dot{\tilde{x}}^a - \dot{x}^a\right) &= f\left(\tilde{x}^a, \dot{\tilde{x}}^a, \tilde{\Omega}^a_{ia}, \dot{\tilde{\Omega}}^a_{ia}, \tilde{f}^a, \tilde{g}^a\left(\tilde{x}^a\right)\right) \\
&\quad -f\left(x^a, \dot{x}^a, \dot{\Omega}^a_{ia}, \Omega^a_{ia}, f^a, g^a\left(x^a\right)\right) \qquad (3.1) \\
&= \frac{\partial f}{\partial x^a}\left(\tilde{x}^a - x^a\right) + \frac{\partial f}{\partial \dot{x}^a}\left(\dot{\tilde{x}}^a - \dot{x}^a\right) + \frac{\partial f}{\partial \Omega^a_{ia}}\left(\tilde{\Omega}^a_{ia} - \Omega^a_{ia}\right) \\
&+ \frac{\partial f}{\partial \dot{\Omega}^a_{ia}}\left(\dot{\tilde{\Omega}}^a_{ia} - \dot{\Omega}^a_{ia}\right) + \frac{\partial f}{\partial f^a}\left(\tilde{f}^a - f^a\right) + \frac{\partial f}{\partial g^a}\left(\tilde{g}^a - g^a\right) + ..., \\
&\quad \frac{d}{dt}\left(\tilde{x}^a - x^a\right) = \dot{\tilde{x}}^a - \dot{x}^a, \qquad (3.2)
\end{aligned}$$

where the function f is the right hand side of the upper equation in (2.7). The "tilde" entities correspond to the solution with measured driving funtions, and the "tilde-free" to the true solution. Observe that the gravitation error $\tilde{g}^a\left(\tilde{x}^a\right) - g^a\left(x^a\right)$ depends on two error sources, namely $\tilde{x}^a - x^a$ and error in the model of the gravitation, i.e. the function g^a itself.

The expansion can be interpreted as a Taylor expansion around the approximate value. Differences with respect to the "Taylor expansion point " $\left(\tilde{x}^a, \dot{\tilde{x}}^a, \tilde{\Omega}^a_{ia}, \dot{\tilde{\Omega}}^a_{ia}, \tilde{f}^a, \tilde{g}^a\right)$ represent negative errors and partial derivatives are evaluated in this point. In the sequel the error dynamics for higher order terms are neglected.

3.2 Linear Error Dynamics in the a-frame

Applying (3.1) to (2.7) with the perturbations denoted by the prefix "δ", we have the linearized error equations

$$\begin{aligned}
\frac{d}{dt}\delta\dot{x}^a &= -2\delta\Omega^a_{ia}\dot{x}^a - 2\Omega^a_{ia}\delta\dot{x}^a - \left(\delta\dot{\Omega}^a_{ia} + \delta\Omega^a_{ia}\Omega^a_{ia} + \Omega^a_{ia}\delta\Omega^a_{ia}\right)x^a \qquad (3.3) \\
&\quad -\left(\dot{\Omega}^a_{ia} + \Omega^a_{ia}\Omega^a_{ia} - \Gamma^a\right)\delta x^a + \delta f^a + \delta g^a \\
\frac{d}{dt}\delta x^a &= \delta\dot{x}^a,
\end{aligned}$$

where $\Gamma^a = \frac{\partial g^a}{\partial x^a}$. The δg^a-term arises as a consequence of possible modelling errors of the gravitation vector and is equal to $g^a(\tilde{x}^a) - g^a(x^a)$. The perturbation δf^a consists not only of accelerometer errors but also orientation errors. Taking the differential of the relation $f^a = C_b^a f^b$ gives

$$\delta f^a = \delta C_b^a f^b + C_b^a \delta f^b. \tag{3.4}$$

The differential δC_b^a is caused by errors in the orientation of the b-frame relative to the general a-frame. A vector of small error angles are introduced as $\phi^a = (\phi_1^a, \phi_2^a, \phi_3^a)^T$ and the skew-symmetric matrix equivalent is

$$\Phi^a = \begin{pmatrix} 0 & -\phi_3^a & \phi_2^a \\ \phi_3^a & 0 & -\phi_1^a \\ -\phi_2^a & \phi_1^a & 0 \end{pmatrix}. \tag{3.5}$$

The transformation from the true a-frame to an erroneously computed a-frame can be expressed as

$$\tilde{C}_b^a = (I - \Phi^a) C_b^a. \tag{3.6}$$

Then the differential of C_b^a is

$$\delta C_b^a = \tilde{C}_b^a - C_b^a = -\Phi^a C_b^a, \tag{3.7}$$

and when substituted into (3.4), it gives the acceleration error

$$\begin{aligned} \delta f^a &= -\Phi^a C_b^a f^b + C_b^a \delta f^b \\ &= -\Phi^a f^a + C_b^a \delta f^b \\ &= -[\phi^a \times] f^a + C_b^a \delta f^b \\ &= f^a \times \phi^a + C_b^a \delta f^b. \end{aligned} \tag{3.8}$$

Next, the dynamics of the error angles will be derived by first taking the differential of (2.9):

$$\delta \dot{C}_b^a = \delta C_b^a \Omega_{ab}^b + C_b^a \delta \Omega_{ab}^b, \tag{3.9}$$

where $\delta \Omega_{ab}^b = \tilde{\Omega}_{ab}^b - \Omega_{ab}^b$ is the perturbation in angular rate. If (3.7) is differentiated with respect to time and set equal to (3.9), this leads to the expression

$$-\dot{\Phi}^a C_b^a - \Phi^a C_b^a \Omega_{ab}^b = \delta C_b^a \Omega_{ab}^b + C_b^a \delta \Omega_{ab}^b, \tag{3.10}$$

where also the expression for $\dot{C}_b^a$ according to (2.9) is used. Substituting (3.7) results in the matrix equation

$$\dot{\Phi}^a = -C_b^a \delta \Omega_{ab}^b C_a^b, \tag{3.11}$$

and the corresponding vector equation

$$\dot{\phi}^a = -C_b^a \delta \omega_{ab}^b, \tag{3.12}$$

The gyros sense

$$\omega_{ib}^b = \omega_{ia}^b + \omega_{ab}^b = C_a^b \omega_{ia}^a + \omega_{ab}^b. \tag{3.13}$$

Perturbing this equation and using (3.7) as $\delta C_b^a = -\Phi^a C_b^a = (\delta C_a^b)^T$ leads to

$$\begin{aligned}\delta\omega_{ib}^{b} &= \delta C_{a}^{b}\omega_{ia}^{a} + C_{a}^{b}\delta\omega_{ia}^{a} + \delta\omega_{ab}^{b} \\ &= \left(-\Phi^{a}C_{b}^{a}\right)^{T}\omega_{ia}^{a} + C_{a}^{b}\delta\omega_{ia}^{a} + \delta\omega_{ab}^{b} \\ &= C_{a}^{b}\Phi^{a}\omega_{ia}^{a} + C_{a}^{b}\delta\omega_{ia}^{a} + \delta\omega_{ab}^{b},\end{aligned} \tag{3.14}$$

where we also have used the skew-symmetry of Φ^{a}, i.e. $\Phi^{aT} = -\Phi^{a}$.

Then (3.14) is solved for $\delta\omega_{ab}^{b}$ and then substituted into (3.12):

$$\begin{aligned}\dot{\phi}^{a} &= -C_{b}^{a}\delta\omega_{ab}^{b} \\ &= -C_{b}^{a}\left(\delta\omega_{ib}^{b} - C_{a}^{b}\Phi^{a}\omega_{ia}^{a} - C_{a}^{b}\delta\omega_{ia}^{a}\right) \\ &= -C_{b}^{a}\delta\omega_{ib}^{b} + \Phi^{a}\omega_{ia}^{a} + \delta\omega_{ia}^{a} \\ &= -C_{b}^{a}\delta\omega_{ib}^{b} + \left[\phi^{a}\times\right]\omega_{ia}^{a} + \delta\omega_{ia}^{a} \\ &= -C_{b}^{a}\delta\omega_{ib}^{b} - \omega_{ia}^{a}\times\phi^{a} + \delta\omega_{ia}^{a}.\end{aligned} \tag{3.15}$$

Using (3.3), (3.15), (3.8) and some vector matrix manipulations, we can summarize the dynamics of the error states $\begin{pmatrix} \phi^{a} & \delta\dot{x}^{a} & \delta x^{a} \end{pmatrix}^{T}$ as

$$\begin{aligned}\frac{d}{dt}\phi^{a} &= -\omega_{ia}^{a}\times\phi^{a} - C_{b}^{a}\delta\omega_{ib}^{b} + \delta\omega_{ia}^{a}, \\ \frac{d}{dt}\delta\dot{x}^{a} &= \left[f^{a}\times\right]\phi^{a} - 2\Omega_{ia}^{a}\delta\dot{x}^{a} - \left(\dot{\Omega}_{ia}^{a} + \Omega_{ia}^{a}\Omega_{ia}^{a} - \Gamma^{a}\right)\delta x^{a} \\ &\quad + C_{b}^{a}\delta f^{b} + \delta g^{a} + 2\dot{x}^{a}\times\delta\omega_{ia}^{a} + 2\Omega_{ia}^{a}\left[x^{a}\times\right]\delta\omega_{ia}^{a} + \left[x^{a}\times\right]\delta\dot{\omega}_{ia}^{a}, \\ \frac{d}{dt}\delta x^{a} &= \delta\dot{x}^{a},\end{aligned} \tag{3.16}$$

which also can be written in block matrix state space form as

$$\begin{aligned}\frac{d}{dt}\begin{pmatrix}\phi^{a}\\ \delta\dot{x}^{a}\\ \delta x^{a}\end{pmatrix} &= \begin{pmatrix}-\Omega_{ia}^{a} & 0 & 0\\ \left[f^{a}\times\right] & -2\Omega_{ia}^{a} & -\left(\dot{\Omega}_{ia}^{a} + \Omega_{ia}^{a}\Omega_{ia}^{a} - \Gamma^{a}\right)\\ 0 & I & 0\end{pmatrix}\begin{pmatrix}\phi^{a}\\ \delta\dot{x}^{a}\\ \delta x^{a}\end{pmatrix} \\ &\quad + \begin{pmatrix}-C_{b}^{a} & 0 & 0\\ 0 & C_{b}^{a} & I\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\delta\omega_{ib}^{b}\\ \delta f^{b}\\ \delta g^{a}\end{pmatrix} \\ &\quad + \begin{pmatrix}I & 0\\ 2\left[\dot{x}^{a}\times\right] + 2\Omega_{ia}^{a}\left[x^{a}\times\right] & \left[x^{a}\times\right]\\ 0 & 0\end{pmatrix}\begin{pmatrix}\delta\omega_{ia}^{a}\\ \delta\dot{\omega}_{ia}^{a}\end{pmatrix}.\end{aligned} \tag{3.17}$$

3.3 Linear Error Dynamics in the i-frame

Substituting i for a in (3.16) and using $\Omega_{ii}^{i} = 0$, $\dot{\Omega}_{ii}^{i} = 0$, $\delta\omega_{ii}^{i} = 0$, $\delta\dot{\omega}_{ii}^{i} = 0$ leads to

$$\begin{aligned}\frac{d}{dt}\phi^{i} &= -C_{b}^{i}\delta\omega_{ib}^{b}, \\ \frac{d}{dt}\delta\dot{x}^{i} &= \left[f^{i}\times\right]\phi^{i} + -\Gamma^{i}\delta x^{i} + C_{b}^{i}\delta f^{i} + \delta g^{i}, \\ \frac{d}{dt}\delta x^{i} &= \delta\dot{x}^{i},\end{aligned} \tag{3.18}$$

and the same substitutions in (3.17) lead to

$$\frac{d}{dt}\begin{pmatrix}\phi^i\\ \delta\dot{x}^i\\ \delta x^i\end{pmatrix} = \begin{pmatrix}0 & 0 & 0\\ [f^i\times] & 0 & \Gamma^i\\ 0 & I & 0\end{pmatrix}\begin{pmatrix}\phi^i\\ \delta\dot{x}^i\\ \delta x^i\end{pmatrix} + \begin{pmatrix}-C_b^i & 0 & 0\\ 0 & C_b^i & I\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\delta\omega_{ib}^b\\ \delta f^b\\ \delta g^i\end{pmatrix}. \tag{3.19}$$

3.4 Linear Error Dynamics in the e-frame

The constant earth rate ω_{ie}^e implies $\delta\omega_{ie}^e = 0$ and $\dot{\omega}_{ie}^e = 0$ as well as $\delta\dot{\omega}_{ie}^e = 0$. The resulting dynamics of the error states $\bar{x} = \begin{pmatrix}\phi^e & \delta\dot{x}^e & \delta x^e\end{pmatrix}^T$ can be written as

$$\begin{aligned}\frac{d}{dt}\phi^e &= -\omega_{ie}^e\times\phi^e - C_b^e\delta\omega_{ib}^b, \\ \frac{d}{dt}\delta\dot{x}^e &= [f^e\times]\,\phi^e - 2\Omega_{ie}^e\delta\dot{x}^e - (\Omega_{ie}^e\Omega_{ie}^e - \Gamma^e)\,\delta x^e + C_b^e\delta f^b + \delta g^e, \\ \frac{d}{dt}\delta x^e &= \delta\dot{x}^e,\end{aligned} \tag{3.20}$$

and in block matrix form

$$\frac{d}{dt}\begin{pmatrix}\phi^e\\ \delta\dot{x}^e\\ \delta x^e\end{pmatrix} = \begin{pmatrix}-\Omega_{ie}^e & 0 & 0\\ [f^e\times] & -2\Omega_{ie}^e & -(\Omega_{ie}^e\Omega_{ie}^e - \Gamma^e)\\ 0 & I & 0\end{pmatrix}\begin{pmatrix}\phi^e\\ \delta\dot{x}^e\\ \delta x^e\end{pmatrix} + \begin{pmatrix}-C_b^e & 0 & 0\\ 0 & C_b^e & I\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\delta\omega_{ib}^b\\ \delta f^b\\ \delta g^e\end{pmatrix} \tag{3.21}$$

3.5 Error Dynamics in the e-frame on component form

Below are the explicit linear error dynamics of the navigation equations. Here we use the relaxed notation where the superscript e is neglected for the position ,

$$x^e = \begin{pmatrix}x_1 & x_2 & x_3\end{pmatrix}^T \tag{3.22}$$

and the equivalently for the error angles, the position error and velocity errors as

$$\phi^e = \begin{pmatrix}\phi_1 & \phi_2 & \phi_3\end{pmatrix}^T, \delta x^e = \begin{pmatrix}\delta x_1 & \delta x_2 & \delta x_3\end{pmatrix}^T, \delta\dot{x}^e = \begin{pmatrix}\delta\dot{x}_1 & \delta\dot{x}_2 & \delta\dot{x}_3\end{pmatrix}^T. \tag{3.23}$$

respectuively.

