

ТЕОРІЯ ТА ПРАКТИКА НАВІГАЦІЙНИХ ПРИЛАДІВ І СИСТЕМ

UDC 629.051

PRECISION QUATERNION BASED ONE STEP STRAPDOWN ATTITUDE ALGORITHM¹Yuriy Lazarev, ¹Vadym Avrutov, ¹Pavlo Myronenko, ²Sergiy Davydenko, ¹Oleksandr Sapegin¹National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»,
Kyiv, Ukraine²Yuzhnoye State Design Office, Dnipro, UkraineE-mail: mironenkops46@gmail.com

Introduction. Calculative algorithms of Strapdown inertial navigation systems (SINS) can be divided on navigation algorithms, which transform accelerometers output signals into local coordinates and attitude algorithms, which transform gyroscopes output signals into vehicle angular attitude [1]. Wherein, navigation task solution requires double integration of acceleration and attitude task – integration of kinematic attitude equation, related measured object angular velocity with attitude parameters.

Paper considered of autonomous position determination methods based on vehicle angular velocity information without acceleration measurement. Thus, attitude algorithms are considered only.

Paper researched the errors of algorithm based on quaternion attitude equation, moreover algorithm error drifts were accepted as a main accuracy characteristic

The main part. Algorithm researched by imitation modeling of vehicle's computer with SINS attitude algorithm. The main task of modeling is defining depends between algorithm drift and sensor's call step in cases of different frequencies and amplitudes of base angular oscillations. It was researched four one-step algorithms: reverse, based on modified Euler method; Picard method with two successive approximations and the new author's algorithm which combines formulas of first two algorithms. It was studied depends of algorithm drift and faze shift between two orthogonal axes oscillations. It was shown, the biggest drift values are obtained in case of base conning movement. It was made the modeling researches of algorithm drift amplitudes relatively to sensor sample steps and oscillation frequencies in dimensionless form. It was shown, substantial increase new algorithm accuracy compared to other researched.

Conclusions. The algorithm drift accuracy of new algorithm in 2600 times exceeds the revers algorithm. Small modification of one-step algorithm allowed increase accuracy in few orders, almost without computing increase. Received results allows to expend attitude algorithms application area and prognose their accuracy with different base movement.

Keywords: strapdown inertial navigation system; attitude algorithm; one-step algorithm; algorithm drift.

Introduction

There are known six types of kinematic parameters [2]:

- Euler-Krylov rotation angles;
- directional cosines between navigation and base frames;
- Euler orientation vector;
- Gibbs and Rodrigo rotation vectors which are variants of orientation vector;
- quaternions of rotation [3].

Each of kinematic parameters conform deferential kinematic equation – attitude equation. Euler-Krylov rotation angles are not commonly used in Strapdown inertial navigation systems (SINS), because they could annihilate in some angle values. More acceptable is Poisson deferential equation, which defines directional cosine matrix. But the number of kinematic parameters becomes nine, which increases computations. The most optimal for practical

application is quaternion kinematic equation. Kinematic equation, based on Euler orientation vector, known as Bortz equation [4]. SINS attitude algorithm synthesis based on Picard successive approximations method for quaternion attitude equation presented in [5, 6]. Miller developed his own attitude algorithm method in 1983. It was based on the Teylor series expansion of Bortz equation. This method is commonly used now in western [8-18] and eastern [19-24] papers as well. Significant contribution of Chinese scientists should be noted in attitude algorithms development for last years [25-30].

SINS algorithm researching [31, 32] showed: mean errors values increasing, in case of some angular motions, with constant speed, which called algorithm error's drift speed or "algorithm drift". Wherein, algorithm drift, in some cases, could be so high, that in a few minutes its value becomes much bigger than other errors. Since that time, the error which depends of

algorithm drift defines the algorithm accuracy. Therefore, algorithm drift errors can be used as main SINS algorithm accuracy characteristic.

Problem statement

Attitude algorithms based on measurement of angular velocity vector $\boldsymbol{\omega}_k = \boldsymbol{\omega}(t_k)$ or quasi-coordinate vector \mathbf{q}_k in discrete time moment separated by sensor sample step h : $\mathbf{q}_k = \int_{t_{k-h}}^{t_k} \boldsymbol{\omega}_k dt$.

The algorithm error researching is easy to make by computer simulation in Matlab which allows to imitate data computing on-board by SINS.

