UDK: 681.883.48:519.62
COSATI: 17-07

Comparison of Different Computation Methods for Strapdown Inertial Navigation Systems

Mohamed S. Ahmed, MSc (Eng)[1)]
Danilo V. Ćuk, PhD (Eng)[1)]

A series of numerical experiments were conducted to test three different methods for solving the SDINS navigation equations: Runge-Kutta method, Runge-Kutta method with sampling process and three speed navigation algorithms. A stochastic numerical simulator, which solves the navigation equations in navigation frame by using the given functions of Euler's angles and velocity components as inputs, is proposed to simulate the IMU sensors output. The generated angular body rates and specific force were inputs to the three integration methods. For the case when the sensors outputs are contaminated by white noise the obtained results shows that the ratio of the absolute error by navigation algorithm to the absolute error obtained by Runge-Kutta sampling method is reduced compared to the same quantity without the noise.

Key words: **navigation system, inertial navigation, numerical methods, Runge-Kutta method.**

Nomenclature

Convention

$\mathbf{C}_{A_1}^{A_2}$ – direction cosine matrix which transforms vector from A_1 to A_2 frame
$\mathbf{b}^{A_1}$ – vector $\mathbf{b}$ with components in A_1 frame
$\boldsymbol{\omega}_{A_1A_2}^{A_2}$ – angular rate of A_2 frame relative to A_1 frame expressed with components in A_2 frame

Axis systems (reference frames), angles and transformation operators

i – inertial reference frame
e – Earth-fixed reference frame
n – navigation reference frame
b – body reference frame
$O x_i\, y_i\, z_i$ – inertial axis system
$O x_e\, y_e\, z_e$ – Earth-fixed axis system
$O_n\, x_n\, y_n\, z_n$ – navigation axis system
$O x_b\, y_b\, z_b$ – body axis system
h – altitude
ϕ – latitude
λ – longitude
Φ – roll angle
θ – pitch angle
Ψ – yaw angle
$\mathbf{C}_b^n$ – direction cosine matrix transforming quantities from b frame to n frame
$\mathbf{C}_n^b$ – direction cosine matrix transforming quantities from n frame to b frame where $\mathbf{C}_n^b = (\mathbf{C}_b^n)^{\mathrm{T}}$

Earth quantities

g_n – nominal gravitational acceleration $(\phi = 45^\circ)$
R_0 – mean radius of Earth, R_0=6356766 m
ω_{ie} – Earth rate with respect to i frame, $\omega_{ie} = 7.292116\text{x}10^{-5}$ rad/s
$\mathbf{g}_l$ – local gravity column matrix

Dynamic quantities

t, T – time, sampling time
V_e – kinematic velocity (velocity of the vehicle relative to the Earth)
V_N, V_E, V_D – the north, east and down components of kinematic velocity in n frame
$\mathbf{V}_e^n$ – kinematic velocity expressed in n frame
$\boldsymbol{\omega}_{nb}^b$ – angular rate of b frame relative to n frame expressed in b frame
$\boldsymbol{\omega}_{en}^n$ – angular rate of n frame relative to e frame expressed in n frame
$\boldsymbol{\omega}_{ib}^b$ – angular rate of b frame relative to i frame expressed in b frame
f_x, f_y, f_z – components of body specific force expressed in b frame
$\mathbf{f}^b$ – specific force in b frame
$\mathbf{f}^n$ – specific force expressed in n frame

[1)] Faculty of Mechanical Engineering, Kraljice Marije 16, 11120 Belgrade

Definition of important matrices

$(\boldsymbol{\omega}^A \text{x})$ – skew symmetric matrix with components of $\boldsymbol{\omega}$ in A frame

$$(\boldsymbol{\omega}^A \text{x}) = \boldsymbol{\Omega}(\boldsymbol{\omega}^A) = \begin{bmatrix} 0 & -\omega_{z_A} & \omega_{y_A} \\ \omega_{z_A} & 0 & -\omega_{x_A} \\ -\omega_{y_A} & \omega_{x_A} & 0 \end{bmatrix}$$

Subscripts

l, m and *n* – indexes for high speed computer cycle (*l*-cycle), moderate speed computer cycle (*m*-cycle) and low speed computer cycle (*n*-cycle) respectively

N, E and *D* – North, East and Down components (*n* frame components)

Abbreviations

IMU – inertial measurement unit
INS – inertial navigation system
SDINS – strapdown inertial navigation system
DCM – direction cosine matrix
RK – Runge-Kutta method
RKS – Runge-Kutta method with sampling process
NA – navigation algorithm

Introduction

The original applications of inertial navigation technology used stable platform techniques. In such navigation technology the inertial sensors are mounted on stabilized platform and are mechanically isolated from the rotational motion of the vehicle. Platform systems are still in common use particularly for those applications requiring very accurate estimates of navigation data such as ships and submarine. This type of inertial navigation system is referred to as Space Stabilized Inertial Navigation System (SSINS). Most of the mechanical complexity of the platform systems has been removed by having the sensors attached rigidly or strapped down to the body of the host vehicle. This type of inertial navigation system is referred to as StrapDown Inertial Navigation System (SDINS).