The earth rate is constant, giving $\dot{\Omega}_{ie}^e = \begin{pmatrix}0 & 0 & 0\end{pmatrix}^T$. Writing the gravitation vector in (2.19) $g^e(x^e)$ as $\begin{pmatrix}g_1 & g_2 & g_3\end{pmatrix}^T$ and taking the partial derivative leeds to

$$\Gamma^e = \frac{\partial g^e(x^e)}{\partial x^e} = \begin{pmatrix}\frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3}\\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3}\\ \frac{\partial g_3}{\partial x_1} & \frac{\partial g_3}{\partial x_2} & \frac{\partial g_3}{\partial x_3}\end{pmatrix} = GM\begin{pmatrix}\frac{-1}{R^3} + 3\frac{x_1^2}{R^5} & 3\frac{x_1x_2}{R^5} & 3\frac{x_1x_3}{R^5}\\ 3\frac{x_2x_1}{R^5} & \frac{-1}{R^3} + 3\frac{x_2^2}{R^5} & 3\frac{x_2x_3}{R^5}\\ 3\frac{x_3x_1}{R^5} & 3\frac{x_2x_3}{R^5} & \frac{2}{R^3} + 3\frac{x_3^2}{R^5}\end{pmatrix} \tag{3.24}$$

where $R = \sqrt{x_1^2 + x_2^2 + x_3^2}$, and for the case when navigating locally (small x_1 and x_2 components) then the terms containing R^{-5} can be neglected, and

$$\Gamma^e \approx \frac{GM}{R^3} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{pmatrix}. \tag{3.25}$$

The driving terms in the time derivatives of the angles and the translational velocities are defined as

$$\delta\omega_{ib}^b = \begin{pmatrix} \delta p & \delta q & \delta r \end{pmatrix}^T \tag{3.26}$$
$$\delta f^b = \begin{pmatrix} \delta f_1 & \delta f_2 & \delta f_3 \end{pmatrix}^T \tag{3.27}$$
$$\delta g^e = \begin{pmatrix} \delta g_1 & \delta g_2 & \delta g_3 \end{pmatrix}^T \tag{3.28}$$

and then transformed from b- to e-frame with the matrix C_b^e defined as

$$\begin{aligned} C_b^e \delta\omega_{ib}^b &= C_n^e C_b^n \delta\omega_{ib}^b \qquad (3.29) \\ &= \begin{pmatrix} -\sin L\cos\lambda & -\sin\lambda & -\cos L\cos\lambda \\ -\sin L\sin\lambda & \cos\lambda & -\cos L\sin\lambda \\ \cos L & 0 & -\sin L \end{pmatrix} \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \begin{pmatrix} \delta p \\ \delta q \\ \delta r \end{pmatrix} \\ &= \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix} \begin{pmatrix} \delta p \\ \delta q \\ \delta r \end{pmatrix} = \begin{pmatrix} d_{11}\delta p + d_{12}\delta q + d_{13}\delta r \\ d_{21}\delta p + d_{22}\delta q + d_{23}\delta r \\ d_{31}\delta p + d_{32}\delta q + d_{33}\delta r \end{pmatrix} \end{aligned}$$

where the following extra variables has been used

$$\begin{aligned} d_{11} &= -(\sin L\cos\lambda)\, c_{11} - (\sin\lambda)\, c_{21} - (\cos L\cos\lambda)\, c_{31} \qquad (3.30) \\ d_{12} &= -(\sin L\cos\lambda)\, c_{12} - (\sin\lambda)\, c_{22} - (\cos L\cos\lambda)\, c_{32} \\ d_{13} &= -(\sin L\cos\lambda)\, c_{13} - (\sin\lambda)\, c_{23} - (\cos L\cos\lambda)\, c_{33} \\ d_{21} &= -(\sin L\sin\lambda)\, c_{11} + (\cos\lambda)\, c_{21} - (\cos L\sin\lambda)\, c_{31} \\ d_{22} &= -(\sin L\sin\lambda)\, c_{12} + (\cos\lambda)\, c_{22} - (\cos L\sin\lambda)\, c_{32} \\ d_{23} &= -(\sin L\sin\lambda)\, c_{13} + (\cos\lambda)\, c_{23} - (\cos L\sin\lambda)\, c_{33} \\ d_{31} &= (\cos L)\, c_{11} - (\sin L)\, c_{31} \\ d_{32} &= (\cos L)\, c_{12} - (\sin L)\, c_{32} \\ d_{33} &= (\cos L)\, c_{13} - (\sin L)\, c_{33} \end{aligned}$$

and c_{ij} , $i = 1, 2, 3, j = 1, 2, 3$, are defined by the differential equations (2.41)-(2.43). Inserting the previously defined angular velocities and needed expressions: (3.23) - (3.30) into (3.20) leads to the following time derivatives of the error angles

$$\frac{d}{dt}\phi_1 = \omega_e\phi_2 - d_{11}\delta p + d_{12}\delta q + d_{13}\delta r, \tag{3.31}$$
$$\frac{d}{dt}\phi_2 = -\omega_e\phi_1 - d_{21}\delta p + d_{22}\delta q + d_{23}\delta r, \tag{3.32}$$
$$\frac{d}{dt}\phi_3 = -d_{31}\delta p + d_{32}\delta q + d_{33}\delta r, \tag{3.33}$$

and of the velocities

$$\begin{aligned}\frac{d}{dt}\delta\dot{x}_1 &= -f_3^e\phi_2 + f_2^e\phi_3 + 2\omega_e\delta\dot{x}_2 \\ &\quad + \left(\omega_e^2 - \frac{GM}{\left(x_1^2+x_2^2+x_3^2\right)^{3/2}}\right)\delta x_1 \\ &\quad + d_{11}\delta f_1 + d_{12}\delta f_2 + d_{13}\delta f_3 + \delta g_1,\end{aligned} \tag{3.34}$$

$$\begin{aligned}\frac{d}{dt}\delta\dot{x}_2 &= f_3^e\phi_1 - f_1^e\phi_3 - 2\omega_e\delta\dot{x}_1 \\ &\quad + \left(\omega_e^2 - \frac{GM}{\left(x_1^2+x_2^2+x_3^2\right)^{3/2}}\right)\delta x_2 \\ &\quad + d_{21}\delta f_1 + d_{22}\delta f_2 + d_{23}\delta f_3 + \delta g_2,\end{aligned} \tag{3.35}$$

$$\begin{aligned}\frac{d}{dt}\delta\dot{x}_3 &= -f_2^e\phi_1 + f_1^e\phi_2 + 2\frac{GM}{\left(x_1^2+x_2^2+x_3^2\right)^{3/2}}\delta x_3 \\ &\quad + d_{31}\delta f_1 + d_{32}\delta f_2 + d_{33}\delta f_3 + \delta g_3,\end{aligned} \tag{3.36}$$

and of the positions

$$\frac{d}{dt}\delta x_1 = \delta\dot{x}_1, \tag{3.37}$$

$$\frac{d}{dt}\delta_2 = \delta\dot{x}_2, \tag{3.38}$$

$$\frac{d}{dt}\delta x_3 = \delta\dot{x}_3, \tag{3.39}$$

3.6 Linear Error Dynamics in the n-frame

Now the objective is to formulate the error dynamics with respect to the geodetic coordinates $x^n = \begin{pmatrix} L & \lambda & h \end{pmatrix}^T$. The error state is then defined as

$$\bar{x} = \begin{pmatrix} \phi^n & \delta v^n & \delta x^n \end{pmatrix}^T. \tag{3.40}$$

Instead of specializing (3.16), the compact form of the navigation equations expressed in the n-frame (2.27) is perturbed. The resulting dynamics of the velocity error state, δv^n, can then be written as

$$\frac{d}{dt}\delta v^n = -\delta\left(\Omega_{in}^n + \Omega_{ie}^n\right)v^n - \left(\Omega_{in}^n + \Omega_{ie}^n\right)\delta v^n + \delta f^n + \bar{\Gamma}^n\delta x^n + \delta\bar{g}^n \tag{3.41}$$

where $\bar{\Gamma}^n = \frac{\partial\bar{g}^n}{\partial x^n}$.

The previously derived attitude errors (3.15) for the case of using the general a-frame follow the equation

$$\dot{\phi}^a = -C_b^a\delta\omega_{ib}^b - \omega_{ia}^a \times \phi^a + \delta\omega_{ia}^a.$$

and substituting n for all a results in the expression for the n-frame

$$\dot{\phi}^n = -C_b^n\delta\omega_{ib}^b - \omega_{in}^n \times \phi^n + \delta\omega_{in}^n.$$

The latitude (L), longitude (λ), and the height above the earth (h) are are used as state variables for the position and defined through the three first order differential equations (2.32) and also $v^n = \begin{pmatrix} v_N & v_E & v_D \end{pmatrix}^T$ are needed and defined by (2.21).

3.7 Error Dynamics in the n-frame on component form

Below the explicit linear error dynamics of the navigation equations in the navigation frame are derived. Here as also previously a relaxed notation is used, where the superscript n is neglected for the components of the angular error, the position error and the velocity error as

$$\begin{aligned} \phi^n &= \begin{pmatrix} \phi_N & \phi_E & \phi_D \end{pmatrix}^T, && (3.42)\\ \delta v^n &= \begin{pmatrix} \delta v_N & \delta v_E & \delta v_D \end{pmatrix}^T, && (3.43)\\ \delta x^n &= \begin{pmatrix} \delta L & \delta\lambda & \delta h \end{pmatrix}^T, && (3.44) \end{aligned}$$

where "δ" means, as before, perturbation from true values. The error state vector is then

$$\begin{aligned} \bar{x} &= \begin{pmatrix} \phi^n & \delta v^n & \delta x^n \end{pmatrix}^T && (3.45)\\ &= \begin{pmatrix} \phi_N & \phi_E & \phi_D & \delta v_N & \delta v_E & \delta v_D & \delta L & \delta\lambda & \delta h \end{pmatrix}^T. && (3.46) \end{aligned}$$

Starting with (3.41), the explicit expressions in component form are derived term by term. Taking the partial derivative of the angular rates (2.34) gives

$$\begin{aligned} -\ & \delta\left(\Omega^n_{en} + 2\Omega^n_{ie}\right) v^n = -\delta\left(\Omega^n_{in} + \Omega^n_{ie}\right) v^n && (3.47)\\ = -\ & \begin{pmatrix} \left(2\omega_e \cos L\delta L + \frac{\tan L\delta v_E}{R_E+h} + \frac{v_E\left(1+\tan^2 L\right)\delta L}{R_E+h} - \frac{v_E \tan L\delta h}{R_E+h}\right) v_E + \left(-\frac{\delta v_N}{R_N+h} + \frac{v_N\delta h}{(R_N+h)^2}\right) v_D \\ \left(-2\omega_e \cos L\delta L - \frac{\tan L\delta v_E}{R_E+h} - \frac{v_E\left(1+\tan^2 L\right)\delta L}{R_E+h} + \frac{v_E \tan L\delta h}{(R_E+h)^2}\right) v_N + \left(2\omega_e \sin L\delta L - \frac{\delta v_E}{R_E+h} - \frac{v_E\delta h}{(R_E+h)^2}\right) v_D \\ \left(\frac{\delta v_N}{R_N+h} - \frac{v_N\delta h}{(R_N+h)^2}\right) v_N + \left(-2\omega_e \sin L\delta L + \frac{\delta v_E}{R_E+h} - \frac{v_E\delta h}{(R_E+h)^2}\right) v_E \end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} -\left(\Omega^n_{en} + 2\Omega^n_{ie}\right)\delta v^n &= -\left(\Omega^n_{in} + \Omega^n_{ie}\right)\delta v^n = && (3.48)\\ &= -\begin{pmatrix} \left(2\omega_e \sin L + \frac{v_E \tan L}{R_E+h}\right)\delta v_E - \frac{v_N\delta v_D}{R_N+h} \\ \left(-2\omega_e \sin L - \frac{v_E \tan L}{R_E+h}\right)\delta v_N + \left(-2\omega_e \cos L - \frac{v_E}{R_E+h}\right)\delta v_D \\ \frac{v_N\delta v_N}{R_N+h} + \left(2\omega_e \cos L - \frac{v_E}{R_E+h}\right)\delta v_E \end{pmatrix}. \end{aligned}$$

The third termin in (2.34), δf^n, is derived by substituting all indeices a with n in (3.8) and substituting (3.26), (3.27) and (), as for the e-frame and $f^n = \begin{pmatrix} f_N & f_E & f_D \end{pmatrix}^T$, which results in

$$\begin{aligned} \delta f^n &= f^n \times \phi^n + C^n_b \delta f^b && (3.49)\\ &= \begin{pmatrix} -f_N\phi_E + f_E\phi_D \\ f_D\phi_N - f_N\phi_D \\ -f_E\phi_N + f_N\phi_E \end{pmatrix} + \begin{pmatrix} c_{11}\delta f_1 + c_{12}\delta f_2 + c_{13}\delta f_3 \\ c_{21}\delta f_1 + c_{22}\delta f_2 + c_{23}\delta f_3 \\ c_{31}\delta f_1 + c_{32}\delta f_2 + c_{33}\delta f_3 \end{pmatrix} \end{aligned}$$

where the last term $C^n_b\delta f^b$ is the driving term of the acceleration error dynamics originating from accelerometer errors. Next the term $\bar{\Gamma}^n\delta x^n$ is derived by taking the partial derivatives of the gravitation vector

$$\bar{g}^n\left(x^n\right) = GM\begin{pmatrix} 0 & 0 & \frac{1}{(R+h)^2} \end{pmatrix}^T \qquad (3.50)$$

with respect to $x^n = \begin{pmatrix} L & \lambda & h \end{pmatrix}^T$.

$$\bar{\Gamma}^n \delta x^n = \frac{\partial g^n}{\partial x^n}\delta x^n = \begin{pmatrix} \frac{\partial g_1}{\partial L} & \frac{\partial g_1}{\partial \lambda} & \frac{\partial g_1}{\partial h} \\ \frac{\partial g_2}{\partial L} & \frac{\partial g_2}{\partial \lambda} & \frac{\partial g_2}{\partial h} \\ \frac{\partial g_3}{\partial L} & \frac{\partial g_3}{\partial \lambda} & \frac{\partial g_3}{\partial h} \end{pmatrix} \delta x^n \tag{3.51}$$

$$= GM \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2\frac{1}{\left(1+\frac{h}{R}\right)^3 R^3} \end{pmatrix} \begin{pmatrix} \delta L \\ \delta\lambda \\ \delta h \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -2\frac{GM\delta h}{\left(1+\frac{h}{R}\right)^3 R^3} \end{pmatrix} \tag{3.52}$$

Finally the attitude error equations $\dot{\phi}^n = -C_b^n \delta\omega_{ib}^b - \omega_{in}^n \times \phi^n + \delta\omega_{in}^n$ are treated term by term starting with $-C_b^n \delta\omega_{ib}^b$

$$-C_b^n \delta\omega_{ib}^b = -\begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \begin{pmatrix} \delta p \\ \delta q \\ \delta r \end{pmatrix} = \begin{pmatrix} -c_{11}\delta p - c_{12}\delta q - c_{13}\delta r \\ -c_{21}\delta p - c_{22}\delta q - c_{23}\delta r \\ -c_{31}\delta p - c_{32}\delta q - c_{33}\delta r \end{pmatrix} \tag{3.53}$$

where the c_{ij}:s are calculated from (2.41-2.43) and $\delta\omega_{ib}^b = \begin{pmatrix} \delta p & \delta q & \delta r \end{pmatrix}^T$. Then ω_{ie}^n and ω_{en}^n in (2.33) are added together using (2.32) resulting in

$$\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n = \begin{pmatrix} \omega_e \cos L + \frac{v_E}{R+h} \\ -\frac{v_N}{R+h} \\ -\omega_e \sin L - \frac{v_E}{(R+h)} \tan L \end{pmatrix}. \tag{3.54}$$

and then

$$-\omega_{in}^n \times \phi^n = -\begin{pmatrix} \omega_e \cos L + \frac{v_E}{R+h} \\ -\frac{v_N}{R+h} \\ -\omega_e \sin L - \frac{v_E}{(R+h)} \tan L \end{pmatrix} \times \begin{pmatrix} \phi_N \\ \phi_E \\ \phi_D \end{pmatrix} \tag{3.55}$$

$$= -\begin{pmatrix} -\frac{v_N}{R+h}\phi_D - \left(-\omega_e \sin L - \frac{v_E}{R+h}\tan L\right)\phi_E \\ \left(-\omega_e \sin L - \frac{v_E}{R+h}\tan L\right)\phi_N - \left(\omega_e \cos L + \frac{v_E}{R+h}\right)\phi_D \\ \left(\omega_e \cos L + \frac{v_E}{R+h}\right)\phi_E + \frac{v_N}{R+h}\phi_N \end{pmatrix}$$

The last term $\delta\omega_{in}^n$ can be obtained by partial differentiation of ω_{in}^n in (3.54), with respect to $\bar{x}$ resulting in

$$\delta\omega_{in}^n = \begin{pmatrix} \frac{\delta v_E}{R+h} - \omega_e \sin L \delta L - \frac{v_E \delta h}{(R+h)^2} \\ -\frac{\delta v_N}{R+h} + \frac{v_N \delta h}{(R+h)^2} \\ -\frac{\tan L \delta v_E}{R+h} + \left(-\omega_e \cos L - \frac{v_E\left(1+\tan^2 L\right)}{R+h}\right)\delta L + \frac{v_E \tan L \delta h}{(R+h)^2} \end{pmatrix} \tag{3.56}$$

The final collection of the above terms are straight forward.