It was studied the error drift of ψ angle, assuming the base movement, when this angle remains unchanged (and equal zero). At the same time ϑ and γ angles makes synchronic oscillations with frequency ω , same amplitude a_m and phase shift ε . This ensures the best conditions to algorithm drift appearance. Such type of base moving is customary to call “coning” in special inertial navigation works.

The final result of computer simulation is defining the dependence between attitude integration algorithm errors and sensors polling rate with different base angular oscillation amplitudes and frequencies.

One-step revers algorithm

One-step revers algorithm is commonly used in practice currently. It calculates quaternion values on n -th step by known value on $(n-1)$ -th step and measured angular rate at the same times by formulas:

$$\begin{aligned} \lambda_0^n &= \lambda_0^{n-1} - 0,25\tau_h (\omega_{x1}^{n-1} \cdot \lambda_1^{n-1} + \omega_{y1}^{n-1} \cdot \lambda_2^{n-1} + \omega_{z1}^{n-1} \cdot \lambda_3^{n-1}), \\ \lambda_1^n &= \lambda_1^{n-1} + 0,25\tau_h (\omega_{x1}^{n-1} \cdot \lambda_0^{n-1} + \omega_{z1}^{n-1} \cdot \lambda_2^{n-1} - \omega_{y1}^{n-1} \cdot \lambda_3^{n-1}), \\ \lambda_2^n &= \lambda_2^{n-1} + 0,25\tau_h (\omega_{y1}^{n-1} \cdot \lambda_0^{n-1} + \omega_{x1}^{n-1} \cdot \lambda_3^{n-1} - \omega_{z1}^{n-1} \cdot \lambda_1^{n-1}), \\ \lambda_3^n &= \lambda_3^{n-1} + 0,25\tau_h (\omega_{z1}^{n-1} \cdot \lambda_0^{n-1} + \omega_{y1}^{n-1} \cdot \lambda_1^{n-1} - \omega_{x1}^{n-1} \cdot \lambda_2^{n-1}), \\ \lambda_0^n &= \lambda_0^n + 0,25\tau_h (\omega_{z1}^n \cdot \lambda_0^n + \omega_{y1}^n \cdot \lambda_1^n - \omega_{x1}^n \cdot \lambda_2^n), \\ \lambda_1^n &= \lambda_1^n + 0,25\tau_h (\omega_{y1}^n \cdot \lambda_0^n + \omega_{x1}^n \cdot \lambda_3^n - \omega_{z1}^n \cdot \lambda_2^n), \\ \lambda_2^n &= \lambda_2^n + 0,25\tau_h (\omega_{x1}^n \cdot \lambda_0^n + \omega_{z1}^n \cdot \lambda_1^n - \omega_{y1}^n \cdot \lambda_3^n), \\ \lambda_3^n &= \lambda_3^n - 0,25\tau_h (\omega_{x1}^n \cdot \lambda_1^n + \omega_{y1}^n \cdot \lambda_2^n + \omega_{z1}^n \cdot \lambda_3^n), \end{aligned} \quad (1)$$

where, $\lambda_0^{n-1}, \lambda_1^{n-1}, \lambda_2^{n-1}, \lambda_3^{n-1}$ – quaternion values on $(n-1)$ -th calculating step;

$\omega_{x1}^{n-1}, \omega_{y1}^{n-1}, \omega_{z1}^{n-1}$ – angular rate projections on

$(n-1)$ -th calculating step;

$\lambda_0^n, \lambda_1^n, \lambda_2^n, \lambda_3^n$ – quaternion values on n -th calculating step;

$\omega_{x1}^n, \omega_{y1}^n, \omega_{z1}^n$ – angular rate projections on n -th calculating step.

According to Eq.1, quaternion Λ elements initially calculates using angular rate values at the beginning of the sample (“straight run”), then quaternion Λ

elements defines in revers sequence by using angular rate values at the end of the sample (“revers run”). As a result, previous calculating step quaternion $\bar{\Lambda}^{n-1} = \bar{\Lambda}(t^{n-1})$ transforms into current calculating step quaternion $\bar{\Lambda}^n = \bar{\Lambda}(t^n)$.

Let write these formulas compact, using quaternions.