The basic strapdown inertial navigation concept was originally formulated in the 1950s. The main benefits of this type are lower cost, size reduction and greater reliability compared with equivalent platform systems. As a result a small, light weight and accurate inertial navigation systems can now be fitted to a small vehicle. The major drawbacks are a substantial increase in computing complexity and the need to use sensors capable of measuring much higher rates of turn. However, recent advance in computer technology combined with the development of suitable sensors have allowed such a design to become reality.

The general structure of the inertial navigation system is shown in Fig.1 along with other computational procedures implemented onto on board computer of a flying vehicle. Fig.1 shows that the measured vehicle angular rates $\boldsymbol{\omega}_{ib}^b$ and specific forces $\mathbf{f}^b$ taken by the gyroscopes and accelerometers respectively are passed through filtering algorithms to provide the estimated values of these measurements which might be corrupted by noise.

The filtered angular rate measurements obtained from filtering algorithms are input to both the attitude algorithms and the integrated control and guidance law. The attitude algorithms compute the transformation matrix (Direction Cosine Matrix or Quaternion) that will transform the measured specific forces (non-gravitational acceleration) into the desired navigation frame.

The velocity and position are computed by the transformed specific forces measurements using the computed transformation matrix obtained from attitude algorithms. These computed values of velocity and position are also input to the control and guidance law. The proper commands are sent to the vehicle in order to correct its course according to the comparison of the computed values of the velocity position provided by the INS system and the nominal course. The vehicle response to the control commands is sensed by the INS sensors (gyroscopes and accelerometers), the measured values are passed again through the filter algorithms and new cycle of computations starts.

The problem of computing the translational velocity and position relative to the Earth, which has to be solved in INS computer, was addressed by Itzhack in 1977 [1]. Various computational schemes were considered and the split-coordinate computational scheme was selected in which the differential equations are solved in three different computational rates. A computer simulation was carried out on a coning motion example (the same example adopted in this paper). The integration routine utilized a fixed Runge-Kutta procedure with a time step of 0.001s. The obtained results showed a good accuracy.

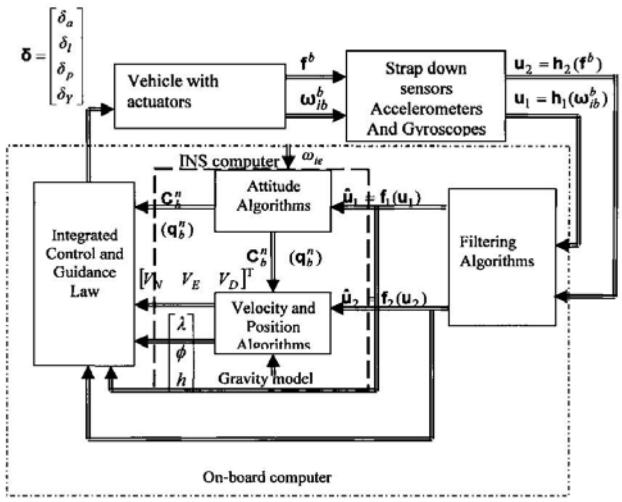

Figure 1. Strapdown inertial navigation system

Dorobantu in 1999 [2] presented some labor simulation results for a low cost SDINS system with zero Earth rotation and constant gravity, due to poor INS sensors and small area of trajectory, using the graphical programming language Simulink. The principles of the implementation of SDINS algorithm and the influence of integration time step and method were tested. The attitude was updated by solving Euler's angles differential equation and then a DCM was constructed using these angles. The sampling frequency up to 50Hz and the 4th order Runge-Kutta integration method were recommended.

Savage in 1998 [3], [4] developed three – speed navigation algorithms (NA) for the computation of attitude parameters, velocity and position.

Waldmann in 2002 [5] presented a novel derivation of a discrete-time version of the relative quaternion differential equation. The proposed quaternion algorithms were simulated assuming perfect sensors and showed robustness to different trajectories. The performance degraded after raising the cone angle.

Apart from, references [3] and [4] probably there are no other published materials dealing with the design of SDINS complete navigation algorithms, though these are without numerical analysis in the presence of random inputs. The general flow chart and computer program for the proposed three speed algorithms were given in [6].

The aim of this paper is to test the accuracy and performances of this numerical algorithms (NA) and compare it with other methods of solving navigation equations (Appendix A) such as the classical fourth order Runge-Kutta method (RK) and the Runge-Kutta method with sampling process (RKS). The numerical analysis will be given for different types of motion without noise and in the presence of the white noise.