4. The studied problem

4.1 Navigations equations

The state equations can be written in the following way, where, if not otherwise stated, all vector components refer to the navigation system (the so called "n system"):

$$\dot{v}_N = f_N - 2\Omega v_E \sin L + \frac{v_N v_D}{R+h} - \frac{v_E^2 \tan L}{R+h}, \tag{4.1}$$

$$\dot{v}_E = f_E + 2\Omega \left(v_N \sin L + v_D \cos L\right) + v_E \frac{v_D + v_N \tan L}{R+h}, \tag{4.2}$$

$$\dot{v}_D = f_D - 2\Omega v_E \cos L - \frac{v_E^2}{R+h} - \frac{v_N^2}{R+h} + g, \tag{4.3}$$

$$\dot{L} = \frac{v_N}{R+h}, \tag{4.4}$$

$$\dot{\lambda} = \frac{v_E}{(R+h)\cos L}, \tag{4.5}$$

$$\dot{h} = -v_D. \tag{4.6}$$

$v_N, v_{E,}$ and v_D are the components of the vehicle's velocity vector relative to the earth, L and λ are the latitude and longitude, respectively, for the vehicle, h is its height over the earth's surface, Ω is the earth's angular speed, and R is the radius of the (spherical) earth.

The specific force components (f_N, f_E, f_D) in the n system are considered input signals.

No attitude is included in this system, i.e. the vehicle is considered as a point moving in space.

4.2 Error dynamics

4.2.1 Exact The driving terms in the above equations are the components of the specific force vectors, in the appropriate coordinate systems. In an Inertial Navigation System, the equations are fed with measurements of these quantities, which necessarily contain errors, which in turn give rise to erroneous results for the integrated velocities and positions. Symbolically, suppose that the navigation equations in an arbitrary coordinate system are

$$\frac{dx}{dt} = f\left(x, u\right), \tag{4.7}$$

where x contains the state vector components corresponding to the coordinate frame in question, and u is the vector of specific force components in the same frame. If (4.7) is driven by measurements, $\tilde{u}$, of the specific forces, instead of the true, error free ones, u, then the resulting state vector, $\tilde{x}$, follows the equation

$$\frac{d\tilde{x}}{dt} = f\left(\tilde{x}, \tilde{u}\right). \tag{4.8}$$

This is the Inertial Navigation System (INS), where, as opposed to the case for (4.7), the resulting state vector is known, because the input, $\tilde{u}$, consists of the known (although more or less erroneous) measurements of the angle velocities and specific forces. Note that the equations themselves, i.e. the function f, is considered to be correct for the INS. Define the errors of the specific force measurements as

$$\Delta u \hat{=} \tilde{u} - u, \tag{4.9}$$

and the error of the resulting INS state vector as

$$\Delta x \hat{=} \tilde{x} - x. \tag{4.10}$$

By subtracting (4.7) from (4.8), this gives the following equation for the INS error Δx :

$$\frac{d}{dt}\Delta x = f(\tilde{x}, \tilde{u}) - f(\tilde{x} - \Delta x, \tilde{u} - \Delta u), \tag{4.11}$$

where the only unknown component is the measurement error Δu.

4.2.2 Linearized If the errors, Δx and Δu, are small enough that the second term on the right hand side of (4.11) can be approximated to a first order Taylor expansion, then this equation can approximately be written

$$\frac{d}{dt}\Delta x = F(\tilde{x}, \tilde{u})\,\Delta x + G(\tilde{x}, \tilde{u})\,\Delta u, \tag{4.12}$$

where $F(\tilde{x}, \tilde{u})$ and $G(\tilde{x}, \tilde{u})$ are the Jacobians with respect to $\tilde{x}$ and $\tilde{u}$, respectively, of the system dynamics function $f(\tilde{x}, \tilde{u})$ (note that both $\tilde{x}$ and $\tilde{u}$ are known entities at each point in time).

5. Estimation methods

5.1 Kalman estimator

In order to compare the method of using nonlinear estimation methods applied to the full state equations (4.1)-(4.6) with the established method of estimating the INS error from an error model based on linearization, we will now recapitulate the Kalman estimation algorithm, both for the discrete and the continuous cases.

5.1.1 Discrete case. Consider the time discrete, linear, time variant, state space model

$$\begin{aligned} x_{k+1} &= F_k x_k + G_k w_k, && (5.1)\\ y_k &= H_k x_k + v_k, && (5.2) \end{aligned}$$

where k is the time index, x_k and y_k are state and measurement vectors, respectively. F_k, G_k, and H_k are time dependent matrices of appropriate dimensions. $w_k \sim N(0, Q_k)$ and $v_k \sim N(0, R_k)$ are white noise. Then it is an established fact, see for example [4], that the recursion

$$\begin{aligned} P_k &= P_{k|k-1} - K_k H_k P_{k|k-1}, && (5.3)\\ P_{k+1|k} &= F_k P_k F_k^T + G_k Q_k G_k^T, && (5.4)\\ K_{k+1} &= P_{k+1|k} H_{k+1}^T \left(H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}\right)^{-1}, && (5.5)\\ \hat{x}_{k+1} &= F_k \hat{x}_k + K_{k+1}\left[y_{k+1} - H_{k+1} F_k \hat{x}_k\right] && (5.6) \end{aligned}$$

minimizes the expected value of $(\hat{x}_k - x_k)^T (\hat{x}_k - x_k)$, i.e. the expected sum of the quadratic estimation errors in all state variables. The matrices P_k and $P_{k|k-1}$ are the covariance matrices for the estimation error at time k with and without the measurement y_k at that time included, respectively. The algorithm described in (5.3) – (5.6) happens to be linear in structure, but is nevertheless optimal (in the given sense) among all possible estimators, linear or nonlinear (if the noises w_k and v_k are not Gaussian, the algorithm is still the optimal one in the class of linear estimators).

5.1.2 Continuous case. In the continuous case, we consider the model

$$\begin{aligned} \dot{x}(t) &= F(t)\, x(t) + G(t)\, w(t), && (5.7)\\ y(t) &= H(t) x(t) + v(t). && (5.8) \end{aligned}$$

Here, t is the time, x and y are the state and measurement vectors. F, G, and H are the system matrices. w and v are continuous white (Gaussian) noise with (possibly time dependent) spectral density matrices Q and R. In this case, the optimal algorithm consists of the two differential equations

$$\begin{aligned} \dot{P} &= FP + PF^T + GQG^T - PH^T R^{-1} HP, && (5.9)\\ \frac{d\hat{x}}{dt} &= F\hat{x} + PH^T R^{-1}\left[y - H\hat{x}\right]. && (5.10) \end{aligned}$$

5.1.3 Kalman Estimation of INS errors The Inertial Navigation System is the set of equations (4.1) – (4.6), or, in compact symbolic form, (4.8). The output from this system differs from the true state as given by the "true system", (4.7), because of the measurement errors Δu, contained in the measurement vector $\tilde{u}$ as defined in (4.9). One possibility to reduce this error is to introduce so called "external measurements", i.e. measurements of the same position and/or velocity coordinates that the INS system tries to calculate. The INS error, Δx, defined in (4.10), can then be estimated by means of Kalman estimation in the following way: the state vector x is measured by means of an external device, that is independent of the INS. This measurement and the measurement error are called x_{ext} and Δx_{ext}, respectively, so we have $x_{ext} = x + \Delta x_{ext}$. According to the definition of the INS error, (4.10), we also have $\tilde{x} = x + \Delta x$. Now the difference Δy between these two known entities is the same as the difference between the corresponding errors, as $\Delta y \hat{=} \tilde{x} - x_{ext} = \Delta x - \Delta x_{ext}$. According to (4.12), if the error Δx is small, it follows the dynamics

$$\dot{\Delta x} = F(\tilde{x}, \tilde{u})\,\Delta x + G(\tilde{x}, \tilde{u})\,\Delta u, \tag{5.11}$$

Now we can regard the equation

$$\Delta y = \Delta x - \Delta x_{ext} \tag{5.12}$$

as a measurement equation belonging to the state equation (5.11). The measurement is a direct observation of the INS error Δx with an error added (or rather, subtracted), that is exactly the external measurement error, for which we know the error covariance $R_{ext} \hat{=} E\left[\Delta x_{ext} \Delta x_{ext}^T\right]$. Therefore, an optimal estimator for Δx is the equations (5.9), (5.10) with $\hat{x}$ replaced by $\Delta\hat{x}$, y by Δy, F by $F(\tilde{x}, \tilde{u})$, G by $G(\tilde{x}, \tilde{u})$, H by the identity matrix I, R by R_{ext}, and Q by Q_{INS}.

5.2 Particle Estimator (PE)

The Kalman Estimator, in the application described above, is an unrivalled tool for using external measurements in estimating the errors of the INS values, as long as these values are "small". When the errors grow in magnitude, however, the underlying error model according to (5.11) will no longer be valid, so we will have a good estimate of a bad model, which is not satisfactory.

A remedy is to go back to the original, nonlinear, equations (4.7), used together with the external measurements of the velocity and/or position coordinates, and apply some kind of nonlinear estimator. One such estimator is the Extended Kalman Filter (EKF), which can be regarded as a generalization of the linear Kalman Estimator to the nonlinear case. However, the EKF is an ad hoc method, for which no general theoretical convergence results exist, so we have no guarantee that it will work (even if it does often enough). Therefore we will try to use a completely different method, based on realizations of discrete random processes, called Particle Estimation.

A complete mathematical derivation is presented in [13]. The idea is to draw a number of points in the state space ("particles") randomly, according to a given initial distribution. For each time step, all the particles are propagated according to (4.8) and (4.9), with noise terms Δu individually drawn from the distribution of the INS measurement errors, and then exclude particles that have too large external measurement residues, so that the particles that are most consistent with the external measurements will be retained. The minimum variance estimate of the state will at each time be the arithmetic mean of all the particles' positions in the state space.

5.2.1 Description A theoretical description is given in [13]. Below follows a brief algorithmic description. x, u are true state values and INS measurements (i.e. specific forces), respectively, that follow (4.7): $\dot{x} = f(x, u)$.

$\tilde{x}, \tilde{u}$ are the corresponding, noisy, INS entities, thus following (4.8): $\dot{\tilde{x}} = f(\tilde{x}, \tilde{u})$. Define the INS state error and the INS measurement error according to (4.10): $\Delta x \triangleq \tilde{x} - x$, and (4.9): $\Delta u \triangleq \tilde{u} - u$, respectively. $\{\ \Delta u(t) : t \geq 0\}$ is supposed to be a stationary random process with probability density function (pdf) $\frac{d}{dm}P(\Delta u \leq m) = p_{\Delta u}(m)$ (not necessarily Gaussian!).

We are going to work with the system at discrete points in time, and we suppose that the time step h is long enough for the time discrete process $\{\Delta u(kh) : k = 0, 1, 2, ...\}$ to be uncorrelated.

For the true state, x, we have $\dot{x} = f(x, u) = f(x, \tilde{u} - \Delta u)$, which describes a random process where $\tilde{u}(t)$ is a known function of time, and $\Delta u\,(t)$ is process noise.

External measurements $\{\tilde{y}_k :\ k = 0, 1, 2, ...\}$ are taken at discrete points in time, according to $\tilde{y}_k = h(x_k) + \mu_k$, where $\{\mu_k : k = 0, 1, 2, ...\}$ is an uncorrelated random process with pdf $\frac{d}{dm}P(\mu_k \leq m) = p_{\mu_k}(m)$, and where $x_k \triangleq x(kh)$.
The algorithm is described as follows::

Algorithm 1 *First, N particles $x_0^i, i = 1, ..., N$ are drawn from a given initial distribution.*
Then, the following is done for $k = 0, 1, 2, ...$:
Measurement update:

- $\tilde{y}_k := h(x_k) + \mu_k$,
- $w_i := p(\tilde{y}_k \,|x_k = x_k^i) = p_{\mu_k}(\tilde{y}_k - h(x_k^i))$ *for* $1 \leq i \leq N$,
- $w_i := w_i / \sum_{j=1}^{N} w_j$ *for* $1 \leq i \leq N$ *(resampling probabilities)* ,
- x_k^{i*} *is drawn from a distribution with pdf* $p_{x_k^{i*}}(x) = \sum_{j=1}^{N} w_j \delta\left(x - x_k^j\right)$ *for* $1 \leq i \leq N$ *(posterior values).*

Prediction step:

- Δu_k^i *is drawn from a distribution with pdf* $p_{\Delta u}(\cdot)$ *for* $1 \leq i \leq N$,
- $\dot{x}^i(\tau) = f(x^i(\tau), \tilde{u}(\tau) - \Delta u_k^i)$ *is integrated with* τ *from* kh *to* $(k+1)h$ *with initial value* $x^i(kh) = x_k^{i*}$ *for* $1 \leq i \leq N$.
- $x_{k+1}^i \triangleq x^i((k+1)h)$ *for* $1 \leq i \leq N$.
- *Finally, the minimum variance estimate is calculated as* $\hat{x}(kh) \triangleq \frac{1}{N}\sum_{i=1}^{N} x_k^{i*}$ *(posterior estimate).*

5.2.2 Observations We make the following observations and comments regarding the algorithm described above.

- If the particles at time kh end up in the same point after the prediction step, i.e. $x_k^j = x_k^i, \forall i, j$, the measurement y_k will have no effect, because the resampling never produces any new points, but only removes or copies old ones.
- Furthermore, if the process noise after this point in time remains zero ($\Delta u_l \equiv 0$, $l > k$), all particles will also in the future end up in the same point , i.e. $x_l^j = x_l^i, \forall i, j$ for $l \geqslant k$, as the N prediction equations in that case are identical and have the same initial value ($\dot{x}^i(\tau) = f(x^i(\tau), \tilde{u}(\tau))$, $kh \leq \tau \leq (k+1)h$ with initial value $x^i(kh) = x_k^{i*}$, which thus is the same for all i). Therefore, the estimator will in this case and from this point in time (kh) simply be driven by the same signal $\{\tilde{u}(\tau) : \tau \geqslant kh\}$ as the INS is driven by. In particular, if all

the particles right from the beginning, i.e. at the time $kh = 0$, are initialized with the same value as the initial value for the INS, then the trajectories for the estimator will coincide with the one for the INS. This will of course be the case irrespective of the number of particles, N. Also, in this case not even the distribution, $\left\{p_{\mu_l}(\cdot) : l \geq k\right\}$, including the accuracy of the external measurement errors, is of any significance whatsoever.

- If the particles at some point in time, e.g. right from the start, have some spreading, and the process noise, as in the description above, is =0 after this time, then the number of points that the particles occupy will (statistically) decrease at each time step, because the resampling reduces the number of points, and the prediction will retain this number, as the prediction step will be completely deterministic. Sooner or later, the particles will assemble in the same point, and according to the reason described above, we will get an "estimator" that does not use its measurements at all! A conclusion is that the process noise, Δu, is necessary for the external measurements to have any significance to the estimates.