We denote:

– quaternion values at the beginning and at the end of polling rate:

$$\Lambda_0 = \lambda_0^{(0)} + \mathbf{i} \cdot \lambda_1^{(0)} + \mathbf{j} \cdot \lambda_2^{(0)} + \mathbf{k} \cdot \lambda_3^{(0)};$$

$$\Lambda_1 = \lambda_0^{(1)} + \mathbf{i} \cdot \lambda_1^{(1)} + \mathbf{j} \cdot \lambda_2^{(1)} + \mathbf{k} \cdot \lambda_3^{(1)};$$

– angular rate-quaternion values at the beginning and at the end of polling rate:

$$\boldsymbol{\omega}_0 = 0 + \mathbf{i} \cdot \omega_{x1}^{(0)} + \mathbf{j} \cdot \omega_{y1}^{(0)} + \mathbf{k} \cdot \omega_{z1}^{(0)};$$

$$\boldsymbol{\omega}_1 = 0 + \mathbf{i} \cdot \omega_{x1}^{(1)} + \mathbf{j} \cdot \omega_{y1}^{(1)} + \mathbf{k} \cdot \omega_{z1}^{(1)}.$$

Let write first four equations from (1) of “straight run” in quaternion form:

$$\Lambda^* = \Lambda_0 + \frac{h}{4} \Lambda_0 \circ \boldsymbol{\omega}_0,$$

where Λ^* – intermediate rate quaternion, h – polling rate, \circ – the quaternion product sign.

Similarly, write last four equations from (1) of “revers run” in quaternion form:

$$\Lambda_1 = \Lambda^* + \frac{h}{4} \Lambda^* \circ \boldsymbol{\omega}_1.$$

These two formulas can be combined into one by excluding of intermediate quaternion Λ^* :

$$\Lambda_1 = \Lambda_0 + \frac{h}{4} \Lambda_0 \circ (\boldsymbol{\omega}_0 + \boldsymbol{\omega}_1) + \frac{h^2}{16} (\Lambda_0 \circ \boldsymbol{\omega}_0) \circ \boldsymbol{\omega}_1. \quad (2)$$

Algorithm, presented in Eq.2, was called Revers1h.

The Matlab code of Revers1h presented below.

```
function Q1 = Revers1h(t,h,Q0)
[psi,tet,ga,ux0,uy0,uz0]=UgDvObj_laz(t);
om0=[0 ux0 uy0 uz0];
[psi,tet,ga,ux1,uy1,uz1]=UgDvObj_laz(t+h);
om1=[0 ux1 uy1 uz1];
Qp=Q0+h/4*multQuat(Q0,om0);
Q1=Qp+h/4*multQuat(Qp,om1);
```

Here used two additional procedures:

– UgDvObj_laz, which calculates current base angular movement;

– multQuat, which made quaternion product:

```
function q=multQuat(q1,q2)
q1s=q1(1); q2s=q2(1);
q1v=q1(2:4); q2v=q2(2:4);
qs=q1s*q2s-q1v*q2v';
qv=q2s*q1v+q1s*q2v+cross(q1v,q2v);
q=[qs,qv];
```

Algorithm RK21 of Runge-Kutta method

One of the most commonly used integration method for mathematical and engineering research is Runge-Kutta algorithm. It could be applied to different accuracy orders or step size.

One-step Runge-Kutta integration method of second-order accuracy was applied to Matlab in algorithm RK21:

```
function [tout,yout]=RK21(Zpfun,h,t,y)
k1=feval(Zpfun,t,y);
k2=feval(Zpfun,t+h,y+h*k1);
F=(k1+k2)/2;
yout=y+h*F; tout=t+h;
```

If RK21 applies to quaternion attitude equation, Zpfun parameter need to assign a name of right part quaternion attitude equation calculating function, for example ZPfun ='KwUrAtt'. Matlab code of KwUrAtt presented below:

```
function z = KwUrAtt(t,y)
[psi,tet,ga,ux,uy,uz]=UgDvObj_laz(t);
om=[0 ux uy uz];
z=multQuat(y,om)/2;
```

Consider the operation of this functions set.