Generation of specific force and angular rate

The integrated specific force and angular rate are used as inputs to the navigation algorithms. The numerical simulator used for generating these input data was developed by solving the navigation equation expressed in n frame given in [7] (Titterton, page 51). This equation can be rewritten for the specific force expressed n frame as

$$\mathbf{f}^n = \dot{\mathbf{V}}_e^n + \left[\mathbf{\Omega}(\boldsymbol{\omega}_{en}^n) + 2\,\mathbf{\Omega}(\boldsymbol{\omega}_{ie}^n)\right]\mathbf{V}_e^n - \mathbf{g}_l^n \tag{1}$$

where the kinematic velocity is

$$\mathbf{V}_e^n = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} \tag{2}$$

To solve Eq.(1) the velocity components and Euler's angles of a vehicle should be available as functions of time and their first derivatives should be computed. The velocity components and Euler's angles and their derivatives are used as input for the simulator. Also, the angular rate vectors $\left(\boldsymbol{\omega}_{en}^n \text{ and } \boldsymbol{\omega}_{ie}^n\right)$ and the gravity vector $\mathbf{g}_l^n$ should be computed:

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} \dot{\lambda}\cos\phi \\ -\dot{\phi} \\ -\dot{\lambda}\sin\phi \end{bmatrix} \tag{3}$$

$$\boldsymbol{\omega}_{ie}^n = \begin{bmatrix} \omega_{ie}\cos\phi \\ 0 \\ -\omega_{ie}\sin\phi \end{bmatrix} \tag{4}$$

Substituting Eqs.(3) and (4) into Eq.(1) the specific force vector can be obtained. The position in terms of latitude, longitude and altitude can be calculated as a function of the velocity vector components from [7] (Titterton, page 53):

$$\dot{h} = -V_D$$

$$\dot{\lambda} = \frac{1}{(R_0 + h)\cos\phi} V_E \tag{5}$$

$$\dot{\phi} = \frac{1}{(R_0 + h)} V_N$$

The specific force vector expressed in n frame computed by Eq.(1) should be transformed to b frame using the direction cosine transformation matrix in terms of Euler's angles given in [7] (page 49). This transformation is written as

$$\mathbf{f}^b = \mathbf{C}_n^b \mathbf{f}^n \tag{6}$$

Mathematically the absolute angular velocity is determined by summing the angular velocity of the b frame relative to n frame $\boldsymbol{\omega}_{nb}$ and the angular velocity of n frame relative to i frame $\boldsymbol{\omega}_{in}$:

$$\boldsymbol{\omega}_{ib} = \boldsymbol{\omega}_{in} + \boldsymbol{\omega}_{nb} \tag{7}$$

The body angular velocity relative to the n frame $\boldsymbol{\omega}_{nb}$ (relative angular velocity) with components in b frame is given by

$$\boldsymbol{\omega}^r = \boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b\,\boldsymbol{\omega}_{in}^n \tag{8}$$

$$\boldsymbol{\omega}_{in}^n = \boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n \tag{9}$$

where

- $\boldsymbol{\omega}_{ib}^b$ – is the angular velocity column vector of b frame relative to i frame expressed in b frame;
- $\boldsymbol{\omega}_{in}^n$ – is the angular velocity column vector of n frame relative to i frame expressed in n frame and it is given in by (9), (3) and (4);
- $\boldsymbol{\omega}_{nb}^b$ – is the angular velocity column vector of b frame relative to n frame expressed in b frame;
- $\mathbf{C}_n^b$ – is the transformation matrix related b frame relative to n frame and it is given in [7] (page 49).

The relative angular velocity of the b frame to the n frame is related to the angular rates of Euler's angles as follows [7]:

$$\boldsymbol{\omega}^r = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\Phi} - \dot{\Psi}\sin\theta \\ \dot{\theta}\cos\Phi + \dot{\Psi}\cos\theta\sin\Phi \\ \dot{\Psi}\cos\theta\cos\Phi - \dot{\theta}\sin\Phi \end{bmatrix} \tag{10}$$

The angular velocity column vector $\boldsymbol{\omega}_{ib}^b$ (measured angular rates) can be found form (8).

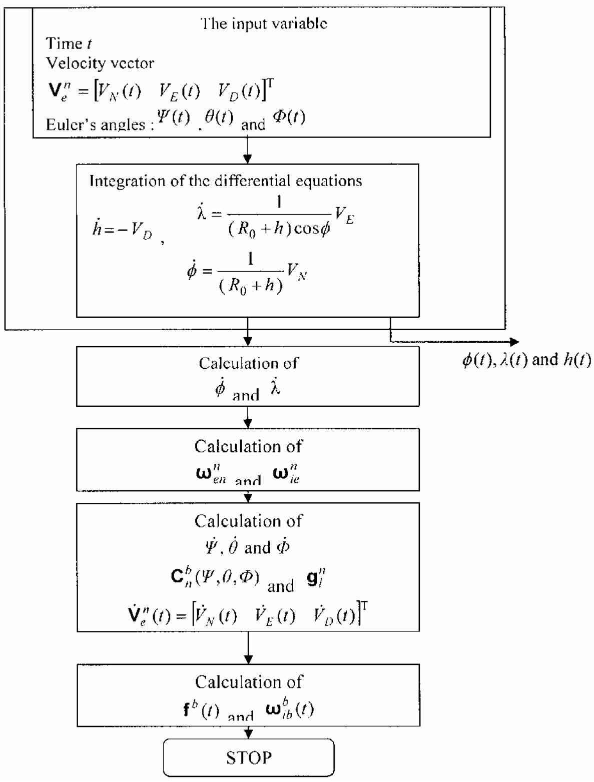

Figure 2. Flow chart of the numerical simulator

The flow chart of the generator (simulator) of the specific force and the angular velocity is shown in Fig.2. The numerical simulator for the generation of the angular velocity and specific force is included into the computer program [6] for the computation of the kinematic parameters according to the chosen method of integration navigation equations. Depending on the applied numerical method (RK, RKS, NA), it is possible to generate instantaneous values of the specific force and the angular velocity of the vehicle or their integrated increments at the desired time during high frequency cycle.