- Compare with the Kalman Estimator:

$$\dot{\hat{x}} = F\ \hat{x} + PH^T R^{-1}\left[y - H\ \hat{x}\right],$$
$$\dot{P} = FP + PF^T + GQG^T - PH^T R^{-1} HP.$$

  Here, the particle distribution corresponds to the P matrix. The particles occupying the same point corresponds to P being =0, which in turn implies that the external measurements have no impact on the estimate. If there is no process noise, i.e. $Q = 0$, P will remain =0, and the external measurements will have no effect on the estimate, just as for the particle estimator. Even if $P \neq 0$, P will converge to 0, (if F is stable). Therefore, the measurements will become insignificant, just as for the particle estimator.

6. Simulation Results

The navigation equations in navigation frame as described in (4.1) - (4.6) have been implemented in a SIMULINK model, an overview of which is described and shown in figure 7.

6.1 Simulation model description

The equations are simulated in two identical, green, blocks. One of these is called "Reality", and the other "INS". The first one of these ("Reality") is fed with exact (error free) measurements of specific forces as inputs, so that its output will be the exact state variables for the true trajectory.

The second block ("INS") is fed with measurements with errors included, thus simulating an Integrated Navigation System.

These measurement errors also feed a linear error model ("nav_error2slw6"), that can calculate the INS errors, provided they are "small". These errors are of course unknown in a real system, so the error calculations are only done for checking purposes.

The same linear model is used in a Kalman Estimator (KE) that estimates the INS error using the method described in subsection 5.1.3. Exact values of velocity and position components are thus taken from the output of "Reality", and measurement errors are added from the block "GPS error". These simulated external measurements (also called "GPS measurements") are subtracted from the corresponding INS values, and the result is used as input to the estimator, which is implemented in continuous form.

The KE is split into two parts: "errTNK_innov6", that calculates the innovation term $PR_{ext}^{-1}\left[\Delta y - \Delta\hat{x}\right]$, and "errTNuppdat6", that adds the innovation to $F\Delta\hat{x}$ in order to get the time derivative of the estimate (see (5.10) with $\hat{x}$ replaced by $\Delta\hat{x}$, H with I, and R with R_{ext}). All inputsignals to the block "errTNK_innov6" are described in table (7.1).

6.2 Trajectory

The trajectory used in the simulations is shown in figures 6.1 – 6.3. It is nearly flat and starts 5 km east of the Greenwhich meridian in north direction and moves in a square with rounded corners. The speed is constant =1000 m/s.

6.3 Generation of Inertial Navigation System measurements

The specific forces f_N, f_E, and f_D to be used in the simulations are calculated from (4.1) – (4.3), with all other variables known as functions of time.

The measurement errors, added to these calculated, exact, values, are of two kinds: Gaussian uncorrelated with zero mean ("white noise") and unknown-but-constant ("bias").

The Gaussian uncorrelated noise process with zero mean (or "white noise") is impossible to realize exactly in a continuous simulation (and even more so in reality), because it has infinite bandwidth, and hence infinite power. However, when input

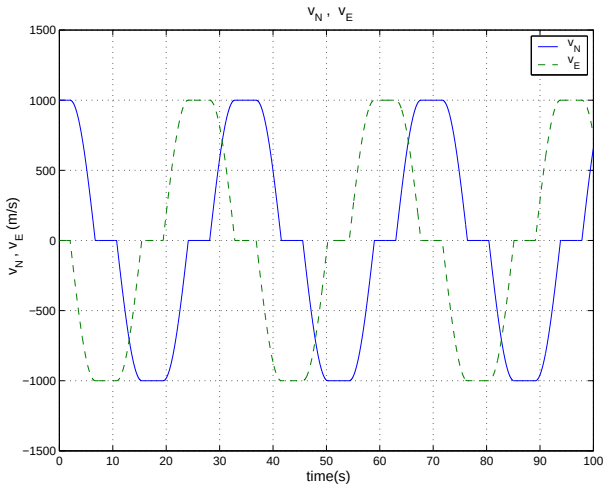


Figure 6.1: Velocity components v_N and v_E for the trajectory

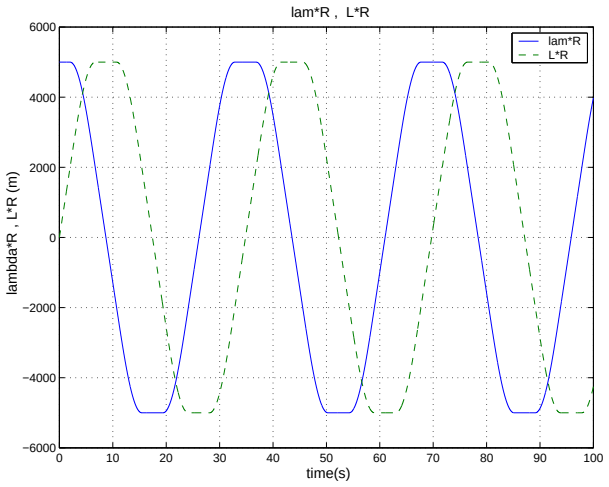


Figure 6.2: Position components λR and LR for the trajectory

to a system with limited bandwidth, the high frequencies in its power spectrum will be damped out and thus have no effect on the output. Therefore, white continuous noise can always be approximated with a discrete sequence of piecewise constant, independent, random values, where the discretization step and the variance of the discrete random values are chosen in such a way that the result on the output will be the same as it would have been, had the white noise been continuous. For a more detailed explanation, see Appendix A.

If the white noise is the only acceleration measurement error, the KE will give a mean square optimal estimate of the INS error. If the acceleration measurement error also contains a constant ("bias"), it can be shown that the Kalman estimate will contain a static error, the size of which however can be reduced by increasing the elements of the process covariance matrix Q_{INS}. The price for this reduction is an increased sensitivity to errors in the external measurements, as the time constant for the error decreases (a remedy would be to introduce the biases as new states, $b = (b_1, b_2, b_3)$, with state equation $\dot{b} = 0$, but this is not done here).

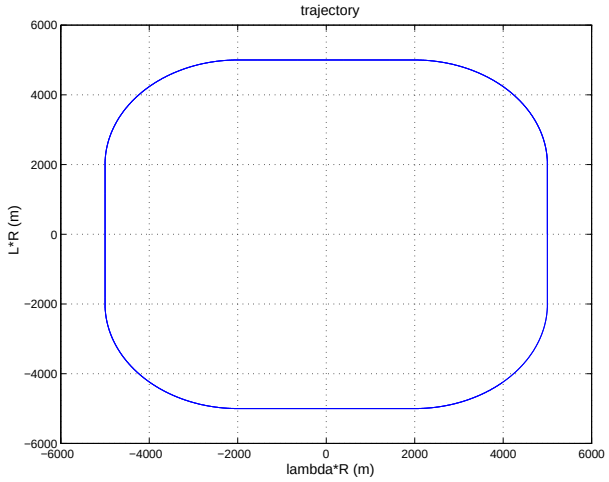


Figure 6.3: Trajectory (LR plotted against λR).

6.4 Results

6.4.1 Comments to plots The upper plot in figure 6.4(a) shows v_N as a function of time in a simulation with no bias in the INS measurements ($b_1 = b_2 = b_3 = 0$) and optimal covariance matrix Q_{INS} in the KE. Five different calculations of v_N are shown: the true v_N, v_N calculated by the Inertial Navigation System (INS), the v_N estimated by the Particle Estimator (PE) by means of the external measurements, and the Kalman estimated v_N, also by means of the external measurements.

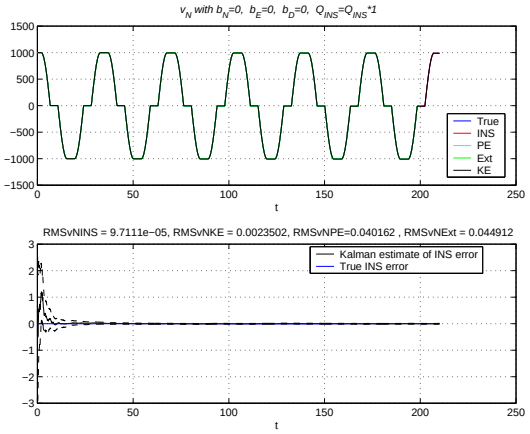
Because of the absence of acceleration measurement bias in this case, all these plots overlap in practice, so they look like one single curve. The lower plot of figure 6.4(a) shows the Kalman estimate of the INS error of the same quantity (v_N). Here is also shown the 3σ-limits for the estimation error.

In figure 6.4(b) these curves are shown magnified in a small time interval. Here we can see that the Kalman estimate is close to zero. The estimated value is still inside the 3σ- limits. The same conclusions can be drawn from the figures 6.5 and 6.6 regarding v_E and v_D, respectively.

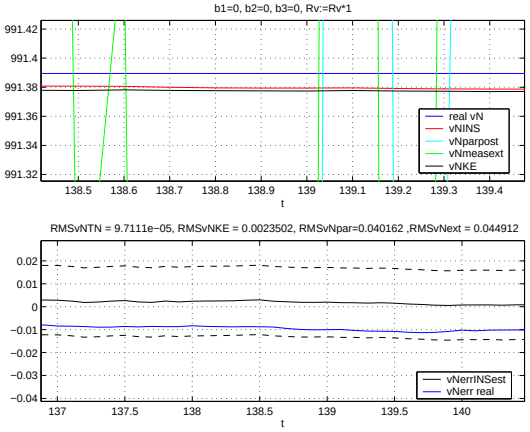
In figures 6.18(a) – 6.20(b), error plots are shown for v_N, v_E, and v_D, respectively, for the same specific force measurement case, i.e. without bias. The upper plots of the (a) parts show four different (true) errors, namely the INS error, the PE error, the external measurements error, and the KE error. The lower plots of the (a) parts show the KE innovation terms for the measurements of the velocity components in question, together with the corresponding 3σ-limits (the dashed straight lines). As there is no bias in the process noise to the KE in this case, these innovations are inside the 3σ-limits most of the time. The (b) parts of these three figures show magnifications of certain chosen time interval of the (a) parts.

In figure (6.7(a)) v_N is shown for the case where the measurement of f_N contains a bias 1 m/s^2. Here it is obvious that the INS value is drifting away from the true value. Furthermore, the Kalman estimate of the v_N error shows a constant bias of approximately 40 m/s^2, whereas the particle estimate still is bouncing around near the true value, which can be seen from the magnification in figure 6.7(b).

As a comparison, look at the figures 6.8(a) and 6.8(b). Here, the simulation duration is only 200s, as compared to 1000s in the previous case. The main change, compared to the earlier case, is, however, that the variances in the process noise matrix, Q_{INS}, is now made 10 times greater in the KE. This gives a smaller Kalman estimate bias ($\approx$ 25m/s). This bias is even smaller ($\approx$ 15m/s) in the case with 100 times greater Q_{INS}-variances, as is shown in figures 6.9(a) and 6.9(b).



(a)



(b)

Figure 6.4: v_N without bias in specific force measurements.

The corresponding results for the v_E component are shown in figures 6.10(a), 6.10(b), 6.11(a), 6.11(b), 6.12(a), and 6.12(b). Here, the Kalman estimate bias is considerably smaller, 0.2m/s with nominal Q_{INS}, and practically zero when Q_{INS} is multiplied by 10 or 100.

The results for v_D, seen in figures 6.13(a)-6.15(b), show that the v_D Kalman estimate error practically disappears when Q_{INS} is multiplied by 10 or 100.

The error plots corresponding to figures 6.7(a)-6.15(b) are shown in figures 6.21(a)-6.29(b). Here, we see biases in estimates and innovations for the v_N (which has a bias in the corresponding specefic force measurement), but not for the other two components (v_E and v_D).

Next, we investigated what happens if, in addition to the bias in the f_N measurement, we also introduce biases in the f_E and f_D measurements. First, it was observed that no change whatsoever occurred in the behavior of the estimates of v_N. Therefore, no plots are shown for this case. For v_E, however, the figures 6.16(a), 6.16(b), and 6.16(c) show that an increase of the Kalman Estimator Q_{INS} matrix elements reduces the estimate bias considerably.

The same thing happens for v_D, which is shown in figures 6.17(a), 6.17(b), and

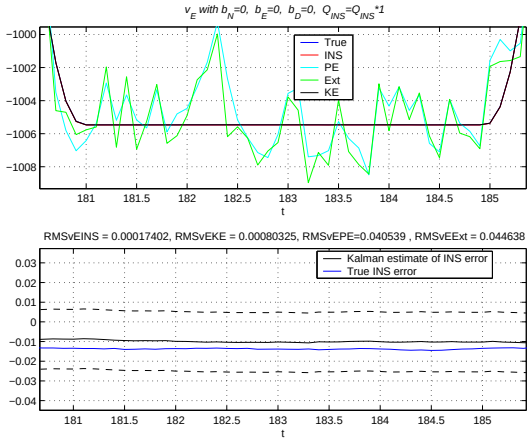


Figure 6.5: v_E without bias in specific force measurements (magnification)

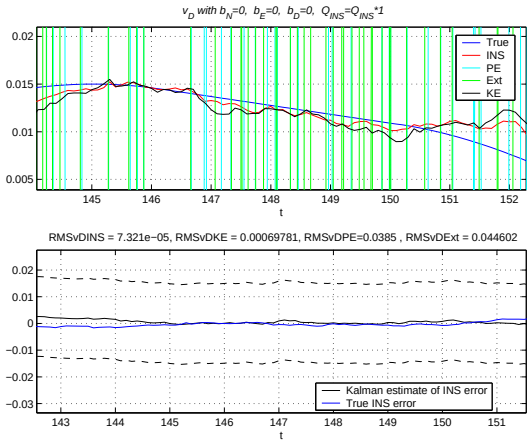
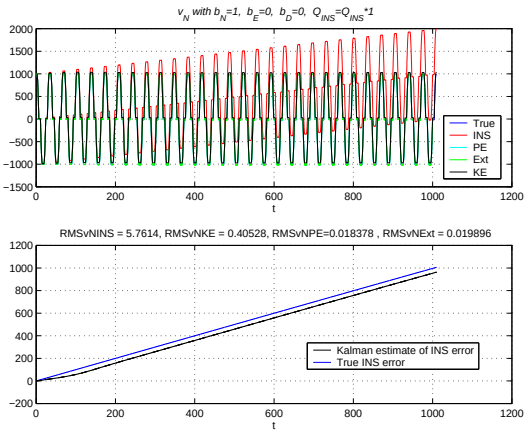


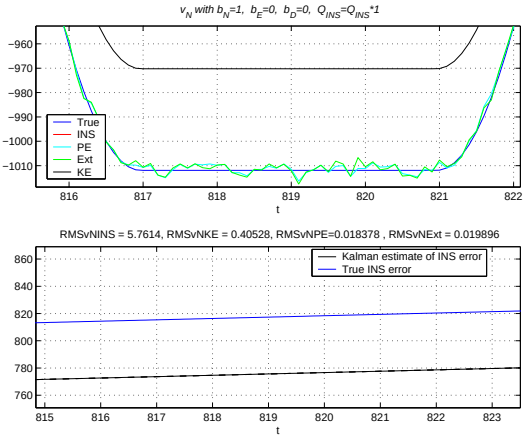
Figure 6.6: v_D without bias in specific force measurements (magnification)

6.17(c).

The error plots for the case with bias in all specific force measurements, finally, are shown in figures 6.30(a)-6.32(c).