First calling $k1=feval(Zpfun,t,y)$ to the the right parts function of quaternion equation, calculates quaternion

$$k_1 = \frac{1}{2} \Lambda_0 \circ \omega_0 .$$

Second calling $k2=feval(Zpfun,t+h,y+h*k1)$ allows to calculate quaternion

$$k_2 = \frac{1}{2} (\Lambda_0 + \frac{h}{2} \Lambda_0 \circ \omega_0) \circ \omega_1 .$$

Operator $F=(k1+k2)/2$ defines a half-sum of this results:

$$F = \frac{1}{2} \left[\frac{1}{2} \Lambda_0 \circ \omega + \frac{1}{2} (\Lambda_0 + \frac{h}{2} \Lambda_0 \circ \omega_0) \circ \omega_1 \right] = \\ = \frac{1}{4} \Lambda_0 \circ (\omega_0 + \omega_1) + \frac{h}{8} (\Lambda_0 \circ \omega_0) \circ \omega_1 .$$

Finally, operator $yout=y+h*F$ defines a value of new quaternion rate after polling rate:

$$\Lambda_1 = \Lambda_0 + \frac{h}{4} \Lambda_0 \circ (\omega_0 + \omega_1) + \frac{h^2}{8} (\Lambda_0 \circ \omega_0) \circ \omega_1 . \quad (3)$$

The results almost coincide with revers algorithm (2). The different only in value of third coefficient.

Algorithm Pic1h2tQuatOm of Picard method

The algorithm of quaternion rate $\Lambda_1 = \Lambda(t_{n+1})$ defining at the time moment $t = t_{n+1} = t_n + h$ (where h – integration step) by it known value $\Lambda_0 = \Lambda(t_n)$ at the previous time moment $t = t_n$ is

$$\Lambda_1 = \Lambda_0 \circ N_1 , \quad (4)$$

where N_1 is solution of equation

$$N_1(t) = 1 + \frac{1}{2} \int_{t_0}^t N(\xi) \circ \omega(\xi) d\xi , \quad (5)$$

at the time moment $t = t_{n+1} = t_n + h$.

Eq. 5 can be solved by Picard method as successive approximations.

First approximation

In the zero approximation will be $N_{(0)}(t) = 1$. Using this to the right part of Eq.3 get solution in first approximation:

$$N_{(1)}(t) = 1 + \frac{1}{2} \int_{t_n}^t N_{(0)}(\xi) \circ \omega(\xi) d\xi = 1 + \frac{1}{2} \int_{t_n}^t \omega(\xi) d\xi .$$

Second approximation

$$N_{(2)}(t) = 1 + \frac{1}{2} \int_{t_n}^t N_{(1)}(\xi) \circ \omega(\xi) d\xi = \\ = 1 + \frac{1}{2} \int_{t_n}^t \left(1 + \frac{1}{2} \int_{t_n}^{\xi} \omega(\eta) d\eta \right) \circ \omega(\xi) d\xi = \\ = 1 + \frac{1}{2} \int_{t_n}^t \omega(\xi) d\xi + \frac{1}{4} \int_{t_n}^t \left(\int_{t_n}^{\xi} \omega(\eta) d\eta \right) \circ \omega(\xi) d\xi .$$

Refine solutions

In case of **one-step** algorithm, there are known two values of angular values ω , measured at the beginning and at the end of polling rate. Denote ω_0 and ω_1 values of this vector at the beginning and at the end of integration step H ($t = t_n$; $\tau = 0$ and $t = t_{n+1}$; $\tau = h$).

In this case, angular rate can be approximated by linear function:

$$\omega(\tau) = \omega_0 + 2\varepsilon \cdot \tau .$$

At the end of polling rate (and integration as well) $\tau = h$ measured value of angular rate will be:

$$\omega_1 = \omega_0 + 2\varepsilon \cdot h .$$

Thus, unknown vector coefficient defines like

$$\varepsilon = \frac{1}{2h} (\omega_1 - \omega_0) .$$

Since the required function $\omega(\tau)$ defines by expression

$$\omega = \omega_0 + (\omega_1 - \omega_0) \cdot \zeta ,$$

where dimensionless time value presented as $\zeta = \frac{\tau}{h}$.

First approximation

Let find the value of integral

$$\int_{t_n}^t \omega(\xi) d\xi = \int_0^{\tau} (\omega_0 + 2\varepsilon \cdot \tau) d\tau = \omega_0 \tau + \varepsilon \cdot \tau^2 .$$

Now Eq.4 in first approximation can be written as:

$$N_{(1)}(\tau) = 1 + \frac{1}{2} [\omega_0 \tau + \varepsilon \cdot \tau^2] =$$

$$= 1 + \frac{1}{2} \left[\boldsymbol{\omega}_0 \tau + \frac{1}{2h} (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_0) \cdot \tau^2 \right] =$$

$$= 1 + \frac{h}{2} \left[\boldsymbol{\omega}_0 \zeta + (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_0) \cdot \frac{\zeta^2}{2} \right].$$