Simulation examples

In order to evaluate the performance of the proposed navigation digital algorithms a computer code was written using the developed algorithms shown in [3], [4] and [6]. A series of numerical experiments were conducted for two examples.

Ballistic trajectory (Example 1)

This example represents a simple motion where the object has a pure ballistic trajectory without any disturbances and with the gravitational acceleration which is equal to g=9.81 m/s^2.

Velocity components

The velocity components in North, East and Down directions are given by the following expression

$$\mathbf{V}=\begin{bmatrix}V_N\\V_E\\V_D\end{bmatrix}=\begin{bmatrix}V_0\cos\theta_0\\0\\-\left(V_0\sin\theta_0-gt\right)\end{bmatrix}\tag{11}$$

The first derivative of the velocity is

$$\dot{\mathbf{V}}=\begin{bmatrix}\dot V_N\\\dot V_E\\\dot V_D\end{bmatrix}=\begin{bmatrix}0\\0\\g\end{bmatrix}\tag{12}$$

The calculated numerical value of the initial velocity for ~150 km is 1200 m/s. The initial latitude, longitude and altitude are: $\phi_0=0.5$ radian, $\lambda_0=0$ and $h_0=0$ m.

Euler's angles

The values of the yaw and roll angles and their derivatives are set to equal zero

$$\begin{aligned}\Psi=\dot\Psi=0\\\Phi=\dot\Phi=0\end{aligned}\tag{13}$$

The pitch angle expression and its derivative can be obtained from (11).

$$\tan\theta=\frac{V_0\sin\theta_0-gt}{V_0\cos\theta_0}\tag{14}$$

The numerical value for the launch angle θ_0 =45^{o} was selected to have maximum down range.

The derivative of the pitch angle is obtained as

$$\dot\theta=-\frac{g}{V_0\cos\theta_0}\cos^2\theta\tag{15}$$

In the case of a pure ballistic trajectory gravitational acceleration is constant (for $\phi=45^{o}$, g = 9.81 m/s^2) and the specific force equal zero. The angular rate is produced only in pitch plane due to the gravity. It is important to remark that for solving navigation equations the gravitational acceleration is variable and it depends on the altitude. Also there is the effect of Coriolis acceleration due to Earth rotation. So, in order to produce a pure ballistic trajectory in the navigation frame under these conditions it is necessary to generate specific force and angular rate to compensate variable gravitational and the Coriolis acceleration. The body angular rates components for this example are shown in Fig.3. The specific force profiles are shown in Fig.4.

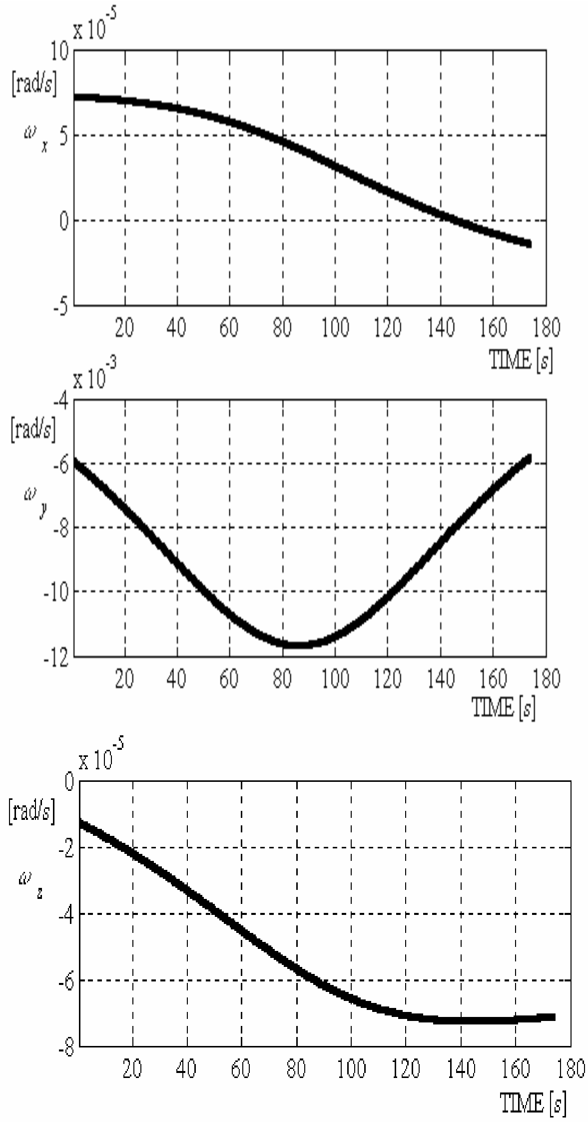

Figure 3. Body angular body rates profiles for Example 1

Conning motion (Example 2)

This example demonstrates the effect of the vehicle coning motion of low amplitude. The dynamics of the vehicle were described by a velocity whose components were

$$\mathbf{V}=\begin{bmatrix}V_N\\V_E\\V_D\end{bmatrix}=\begin{bmatrix}300+100t\\300+100t\\-(300+100t)\end{bmatrix}\tag{16}$$

and by three Euler's angles which described the rotation of the vehicle with respect to n frame.