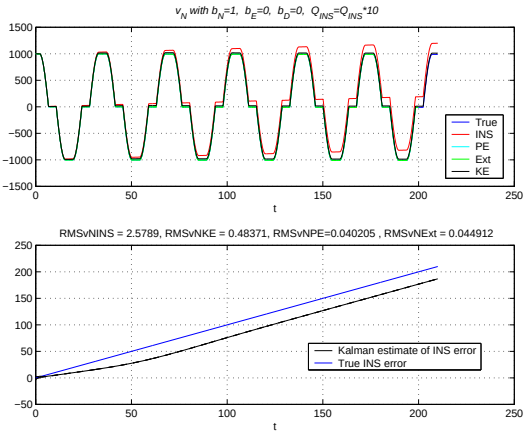


(a)

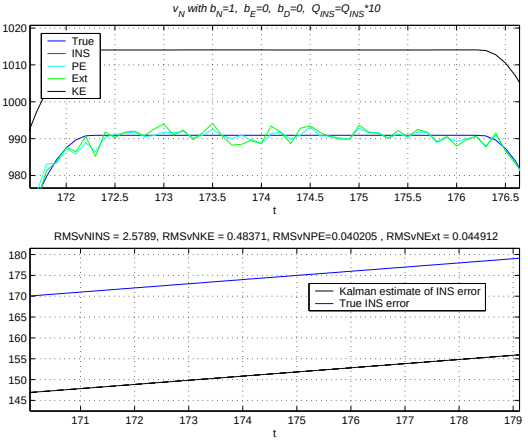


(b)

Figure 6.7: v_N with bias in f_N-measurement.

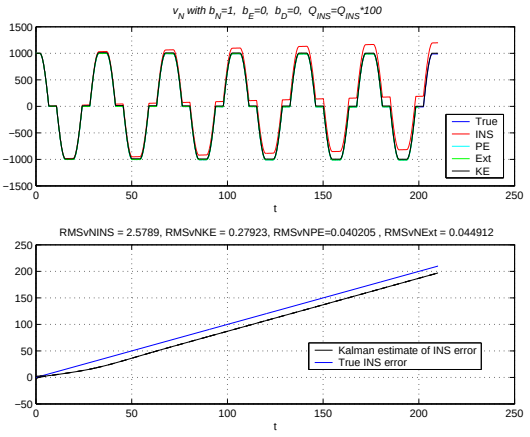


(a)

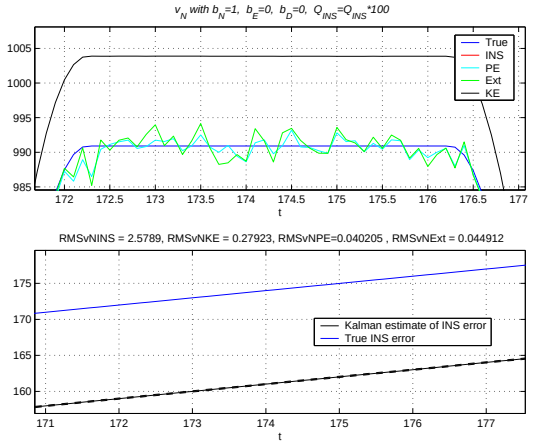


(b)

Figure 6.8: v_N with bias in f_N-measurement and Q_{INS} multiplied by 10.

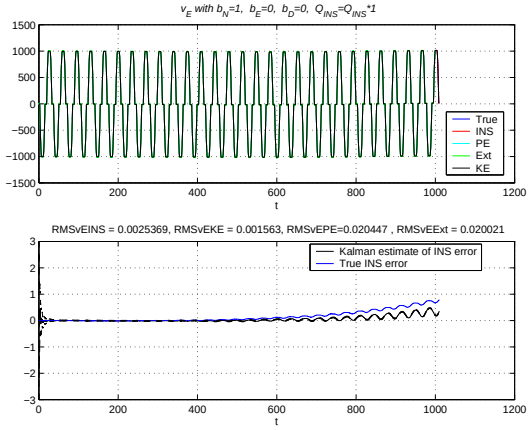


(a)

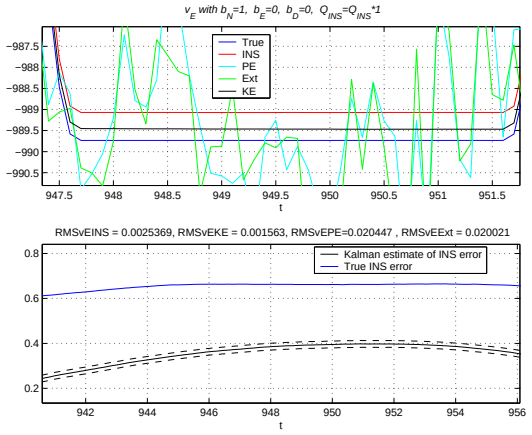


(b)

Figure 6.9: v_N with bias in f_N-measurement and Q_{INS} multiplied by 100.

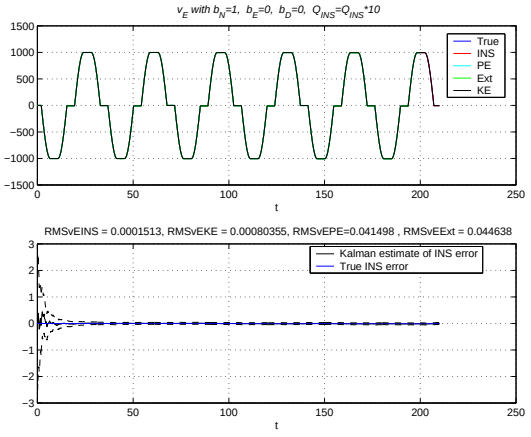


(a)

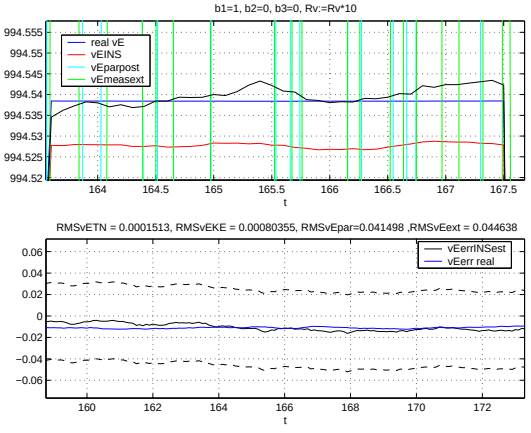


(b)

Figure 6.10: v_E with bias in f_N-measurement.

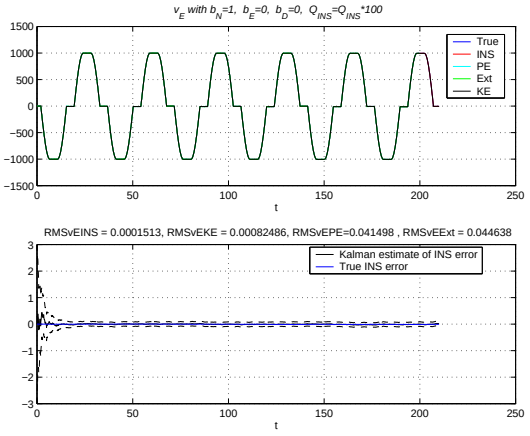


(a)

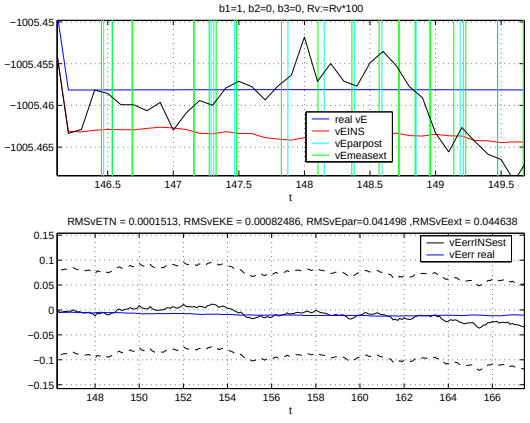


(b)

Figure 6.11: v_E with bias in f_N-measurement and Q_{INS} multiplied by 10.

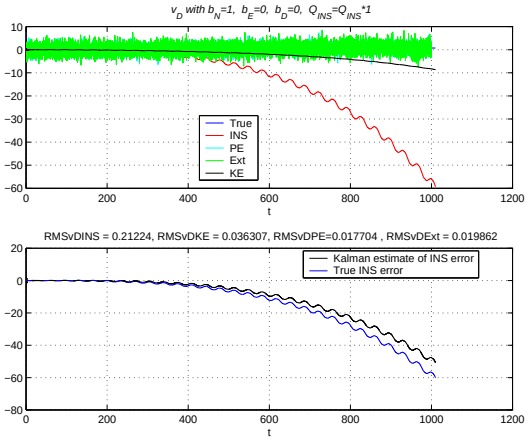


(a)

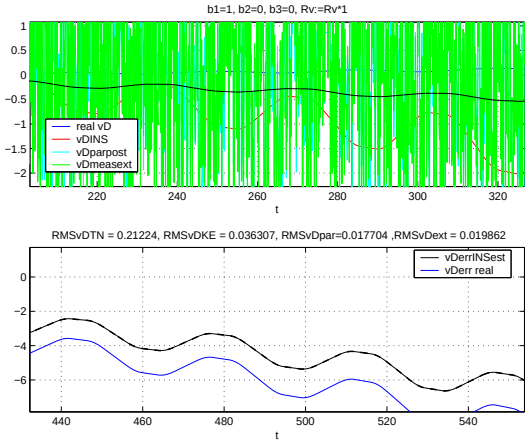


(b)

Figure 6.12: v_E with bias in f_N-measurement and Q_{INS} multiplied by 100.

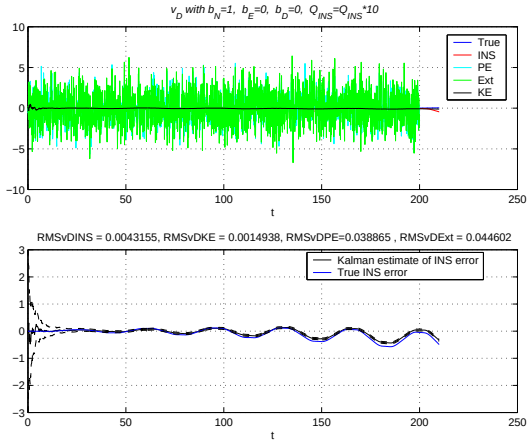


(a) $Q_{INS} * 1$

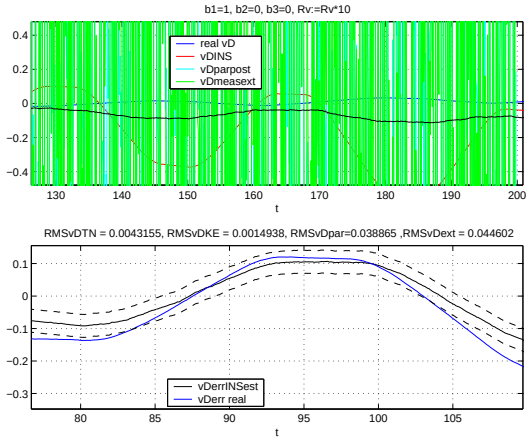


(b) $Q_{INS} * 1$, magnified

Figure 6.13: v_D with bias in f_N-measurement.

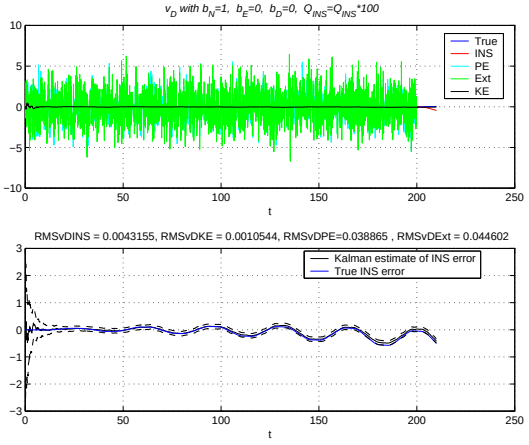


(a) $Q_{INS} * 10$

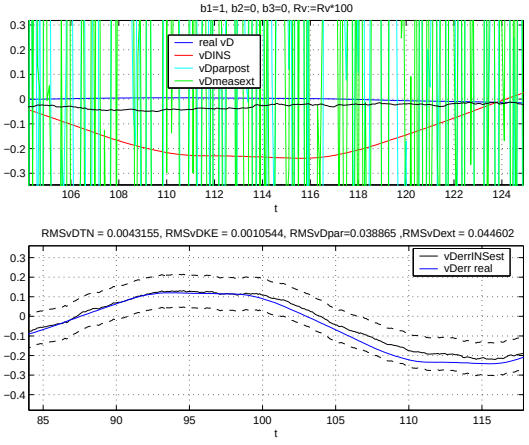


(b) $Q_{INS} * 10$, magnified

Figure 6.14: v_D with bias in f_N-measurement and Q_{INS} multiplied by 10.

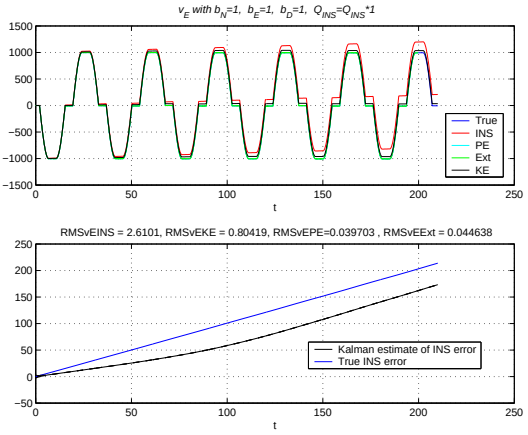


(a) $Q_{INS} * 100$

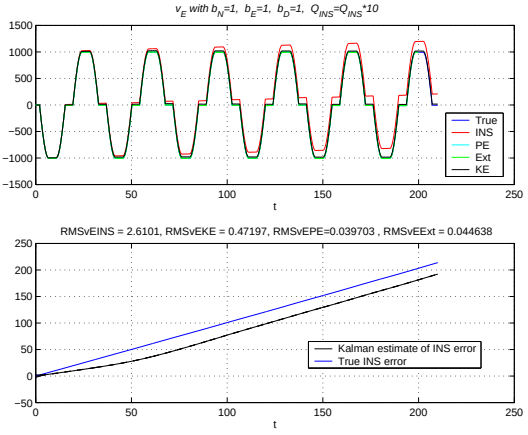


(b) $Q_{INS} * 100$, magnified

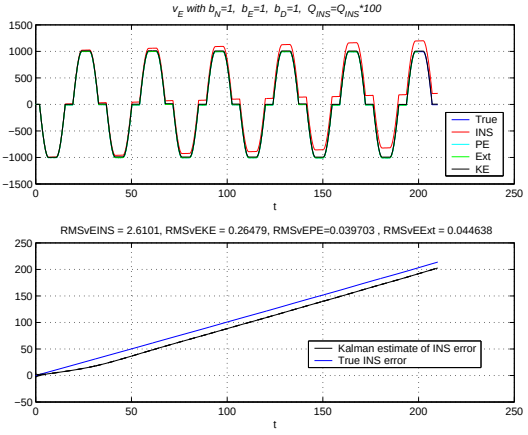
Figure 6.15: v_D with bias in f_N-measurement and Q_{INS} multiplied by 100.