At the end of integration sample $\zeta = 1$, thus final formula of algorithm will be

$$N_1 = 1 + \frac{h}{4} (\boldsymbol{\omega}_1 + \boldsymbol{\omega}_0).$$

Second approximation

First of all, we need to calculate double integral:

$$\int_{t_n}^t \left(\int_{t_n}^{\xi} \boldsymbol{\omega}(\eta) d\eta \right) \circ \boldsymbol{\omega}(\xi) d\xi =$$

$$= \int_0^{\tau} (\boldsymbol{\omega}_0 \tau + \boldsymbol{\varepsilon} \cdot \tau^2) \circ (\boldsymbol{\omega}_0 + 2\boldsymbol{\varepsilon} \cdot \tau) d\tau =$$

$$= \boldsymbol{\omega}_0 \circ \boldsymbol{\omega}_0 \frac{\tau^2}{2} + \boldsymbol{\omega}_0 \circ \boldsymbol{\varepsilon} \frac{2\tau^3}{3} + \boldsymbol{\varepsilon} \circ \boldsymbol{\omega}_0 \frac{\tau^3}{3} + \boldsymbol{\varepsilon} \circ \boldsymbol{\varepsilon} \frac{\tau^4}{2}.$$

Considering that, quaternion product of two quaternion vectors is

$$\mathbf{a} \circ \mathbf{b} = (0 + \mathbf{a}) \circ (0 + \mathbf{b}) = -\mathbf{a} \cdot \mathbf{b} + \mathbf{a} \times \mathbf{b},$$

i.e. difference between their vector's product and scalar product.

Thus,

$$\boldsymbol{\omega}_0 \circ \boldsymbol{\omega}_0 = -\boldsymbol{\omega}_0^2; \quad \boldsymbol{\omega}_0 \circ \boldsymbol{\varepsilon} = \boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon} - \boldsymbol{\omega}_0 \cdot \boldsymbol{\varepsilon};$$

$$\boldsymbol{\varepsilon} \circ \boldsymbol{\omega}_0 = -\boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon} - \boldsymbol{\omega}_0 \cdot \boldsymbol{\varepsilon}; \quad \boldsymbol{\varepsilon} \circ \boldsymbol{\varepsilon} = -\boldsymbol{\varepsilon}^2.$$

Finally get

$$\int_{t_n}^t \left(\int_{t_n}^{\xi} \boldsymbol{\omega}(\eta) d\eta \right) \circ \boldsymbol{\omega}(\xi) d\xi = -\boldsymbol{\omega}_0^2 \frac{\tau^2}{2} +$$

$$+ (\boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon} - 3\boldsymbol{\omega}_0 \cdot \boldsymbol{\varepsilon}) \frac{\tau^3}{3} - \boldsymbol{\varepsilon}^2 \frac{\tau^4}{2} =$$

$$= \frac{h^2 \zeta^2}{2} \left\{ -\boldsymbol{\omega}_0^2 + (\boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon} - 3\boldsymbol{\omega}_0 \cdot \boldsymbol{\varepsilon}) \frac{2h\zeta}{3} - \boldsymbol{\varepsilon}^2 (h\zeta)^2 \right\}.$$

So, the second approximation solution will be

$$N_{(2),n+1} = 1 + \frac{h}{4} (\boldsymbol{\omega}_1 + \boldsymbol{\omega}_0) + \frac{h^2}{8} \times$$

$$\times \left\{ -\boldsymbol{\omega}_0^2 + \frac{2}{3} (\boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon} - 3\boldsymbol{\omega}_0 \cdot \boldsymbol{\varepsilon}) h - \boldsymbol{\varepsilon}^2 h^2 \right\}.$$

Separating scalar part from vector's, we get

$$N_{(2),n+1} = 1 - \frac{h^2}{8} (\boldsymbol{\omega}_0^2 + 2h\boldsymbol{\omega}_0 \cdot \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^2 h^2) +$$

$$+ \frac{h}{4} \left[(\boldsymbol{\omega}_1 + \boldsymbol{\omega}_0) + \frac{h}{3} (\boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon}) \right] =$$

$$= 1 - \frac{h^2}{8} (\boldsymbol{\omega}_0 + h\boldsymbol{\varepsilon}) \cdot (\boldsymbol{\omega}_0 + h\boldsymbol{\varepsilon}) +$$

$$+ \frac{h}{4} \left[(\boldsymbol{\omega}_1 + \boldsymbol{\omega}_0) + \frac{h}{3} (\boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon}) \right] =$$

$$= \left\{ 1 - \frac{h^2}{8} |(\boldsymbol{\omega}_1 + \boldsymbol{\omega}_0)|^2 \right\} + \frac{h}{4} \left[(\boldsymbol{\omega}_1 + \boldsymbol{\omega}_0) + \frac{h}{3} (\boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon}) \right].$$