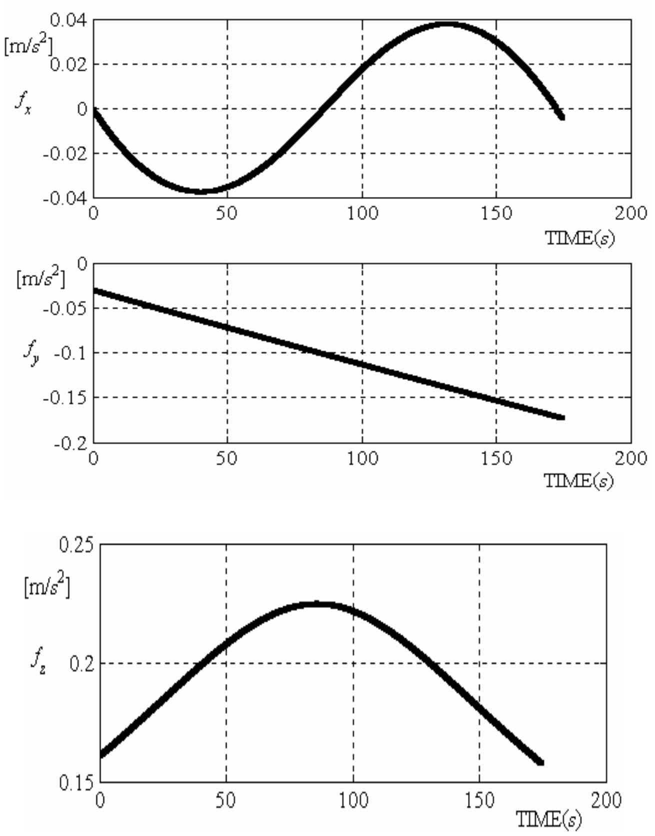

Figure 4. Specific force profiles for Example 1

The three angles were

$$\Psi = \sin\left(\frac{2\pi}{300}t\right) + \frac{1}{2}\sin\left(\frac{2\pi}{1.7}t\right)$$

$$\Phi = \sin\left(\frac{2\pi}{300}t\right) + \frac{1}{2}\sin\left(\frac{2\pi}{0.85}t\right) \qquad (17)$$

$$\theta = \sin\left(\frac{2\pi}{300}t\right) + \frac{1}{2}\sin\left(\frac{2\pi}{1.7}t + 0.3\right)$$

Numerical analysis

The navigation equations (Appendix A) were solved by three methods. The first one is solving these equations using the classical fourth order Runge-Kutta integration method (RK) and continuous form solution for the specific force and angular velocity generated by numerical simulator. This method with integration time step of 0.001s was used for two purposes. The first is to check the accuracy of the generated specific forces and body angular rates obtained by simulator. This can be done by comparing the output results of velocity components and Euler's angles obtained by this method with input values of these parameters for the numerical simulator. The second purpose is to obtain values of the velocity components and Euler angles with maximum accuracy.

The second numerical method is solving navigation equations using Runge-Kutta method with sampling process (RKS) where the specific force and angular rate are constant during the interval of sampling, T_l=0.001.

Finally, these equations were solved by the navigation algorithms (NA), presented in [3], [4] and [6].]. The sampling rate chosen for these experiments is 1KHz, viz. the data with high speed algorithm (l-cycle) are sampled at a period of T_l = 0.001s. The attitude and velocity updating are performed at moderate speed algorithm (m-cycle) slower than the l-cycle by 10 times (100Hz with updating period of T_m = 0.01 s). The position is updated at slower rate than the moderate cycle by 5 times that means the position algorithms (n-cycle) has an updating rate of 20Hz with updating period of T_n = 0.05s.

The generated specific forces and angular body rates were also corrupted by a white noise in order to test the performances of the numerical methods in noisy environment. The standard deviations of the white noise for the body angular rates and specific force were chosen for these experiments as: for body angular rates σ_ω = 1°/hr and for specific forces σ_f = 1mg.

The results of the maximum absolute errors of some kinematic parameters obtained by different numerical methods are shown in Table 1.

The high accuracy of the computation of kinematic parameters was achieved for both ballistic trajectory (example 1) and conning motion (example 2) by Runge-Kutta method with continuous form solution for the specific force and angular velocity. The sampling process in the Runge-Kutta method increases maximum absolute errors of order between 10^{-11} and 10^{-7} to the errors of order between 10^{-10} to 10^{-6} (Table 1).