(a) $Q_{INS} * 1$

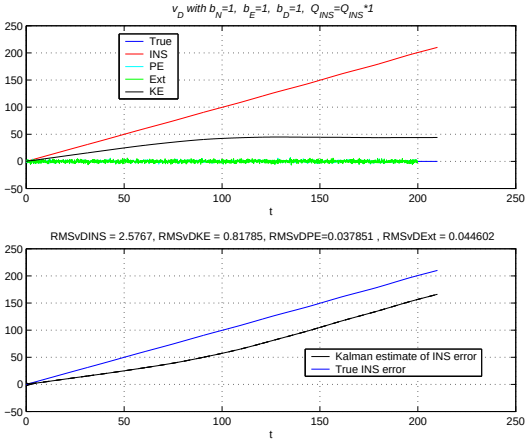


(b) $Q_{INS} * 10$

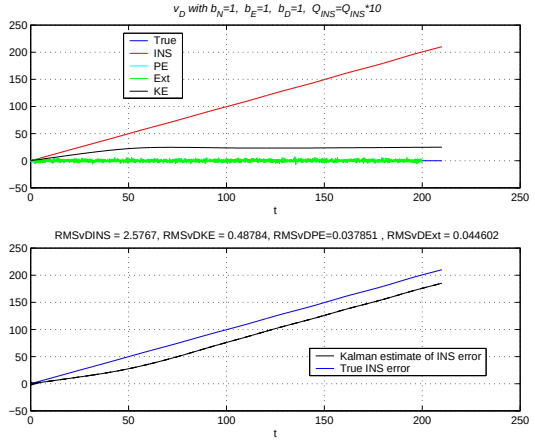


(c) $Q_{INS} * 100$

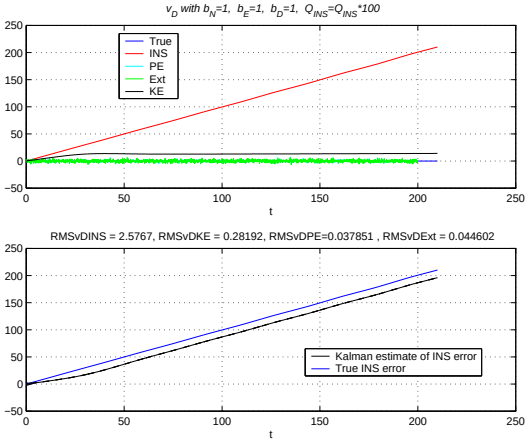
Figure 6.16: v_E with bias in f_N-, f_E-, and f_D-measurements.



(a) $Q_{INS} * 1$

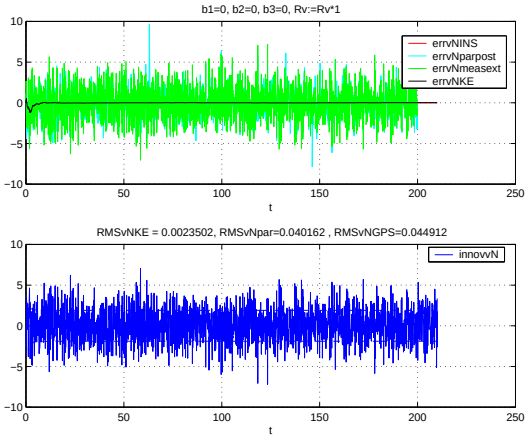


(b) $Q_{INS} * 10$

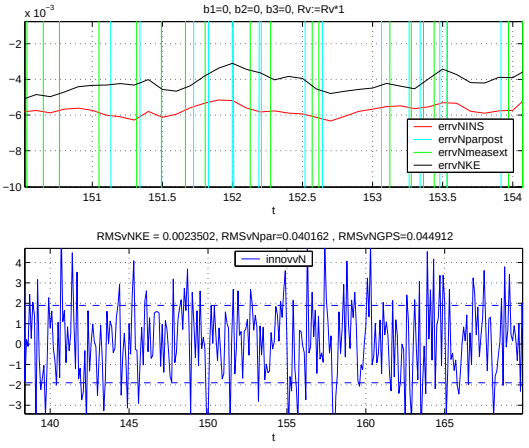


(c) $Q_{INS} * 100$

Figure 6.17: v_D with bias in f_N-, f_E-, and f_D-measurements.

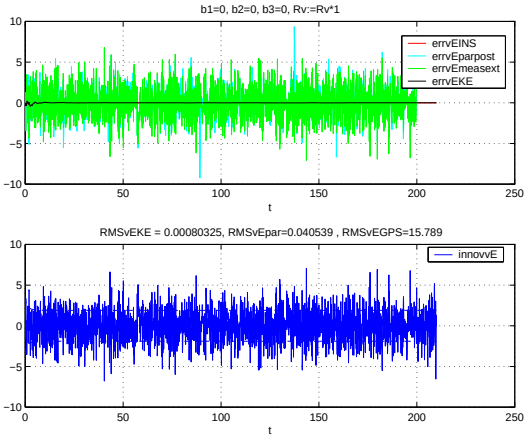


(a)

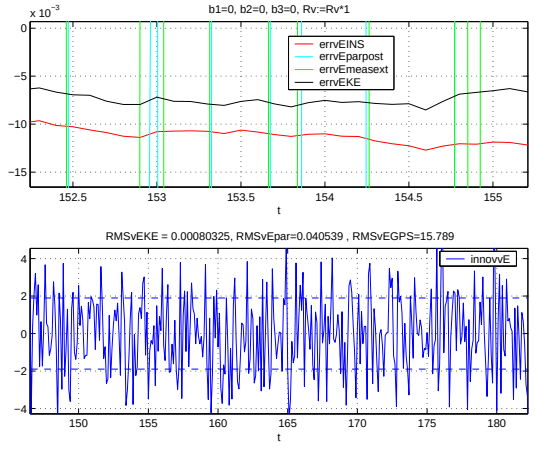


(b) magnified

Figure 6.18: v_N error without bias in specific force measurements.

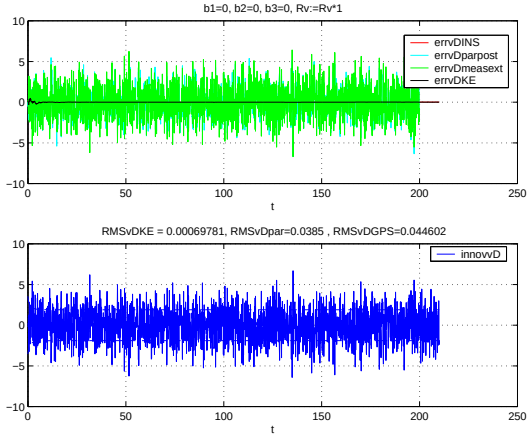


(a)

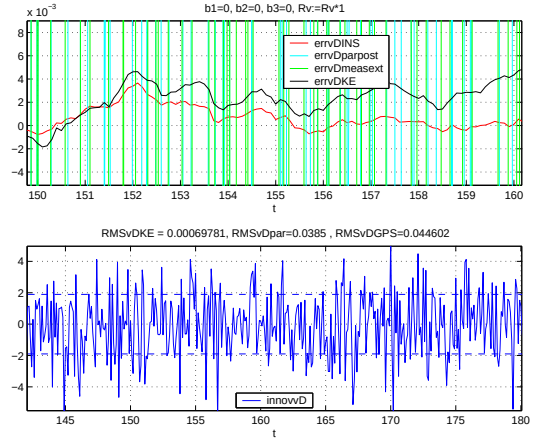


(b) magnified

Figure 6.19: v_E error without bias in specific force measurements.

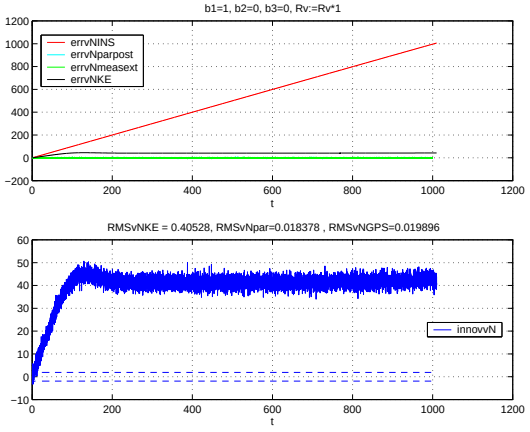


(a)

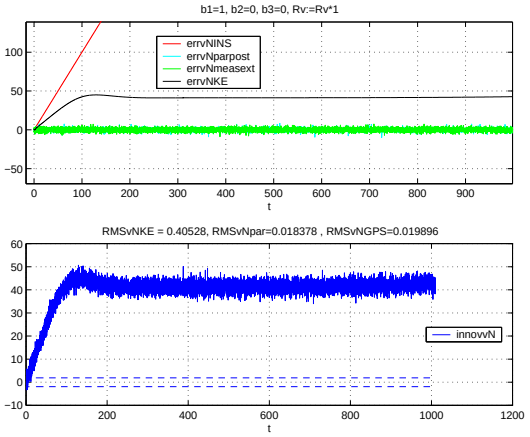


(b) magnified

Figure 6.20: v_E error without bias in specific force measurements.

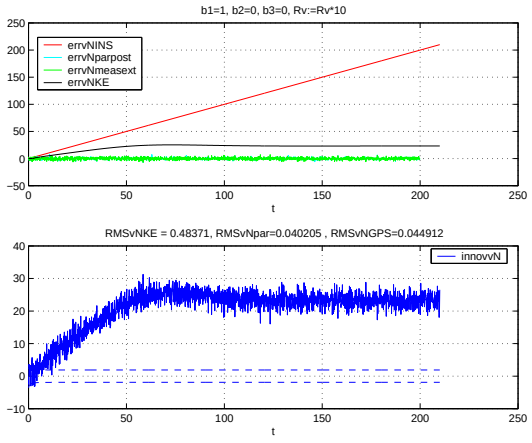


(a)

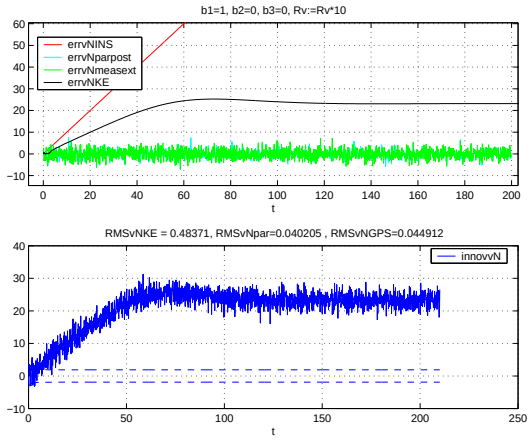


(b) magnified

Figure 6.21: v_N error with bias in f_N-measurement.

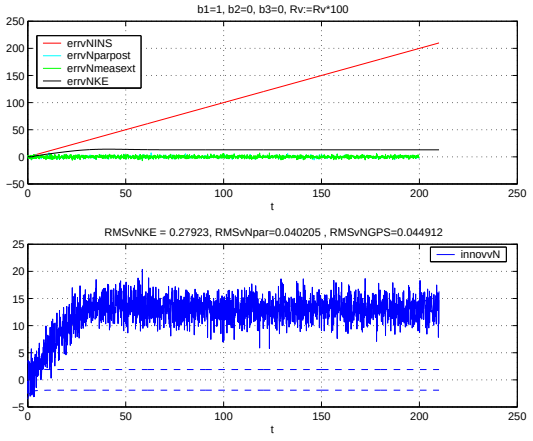


(a)

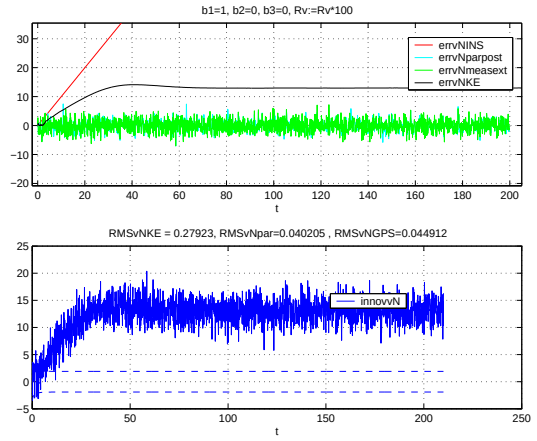


(b) magnified

Figure 6.22: v_N error with bias in f_N-measurement and Q_{INS} multiplied by 10.



(a)



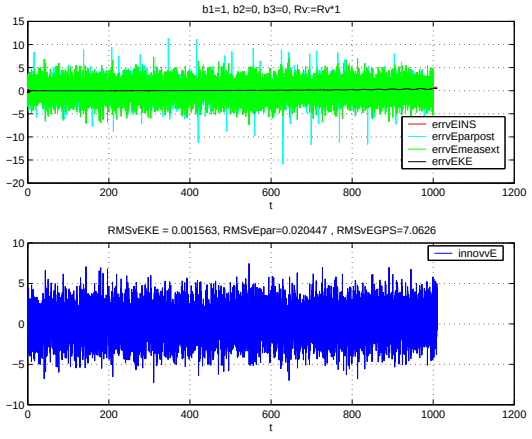
(b) magnified

Figure 6.23: v_N error with bias in f_N-measurement and Q_{INS} multiplied by 100.

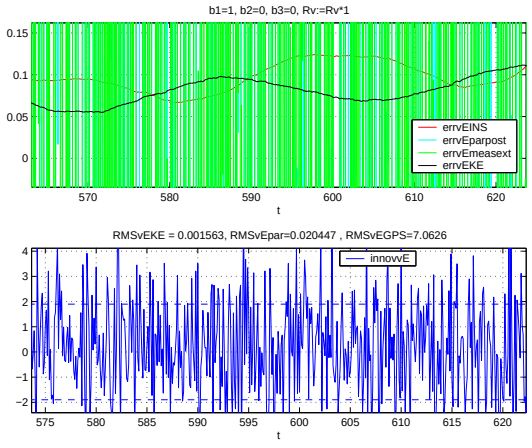
6.4.2 RMS values In tables 6.1, 6.2, and 6.3 RMS values can be compared in three different cases of specific force measurement bias: no bias, bias in f_N, and bias in f_N, f_E, and f_D. All the biases are 1.0 m/s^2. The "estimation" methods compared in these tables are: INS, PE with Q_{INS} multiplied with 1, 10, and 100, PE, and the raw external measurements.

Without any bias, the KE is optimal and gives far better result than the PE. INS tends to be still better in this realization, which can be explained by the quality of the external measurements. Bias in f_N affects of course the INS value for v_N to a very high degree, but also the KE estimates, although in lesser extent with increasing Q_{INS}. It can also be seen that the f_N bias affects the KE estimates of v_E and v_D somewhat.

With bias in all three specific force measurements, the KE estimates of all three velocity components seem to be affected by the same amount. The PE estimates, on the other hand, can be seen to be quite unaffected by the specific force measurements biases, and always be slightly better than the external measuremetns themselves.



(a)



(b) magnified

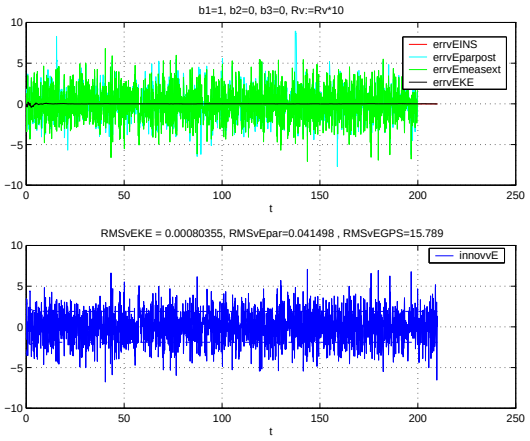
Figure 6.24: v_E error with bias in f_N-measurement.

	No bias	f_N**-bias**	f_N, f_E, f_D**-bias**
INS	$9.7 \cdot 10^{-5}$	2.6	2.6
KE, $Q_{INS} * 1$	$2.3 \cdot 10^{-3}$	0.81	0.81
KE, $Q_{INS} * 10$	-	0.48	0.48
KE, $Q_{INS} * 100$	-	0.28	0.28
PE	$4.0 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$
Ext meas (GPS)	$4.5 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$

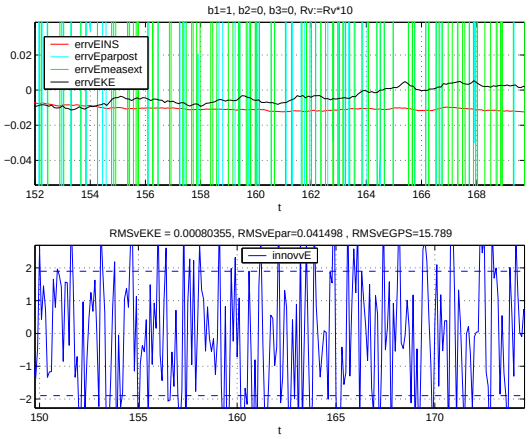
Table 6.1: RMS values for errors in the v_N component.

	No bias	f_N**-bias**	f_N, f_E, f_D**-bias**
INS	$1.7 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$	2.6
KE, $Q_{INS} * 1$	$0.8 \cdot 10^{-4}$	$8 .0 \cdot 10^{-4}$	0.80
KE, $Q_{INS} * 10$	-	$8.0 \cdot 10^{-4}$	0.47
KE, $Q_{INS} * 100$	-	$8.2 \cdot 10^{-4}$	0.26
PE	$4.1 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$
Ext meas, (GPS)	$4.5 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$

Table 6.2: RMS values for errors in the v_E component.



(a)

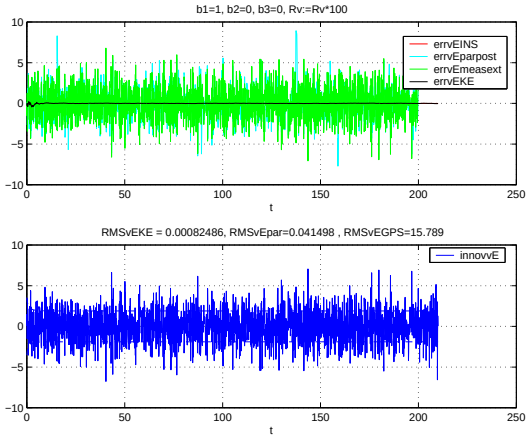


(b) magnified

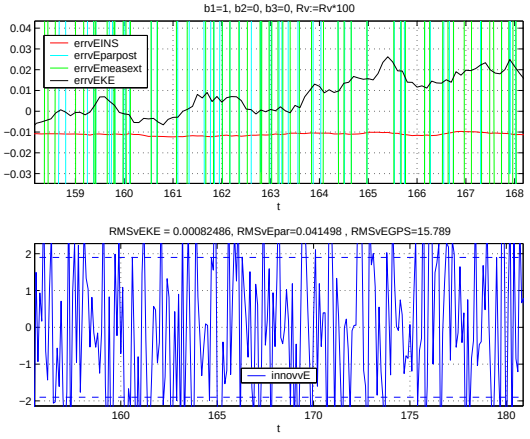
Figure 6.25: v_E error with bias in f_N-measurement and Q_{INS} multiplied by 10.

	No bias	f_N**-bias**	f_N, f_E, f_D**-bias**
INS	$7.3 \cdot 10^{-5}$	$4.3 \cdot 10^{-3}$	2.6
KE, $Q_{INS} * 1$	$7.0 \cdot 10^{-4}$	$2.2 \cdot 10^{-3}$	0.82
KE, $Q_{INS} * 10$	-	$1.5 \cdot 10^{-3}$	0.49
KE, $Q_{INS} * 100$	-	$1.1 \cdot 10^{-3}$	0.28
PE	$3.9 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$	$3.8 \cdot 10^{-2}$
Ext meas, (GPS)	$4.5 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$

Table 6.3: RMS values for errors in the v_D component.

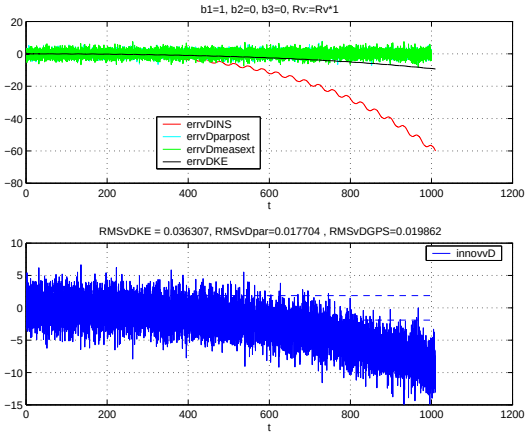


(a)

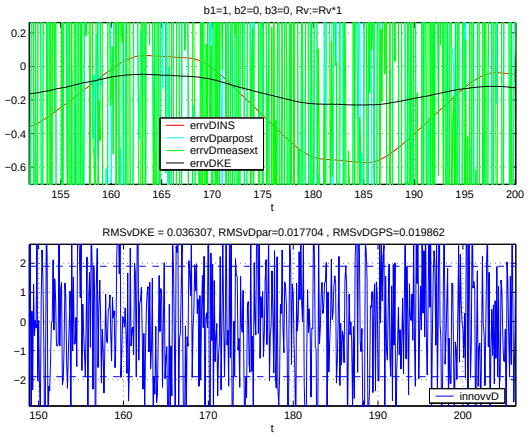


(b) magnified

Figure 6.26: v_E error with bias in f_N-measurement and Q_{INS} multiplied by 100.

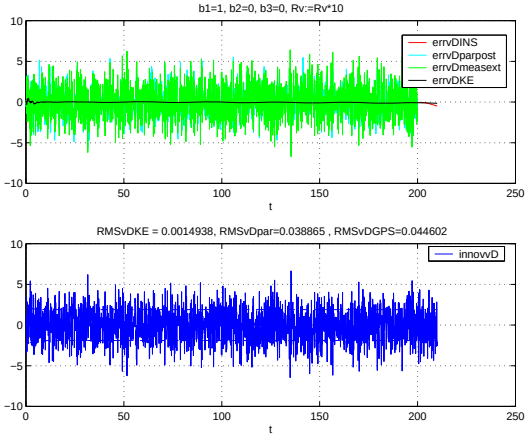


(a) $Q_{INS} * 1$

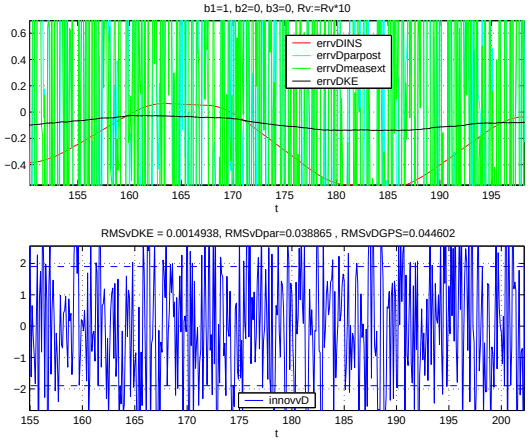


(b) $Q_{INS} * 1$, magnified

Figure 6.27: v_D error with bias in f_N-measurement.

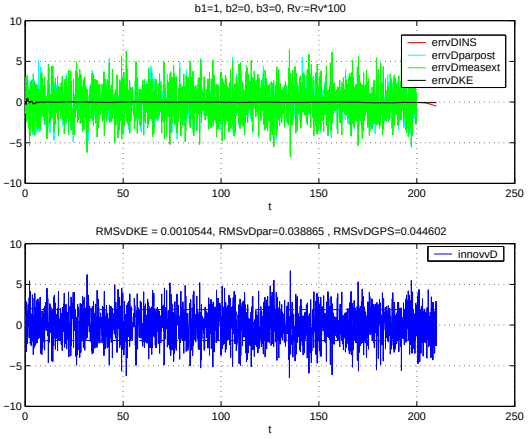


(a) $Q_{INS} * 10$

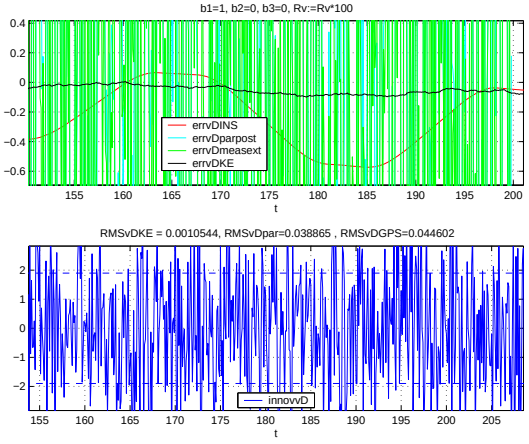


(b) $Q_{INS} * 10$, magnified

Figure 6.28: v_D error with bias in f_N-measurement and Q_{INS} multiplied by 10.

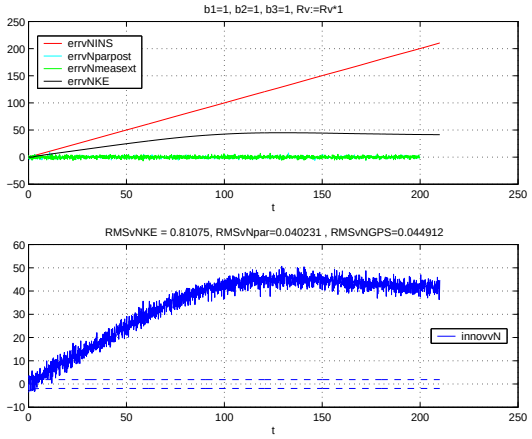


(a) $Q_{INS} * 100$

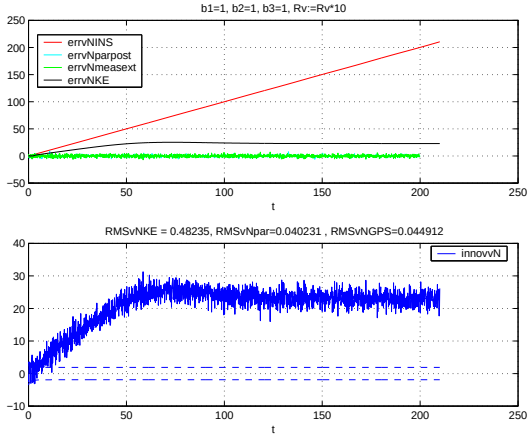


(b) $Q_{INS} * 100$, magnified

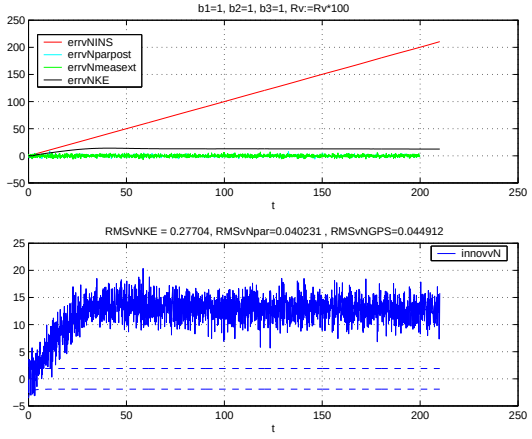
Figure 6.29: v_D error with bias in f_N-measurement and Q_{INS} multiplied by 100.



(a) $Q_{INS} * 1$

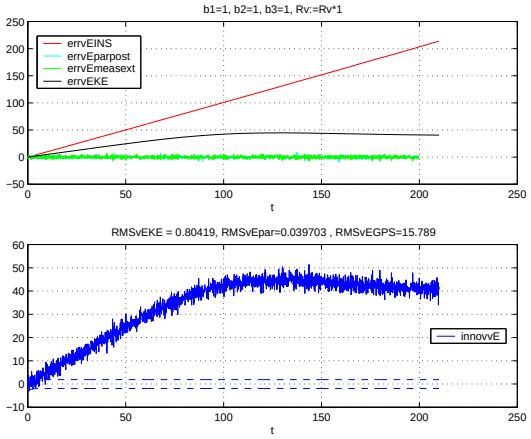


(b) $Q_{INS} * 10$

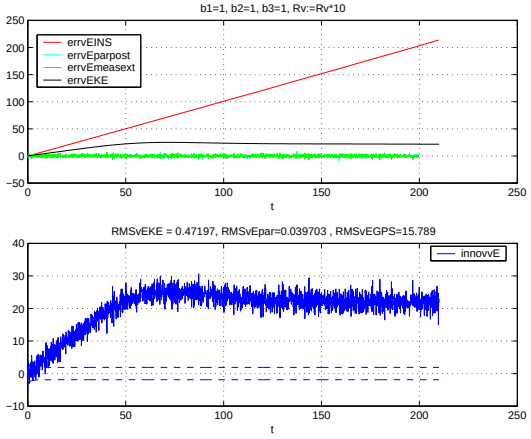


(c) $Q_{INS} * 100$

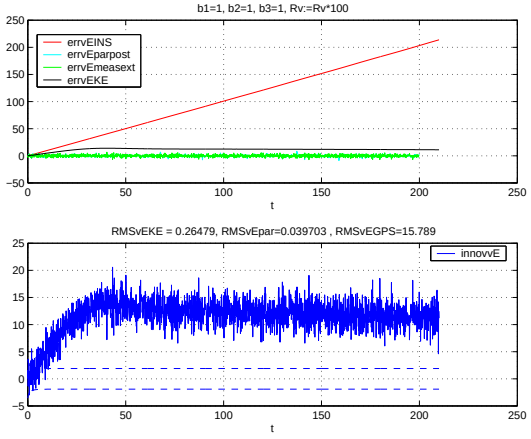
Figure 6.30: v_N error with bias in f_N-, f_E-, and f_D-measurements.



(a) $Q_{INS} * 1$

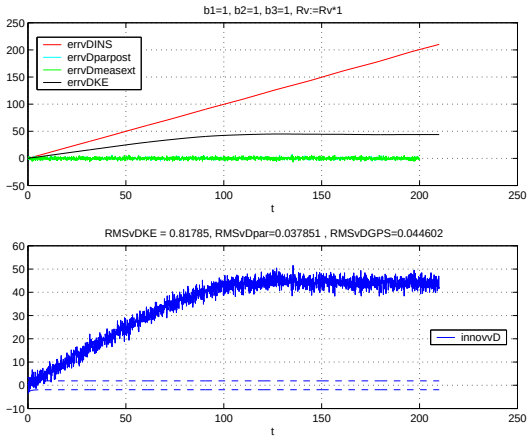


(b) $Q_{INS} * 10$

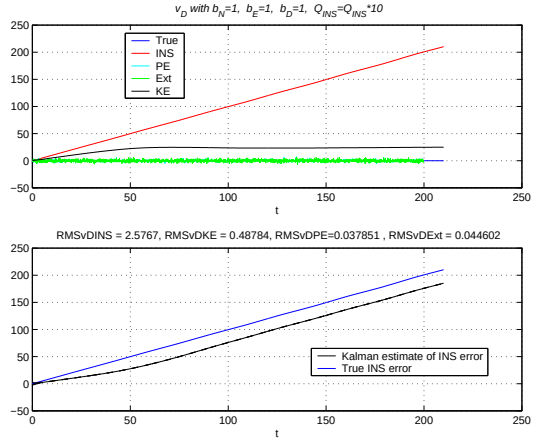


(c) $Q_{INS} * 100$

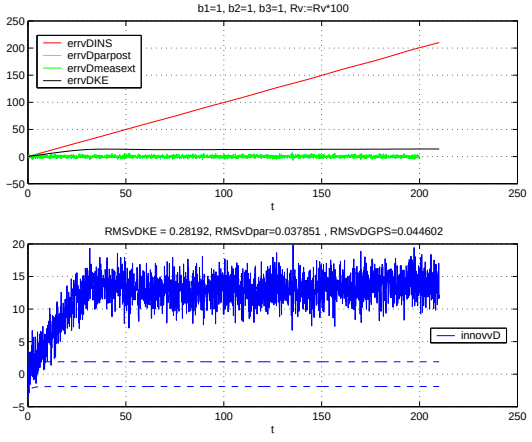
Figure 6.31: v_E error with bias in f_N-, f_E-, and f_D-measurements.



(a) $Q_{INS} * 1$



(b) $Q_{INS} * 10$



(c) $Q_{INS} * 100$

Figure 6.32: v_D error with bias in f_N-, f_E-, and f_D-measurements.

7. Simulation Environment

7.1 Simulink model of the Inertial Navigation System

The inertial navigation system for the special case with no attitudes is implemented in simulink, according to figure 7.1. The assumption of no attitudes means that the angular rate between the n-frame and the b-frame, ω_{nb} , is zero. All the output signals from this block are listed in table 7.2. The two green blocks,"INS" and "Reality", are identical and contain the navigation equations. The upper green block (INS) simulates the Inertial Navigations System with its noisy accelerometer input signals. The lower green block is the "ideal" INS, or the reality, which is fed with the true noise free accelerometer signals.