The results of the three speed navigation algorithms (NA) are good from the practical point of view, especially for the trajectory which is near to the pure ballistic type. The coning effect is better computed by using RK-sampling method. The ratio of the error in computed angle is of the order of 10^6 between NA and RKS method when there is no noise. The corruption of the input data with noise increases the error dramatically compared to the error obtained using the same method in free noise environment. For example, the absolute error $|\Delta\Psi|$ in the presence of the white noise is increased ~25 times comparing with the case of no noise added for the example with coning motion (example 2) by using NA method. The absolute error $|\Delta V_E|$ obtained in the presence of the white noise at the end of simulation time (t = 200s) is less than 0.172% of the exact value (20300 m/s) for RK-sampling method, and 0.344% of the exact value for the NA method. The error obtained by NA in the presence of noise is 2 times the error of RK-sampling method.

Table 1. Maximum absolute errors in computed values of some kinematic parameters

example	parameter	RK	No noise		With noise	
			RKS	NA	RKS	NA
1	$\lvert\Delta\theta\rvert$ (degree)	4.5x10^{-11}	2.6x10^{-10}	2x10^{-8}	1.5x10^{-4}	5x10^{-4}
	$\lvert\Delta V_D\rvert$ (m/s)	3x10^{-10}	6x10^{-9}	2x10^{-6}	5.3x10^{-3}	3x10^{-3}
2	$\lvert\Delta\Psi\rvert$ (degree)	2x10^{-9}	9x10^{-8}	7.1x10^{-2}	0.88	1.75
	$\lvert\Delta V_E\rvert$ (m/s)	2.2x10-7	1.75x10-6	0.91	35	70

It can be said that the NA and RK-sampling methods give the errors of the same order when the noise is introduced. The ratio of the absolute error in V_D obtained by the RK-sampling method to the error obtained by NA method for this particular simulation is ~1.8 (Table 1).The error of the elevation angle obtained by NA is about 3-times greater than the error produced by Runge-Kutta – sampling method.

Conclusions

A series of numerical experiments were conducted to test three different methods for solving the SDINS navigation equations. These methods are: Runge-Kutta method with continuous form solution for the specific force and angular velocity, Runge-Kutta method with sampling process and the navigation algorithms, given in [3], [4] and [6]. Two different examples representing different types of motion were used to test these methods: the pure ballistic trajectory and the conning motion.

The measured quantities (angular body rates and specific force) were generated by the proposed numerical simulator which solves the navigation equations in navigation frame by using the given function of time of Euler's angles and velocity components as inputs.

The free white noise experiments showed that the RK method with 1 KHz computation frequency is very accurate but time consuming (for ballistic trajectory the error in angles is of 10^{-11} degree order and the error in the velocity components is of 10^{-10} m/s order).

In order to study the effect of the white noise and to compare the proposed navigation algorithm with the RK-sampling method the white noise was added to the simulated measured quantities and then the navigation equations were solved by both methods.

The results with white noise showed that the ratio between the absolute error by NA and the absolute error obtained by RK-sampling method is reduced compared to the same quantity without noise. This means that the proposed NA is effective for real time applications. The advantage of the RK-sampling method over the proposed NA in terms of the accuracy of solving navigation equations is dramatically reduced for the real environment application.

Appendix A: Navigation Equations

In order to write navigation equations the following frames were used: inertial (i), Earth fixed (e) and navigation (n), Fig.A.1.

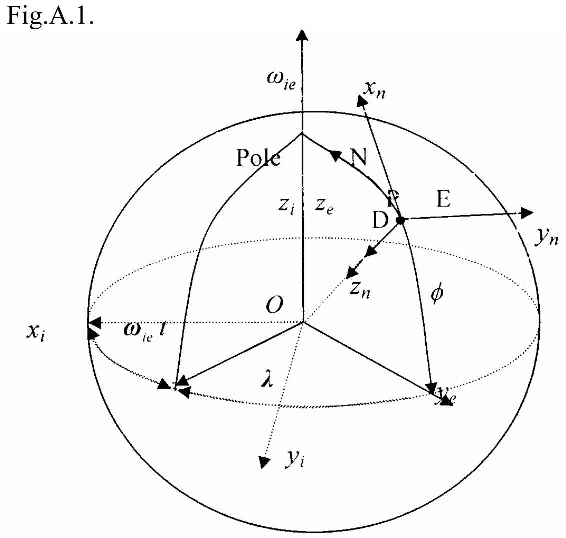

Figure A.1. Inertial, Earth fixed and navigation reference frames

The navigation frame mechanization was chosen. For this type of mechanization the two frames of interest are the navigation frame *(n)* and the body frame *(b)*. The rate of change of direction cosine matrix is

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n\,\mathbf{\Omega}(\boldsymbol{\omega}_{ib}^b) - \mathbf{\Omega}(\boldsymbol{\omega}_{in}^n)\,\mathbf{C}_b^n \tag{A.1}$$

where

$\boldsymbol{\omega}_{ib}^b$ – the angular rate of b frame (body angular rate) relative to i frame expressed in b frame, this quantity in fact measured by gyroscopes.