The Kalman estimator is based on a linear error model *nav_error2slw*, which is the upper red block. All Simulink S-functions are colored red in figure 7.1 and are briefly described in table 7.1. This model is evaluated by comparing its output with the true error, i.e. the difference between the outputs *navoutTN* and *realout.*

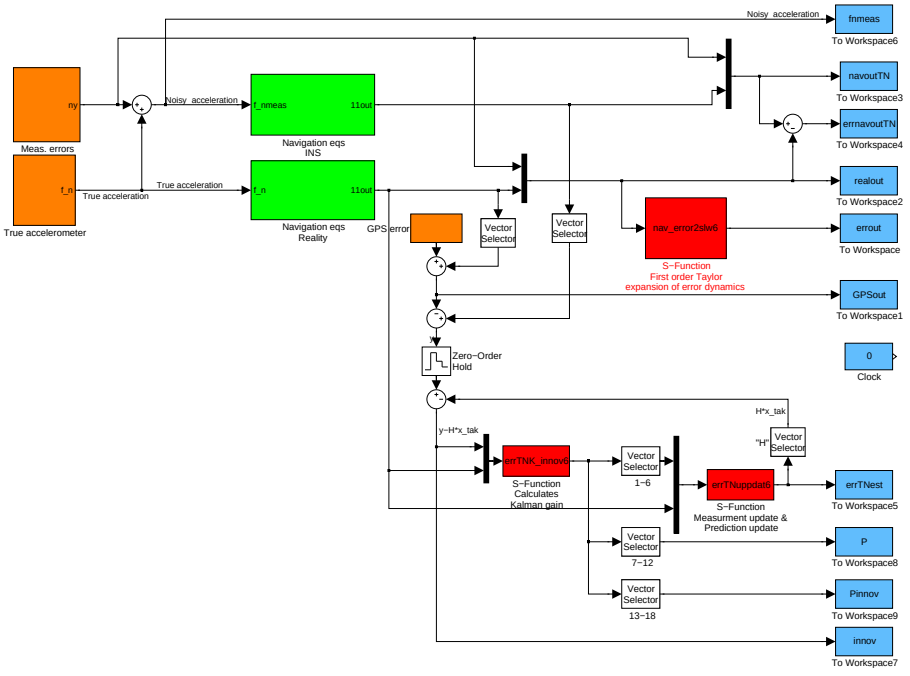


Figure 7.1: simulinkblock

The simulation parameters used are collected in table 7.1. The performance of the inertial sensors (accelerometers) are choosen to be in the same class as Litton LN200. Table 7.5 shows the stochastic parameters and initial variables for the integrated navigation system simulation model.

$^{(*)}$ Band limited white noise, given as noise power with sampling time 0.1 s.

S-Function/file	Description	Input	Output
nav_error2slw6.m	linear error dynamics	realout	errout
errTNK_innov6.m	continous KE gain K , covariances for the	$y - H\Delta\hat{x}$,x	$K\,(y - H\Delta\hat{x})$, $diag(P)$, $diag(Pinnov)$
errTNuppdat6.m	measurement update and time update of KE	$K\,(y - H\Delta\hat{x})$, x	errTNest

Table 7.1: Description of all S-functions, marked red in figure 7.1.

Output	Description	Size
fnmeas	noisy accelerometer signals	3
navoutTN	position, velocity integrated from noisy measurements	14
errnavoutTN	actual errors in position and velocity (navoutTN-realout)	14
realout	position and velocity integrated from true signals	14
errout	linear errors in position and velocity	6
GPSout	noisy GPS measurements in position and velocity	6
Clock	simulation time	1
errTNest	Kalman estimate of position and velocity	6
P	covariance of KE state vector	6
Pinnov	covariance of KE innovations, $y - H\Delta\hat{x}$	6
innov	innovations, measurments - predicted measurments, $y - H\Delta\hat{x}$	6

Table 7.2: Description of all outputs, marked blue, of the simulink block in figure 7.1.

Input signal	Description
K_innov=u(1:6)	K*(y - H*x_tak)
v_n=u(7)	Vel. north
v_e=u(8)	Vel. east
v_d=u(9)	Vel. down
L=u(10)	Latitude
l=u(11)	Longitude
h=u(12)	Height
r_n=u(13)	Earth north radius
r_e=u(14)	Earth east radius
f_n=u(15)	North Specific force expressed in N
f_e=u(16)	East Specific force expressed in N
f_d=u(17)	Down Specific force expressed in N

Table 7.3: Description of the inputsignals of the S-function errTNK innov6.

Description	Value
Solver	ode45 (Dormand-Prince)
Max step size	auto
Min step size	auto
Initial step size	auto
Relative tolerance	1e-3
Absolute tolerance	auto

Table 7.4: Parameters of the simulated Inertial Navigation System.

Description	Variable	Value	Unit
GPS white noise spectral density, position, velocity$^{(*)}$	$Cov\left(\Delta x_{ext}\right)$	$\mathrm{diag}(0.04/R^2,$ $0.04/R^2,$ $0.04,$ $0.04, 0.04, 0.04)$	$rad^2s,$ $rad^2s,$ $m^2s,$ $m/s^3,$ $m/s^3, m/s^3$
Accelerometer. White noise spectral density$^{(*)}$	$Cov\left(\Delta u\right)$	$\left(80 * 10^{-6}g\right)^2,$ $\left(80 * 10^{-6}g\right)^2,$ $\left(80 * 10^{-6}g\right)^2$	$\left(m/s^2\right)^2,$ $\left(m/s^2\right)^2,$ $\left(m/s^2\right)^2$
Initial values position, $x\left(0\right)$	$L\left(0\right),$ $\lambda\left(0\right),$ $h\left(0\right)$	$0, 0, 0$	rad, rad, m
Initial values velocity, $v\left(0\right)$	$v_N\left(0\right),$ $v_E\left(0\right),$ $v_D\left(0\right)$	$1000, 0, 0$	$m/s, m/s, m/s$
Initial INS position error $\Delta x\left(0\right)$ (linear error model)	$\Delta L\left(0\right),$ $\Delta\lambda\left(0\right),$ $\Delta h\left(0\right)$	$0, 0, 0$	rad, rad, m
Initial INS velocity error $\Delta v\left(0\right)$ (linear error model)	$\Delta v_N\left(0\right),$ $\Delta v_E\left(0\right),$ $\Delta v_D\left(0\right)$	$0, 0, 0$	$m/s, m/s, m/s$

Table 7.5: Parameters of the simulated Inertial Navigation System.

7.2 Estimator parameters

The Kalman Estimator (KE) is continuous and it estimates the errors $\Delta\hat{x}_{KE}$. The estimator parameters and initial variables are found in table 7.2.

Description	Variable	Value	Unit
initial estimates	$\Delta\hat{x}_{KE}\left(0\right)$	$0, 0, 0, 0, 0, 0$	$m/s,$ $m/s,$ $m/s,$ $rad,$ $rad,$ m
initial covariances	$Cov\left(\Delta\hat{x}_{KE}\left(0\right)\right)$	$1, 1, 1, 3{\cdot}10^{-12},$ $3 \cdot 10^{-12}, 3$	$\left(m/s\right)^2,$ $\left(m/s\right)^2,$ $\left(m/s\right)^2,$ $rad^2,$ rad^2, m^2
process noise covariance, INS	$Cov\left(\right) = Q_{INS}$		
measurement noise covariance, GPS	$Cov\left(\right) = R_{ext}$		

Table 7.6: Kalman estimator parameters.

The Particle Estimator (PE) estimates the full states $\hat{x}_{PE}$ and the estimator parameters and initial variables are found in table 7.7.

Description	**Variable**	**Value**	**Unit**
initial estimates	$\hat{x}_{PE}(0)$	$1000,\ 0,\ 0,\ 0, 0, 0$	$m/s,\ m/s,\ m/s,\ rad,\ rad, m$
initial covariances	$Cov(\hat{x}_{PE}(0))$	$1,\ 1,\ 1,\ 3 \cdot 10^{-12},\ 3 \cdot 10^{-12}, 3$	$(m/s)^2,\ (m/s)^2,\ (m/s)^2,\ rad^2,\ rad^2,\ m^2$
process noise covariance, INS	$Cov() = Q_{INS}$		
measurement noise covariance, GPS	$Cov() = R_{ext}$		
roughness			
param			
param			
param			

Table 7.7: Particle estimator parameters.

Appendix A: Band limited white noise

The Simulink model used in the simulations of this report works in continuous time. This means that all the white noise components, such as measurement noises, should be specified by means of spectral densities that are constant with respect to frequency. However, as is well known, such noise has infinite power and is therefore not realizable. The concept of continuous white noise can nevertheless be useful for modeling noise sources in continuous simulation models when it is input to a linear system with a bandlimited transfer function. The output signal from such a system will have finite power even when driven by continuous white noise.

In order to realize this white noise process in the simulation model, we look for a corresponding discrete white noise process that gives the same variance on the output as the given continuous white noise process would do. The great advantage with substituting the latter process with the former is that it is very easy to realize a discrete random process with a given covariance matrix. Therefore, suppose a continuous linear system driven by continuous white noise is given, described by the state equations

$$\begin{aligned} \dot{x}_c(t) &= Ax_c(t) + w_c(t), & (7.1)\\ y_c(t) &= Cx_c(t), & (7.2) \end{aligned}$$

where x_c is the state vector, w_c is the white noise process with a spectral density Q_c that is independent of frequency. y_c is the random signal vector at the output of the system, and A and C are constant matrices of appropriate dimensions.

If the process is started at $t = 0$ with zero initial values, the output can be written

$$y_c(t) = C\int_0^t e^{A(t-\tau)} w_c(\tau)\, d\tau. \tag{7.3}$$

Using the fact that $E\left[w_c(\tau)\, w_c^T(s)\right] = Q_c\delta(\tau - s)$, we can therefore write the covariance matrix of the output as

$$\begin{aligned} Cov(y_c(t)) &= E\left[y_c(t)\, y_c^T(t)\right]\\ &= E\left[C\left(\int_0^t e^{A(t-\tau)} w_c(\tau)\, d\tau\right)\left(\int_0^t w_c^T(s)\, e^{A^T(t-s)} ds\right) C^T\right]\\ &= C\cdot E\left[\int_{\tau=0}^t\int_{s=0}^t e^{A(t-\tau)} w_c(\tau)\, w_c^T(s)\, e^{A^T(t-s)} ds d\tau\right]\cdot C^T\\ &= C\cdot\int_{\tau=0}^t\int_{s=0}^t e^{A(t-\tau)} E\left[w_c(\tau)\, w_c^T(s)\right] e^{A^T(t-s)} ds d\tau\cdot C^T\\ &= C\cdot\int_{\tau=0}^t e^{A(t-\tau)} Q_c\left(\int_{s=0}^t \delta(\tau - s)\, e^{A^T(t-s)} ds\right) d\tau\cdot C^T\\ &= Ce^{At}\left(\int_{\tau=0}^t e^{-A\tau} Q_c e^{-A^T\tau} d\tau\right) e^{A^T t} C^T. \end{aligned} \tag{7.4}$$

Now we try to find the covariance matrix Q_d of a discrete white process $\{d_k\}_{k=0}^{\infty}$ that, when driving the same system, will give the same covariance matrix at the output. Therefore we will analyze the solution $y_d(t)$ to the system

$$\dot{x}_d(t) = Ax_d(t) + w_d(t), \tag{7.5}$$
$$y_d(t) = Cx_d(t), \tag{7.6}$$

where the input random signal $w_d(t)$ is piecewise constant according to

$$w_d(t) = d_k \text{ for } kh \le t \le (k+1)h, \tag{7.7}$$
$$E\left[d_k d_l^T\right] = Q_d \delta_{kl}. \tag{7.8}$$

If the step size h is small enough that $e^{-A\tau}$ can be regarded as constant during a sampling interval, i.e. when the variation in τ is less than h, then $\int_{kh}^{(k+1)h} e^{A(t-\tau)} d\tau \approx he^{A(t-kh)}$ if k is a positive integer, and for the output signal after $N = t/h$ discrete steps we get approximately

$$\begin{aligned} y_d(t) &= C\int_0^t e^{A(t-\tau)} w_d(\tau)\, d\tau \\ &= C\sum_{k=0}^{N-1}\left(\int_{kh}^{(k+1)h} e^{A(t-\tau)} d\tau\right) d_k \\ &= hC\sum_{k=0}^{N-1} e^{A(t-kh)} d_k. \end{aligned} \tag{7.9}$$

The covariance matrix for this signal is

$$\begin{aligned} Cov\left(y_d(t)\right) &= E\left[y_d(t)\, y_d^T(t)\right] \\ &= E\left[hC\left(\sum_{k=0}^{N-1} e^{A(t-kh)} d_k\right)\left(\sum_{l=0}^{N-1} d_l^T e^{A^T(t-lh)}\right) C^T h\right] \\ &= h^2 C\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} e^{A(t-kh)} E\left[d_k d_l^T\right] e^{A^T(t-lh)} C^T \\ &= h^2 C\sum_{k=0}^{N-1g}\sum_{l=0}^{N-1} e^{A(t-kh)} Q_d \delta_{kl} e^{A^T(t-lh)} C^T \\ &= h^2 C\sum_{k=0}^{N-1} e^{A(t-kh)} Q_d e^{A^T(t-kh)} C^T \\ &= Ce^{At}\left(\sum_{k=0}^{N-1} h\cdot e^{-Akh}\left(h\cdot Q_d\right) e^{-A^T kh}\right) e^{A^T t} C^T. \end{aligned} \tag{7.10}$$

For h small enough, the sum inside the big parenthesis is approximately equal to the integral inside the parenthesis of (7.4) provided that we choose

$$Q_d = \frac{Q_c}{h}, \tag{7.11}$$

which means that if the continuous white process with spectral density Q_c is replaced with a discrete white process with covariance Q_c/h at the input of the linear system, the covariance at the output will be unchanged.

Bibliography

[1] N.J. Gordon D.J. Salmond A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings-F*, volume 140, April 1993.

[2] Fredrik Gustafsson Fredrik Gunnarsson Niclas Bergman Urban Forsell Jonas Jansson Rickard Karlsson Per-Johan Nordlund. Particle Filters for Positioning, Navigation and Tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, February 2002.

[3] Per-Johan Nordlund Fredrik Gustafsson. Sequential monte carlo filtering techniques applied to integrated navigation systems. In *Proceedings of the American Control Conference*, pages 4375–4380, Arlington,VA, June 2001. AACC.

[4] B. D. O. Andersson J. B. Moore. *Optimal Filtering*. Prentice Hall, 1979.

[5] F. L. T. Eklöf J. G. J. Rantakokko L. E. Pettersson. On broadband adaptive beamforming for tactical radio systems. In *IEEE Military Communications Conference (MILCOM)*, Washington D. C., October 2001.

[6] M. S. Grewal L. R. WeillA. P. Andrews. *Global Positioning Systems, Inertial Navigation and Integration*. John Wiley & Sons, 2001.

[7] R. G. Brown P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 1997.

[8] Niclas Bergman. *Recursive Bayesian Estimation, Navigation and Tracking Applications*. PhD thesis, Linköping University, 1999.

[9] Don Herskovitz. GPS insurance, antijamming the system. *The Journal of Electronic Defence*, 23(12):41–45, December 2000.

[10] Christopher Jekeli. *Inertial Navigation Systems with Geodetic Applications*. de Gruyter, 2001.

[11] Rickard Karlsson. *Simulation Based Methods for Target Tracking*. PhD thesis, Linköping University, 2002.

[12] Per-Johan Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. PhD thesis, Linköping University, 2002.

[13] Sven-Lennart Wirkander. Particle estimation, basic theory. Technical Report FOI-R–0473–SE, Swedish Defence Research Agency, April 2002.