$\boldsymbol{\omega}_{in}^n$ – the angular rate of n frame relative to i frame expressed in n frame.

The general updating algorithm for direction cosine matrix $\mathbf{C}_b^n$ was constructed by using direction cosine matrix product chain rule [3], [4]:

$$\mathbf{C}_{bi(m)}^{ni(n\text{-}1)} = \mathbf{C}_{bi(m\text{-}1)}^{ni(n\text{-}1)}\ \mathbf{C}_{bi(m)}^{bi(m\text{-}1)} \tag{A.2}$$

$$\mathbf{C}_{bi(m)}^{ni(n)} = \mathbf{C}_{ni(n\text{-}1)}^{ni(n)}\ \mathbf{C}_{bi(m)}^{ni(n\text{-}1)} \tag{A.3}$$

where

$\mathbf{C}_{bi(m-1)}^{ni(n\text{-}1)}$ – DCM relating the b frame at time t_{m-1} to the n frame at time t_{n-1}.

$\mathbf{C}_{bi(m)}^{ni(n)}$ – DCM relating the b frame at time t_m to the n frame at time t_n.

$\mathbf{C}_{bi(m)}^{bi(m\text{-}1)}$ – DCM that accounts for b frame rotation relative to inertial frame from its orientation at the time t_{m-1} to its orientation at time t_m.

$\mathbf{C}_{ni(n\text{-}1)}^{ni(n)}$ – DCM that accounts for n frame rotation relative to inertial frame from its orientation at the time t_{n-1} to its orientation at time t_n.

$b_{i(m)}$ – discrete orientation of the b frame in the non-rotating i frame at computer update time t_m.

m – computer cycle index for b frame angular motion updates to $\mathbf{C}_b^n$.

$n_{i(n)}$ – discrete orientation of the n frame in the non-rotating i frame at computer update time t_n.

n – computer cycle index for n frame angular motion updates to $\mathbf{C}_b^n$.

The orientations of both b frame and n frame relative to each other and to the non-rotating i frame is illustrated in Fig.A.2.

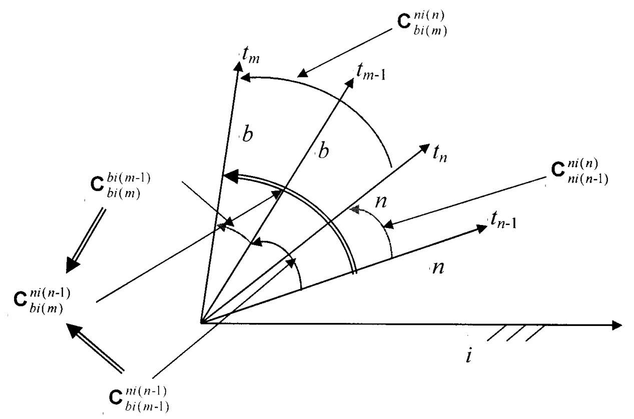

Figure A.2. Relation between i, b and n frames orientations

The equations (A.2) and (A.3) describe an algorithm that relates b frame and n frame orientations at separate times and provides, for both frames, inertial angular motion updates to $\mathbf{C}_b^n$ at different update rates. This angular motion updates are performed by $\mathbf{C}_{bi(m)}^{bi(m\text{-}1)}$ and $\mathbf{C}_{ni(n\text{-}1)}^{ni(n)}$ terms in Eqs. (A.2) and (A.3) for which algorithms are derived separately.

The velocity derivative expressed in navigation frame is given by

$$\dot{\mathbf{V}}_e^n = \mathbf{C}_b^n \mathbf{f}^b - [\boldsymbol{\Omega}(\boldsymbol{\omega}_{en}^n) + 2\boldsymbol{\Omega}(\boldsymbol{\omega}_{ie}^n)]\mathbf{V}_e^n + \mathbf{g}_l^n \qquad (A.4)$$

The rate of n frame relative to e frame expressed in n frame is given y $\boldsymbol{\omega}_{en}^n$. The Earth rate relative to the i frame is ω_{ie}^n.

The position is given in form of the altitude h above the Earth's surface and the $\mathbf{C}_n^e$ direction cosine matrix defining angular orientation between the navigation frame, n frame, and the Earth fixed frame, e frame, from which the latitude and longitude can be extracted.

The algorithm was developed: by using the trapezoidal integration of the velocity. The form can be represented by the continuous differential equation forms as follows:

$$\dot{h} = \mathbf{u}_{zn}^n V_e^n \qquad (A.5)$$

$$\mathbf{u}_{zn}^n = [0 \ \ 0 \ \ -1]^{\mathrm{T}} \qquad (A.6)$$

$$\dot{\mathbf{C}}_n^e = \mathbf{C}_n^e(\boldsymbol{\omega}_{en}^n \mathrm{x}) = \mathbf{C}_n^e \boldsymbol{\Omega}(\boldsymbol{\omega}_{en}^n) \qquad (A.7)$$

References

[1] BAR-ITZHACK,I.Y. *Navigation Computation in Terrestrial Stapdown Inertial navigation Systems*, IEEE Trans. on Aerospace and Electronic Systems, November 1977, Vol.AES-13, No.6, pp.679-689.

[2] DOROBANTU,R. *Simulation des Verhaltens einer Low-Cost Strapdown-IMU unter aborbedingungen,* Techische Univesitat Munchen, 1999, IAPG/FESG No.6, pp.1-17.

[3] SAVAGE,P.G. *Strapdown Inertial Navigation Integration Algorithm, Part 1: Attitude Algorithms,* Journal of Guidance, Control and dynamics, January-February 1998, Vol.21, No.1, pp.19-28.

[4] SAVAGE,P.G. *Strapdown Inertial Navigation Integration Algorithm, Part 2: Velocity and Position Algorithms* , Journal of Guidance, Control and dynamics, March-April 1998, Vol.21, No.2, pp.208-221.

[5] WALDMANN,J. , *XIV* Congress Brasileiro de Automatica, September 2002, pp.2361-2366.

[6] AHMED,S.M. *New Algorithms for the Determination of Kinematical Parameters of Strapdown Inertial Navigation Systems with Random Inputs,* Ph.D Thesis, Faculty of Mechanical Engineering, Belgrade, 2004.

[7] TITTERTON,D.H. and WESTON,J.L. *Strapdown Inertial Navigation Technology,* Peter Peregrinus Ltd on behalf of the Institute of Electrical Engineers, London, 1997.

Received: 01.03.2005.

Uporedna analiza različitih numeričkih metoda besplatformnih inercijalnih navigacionih sistema

Izvršen je niz numeričkih eksperimenata radi provere različitih metoda rešavanja navigacionih jednačina besplatformnih inercijalnih navigacionih sistema: Runge-Kutta metoda sa diskretizacijom mernih signala i navigacioni algoritam sa tri brzine računanja kinematičkih veličina. Prezentiran je stohastički simulator koji generiše izlazne veličine senzora inercijalne merne jedinice primenom zadatih funkcija od vremena za Ojlerove uglove i komponente brzine. Generisane ugaone brzine i specifične sile korišćene su kao ulazne veličine za sve tri metode integracije navigacionih jednačina. Pokazano je da je odnos apsolutnih grešaka kinematičkih veličina dobijenih navigacionim algoritmom i Runge-Kutta metodom sa diskretizacijom mernih signala smanjen kod sistema sa belim šumom u poređenju sa istom veličinom idealnog sistema (bez šuma).

Ključne reči: **mehanika leta, navigacija, navigacioni sistem, inercijalno navođenje, numeričke metode, metoda Runge-Kutta.**

Analyse comparée des différentes méthodes numériques des systèmes de navigation inertiels sans plate - formes

Cet article résume une série d'essais numériques effectués dans le but de vérifier les différentes méthodes de solutions des équations de navigation chez les systèmes de navigation inertiels sans plate - formes. La méthode Runge-Kutta avec discrétisation des signaux de mesure et l'algorithme de navigation à trois vitesses de calcul des valeurs cinématiques. Un simulateur stochastique qui produit les valeurs sortantes des sen sors de l'unité de mesure initiale par l'application des fonctions données du temps pour les angles d'Ojler et les composantes de vitesse est également présenté. Les vitesses d'angle produites et les forces spécifiques sont utilisées comme les valeurs d'entrée pour toutes les trois méthodes d'intégration des équations de navigation. On a démontré que le rapport des erreurs absolues des valeurs cinématiques, obtenues au moyen d'algorithme de navigation et la méthode Runge-Kutta avec la discrétisation des signaux mesurés, est diminué chez les systèmes avec le bruit blanc en comparaison avec la même valeur du système idéal (sans bruit).

Mots clés: **mécanique de vol, navigation; système de navigation, guidage inertiel, méthodes numériques, méthode Runge-Kutta.**

Сравнительный анализ различных численных методов безплощадных инерциальных навигационных систем

Здесь выполнен ряд численных экспериментов ради проверки различных методов решения навигационных уравнений безплощадных инерциальных навигационных систем: метод Рунге-Кутта со выбором дискретных данных измеряемых сигналов и навигационный алгоритм со три скорости вычисления кинематических величин. Здесь показано стохастическое моделирующее устройство, решающее выходные величины чувствительного элемента инерциальной измерительной единицы применением заданных функций от времени для углов Ойлера и для составляющей скорости. Решенные угловые скорости и удельные силы пользованы в роли входных величин для всех трех методов интеграции навигационных уравнений. Здесь тоже показано, что отношение абсолютных ошибок кинематических величин, полученых навигационным алгоритмом и методом Рунге-Кутта со выбором дискретных данных измеряемых сигналов, уменьшено у систем со "белым" шумом в сравнении с такой же величиной идеальной системы (без шума).

Ключевые слова: механика полета, навигация, навигационная система, инерциальное наведение, определение положения, численные мытоды, метод Рунге-Кутта.