The final formula of the algorithm is

$$\Lambda_1 = \Lambda_0 \cdot \left\{ 1 - \frac{h^2}{8} |(\boldsymbol{\omega}_1 + \boldsymbol{\omega}_0)|^2 \right\} + \frac{h}{4} \Lambda_0 \circ (\boldsymbol{\omega}_1 + \boldsymbol{\omega}_0) +$$

$$+ \frac{h^2}{12} \Lambda_0 \circ (\boldsymbol{\omega}_0 \times \boldsymbol{\varepsilon}). \tag{6}$$

Computer simulation of the algorithms

The computer simulation software was created for studying the properties of created algorithms. Software simulates information processing process by vehicle's computer using all received methods.

Some of simulation results presented on Fig. 1 – 3. Test moving parameters were accepted as:

- oscillation amplitude of ϑ and γ angles is $a_m = 0,1$ rad;
- oscillation frequency of this angles is $\omega = 1$ rad/sec;
- phase shift between ϑ and γ angles is $\varepsilon = 90^\circ$.

The main simulation calculated value is error drift of angle ψ defining, which shouldn't move at all.

Studding of the results obtained to the following conclusions:

1) the biggest drift value has RK21 algorithm; it about twice bigger then Revers1h; the most accurate is Pic1h2tQuatOm algorithm, which showed error drift more than 20 times less then Revers1h;

2) all algorithms have second-order accuracy by error drifts;

3) algorithms Revers1h and Pic1h2tQuatOm showed continuously amplitude error incising; Algorithm RK21 showed steady error oscillations by base angle oscillations;

Errors oscillation instability can be eliminated by quaternion rating operation. The simulation result with adding of quaternion rating operation showed on Fig. 4 – 6.

As we see, quaternion rating operation leads to full elimination of error unsteady oscillation. At the same time, it does not change errors of Runge-Kutta method.

Dependence of error drift and phase shift

Influence of base oscillation phase shift value on algorithms error received by computer simulation is presented on Fig. 7. The amplitude frequency of oscillation are $a_m = 0,1$ rad and $\omega = 1$ rad/s;

It should be noticed, that algorithm's RK21 and Revers1h error drifts for any integration step values changes anti-phase to base oscillation phase shift. This circumstance could be used for developing high-accurate algorithm.

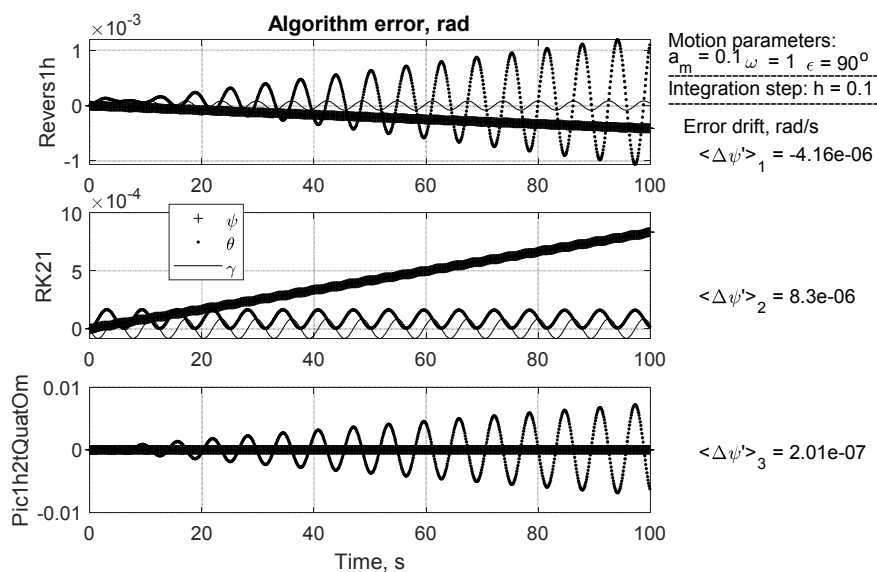


Fig. 1. Integration step 0,1 s

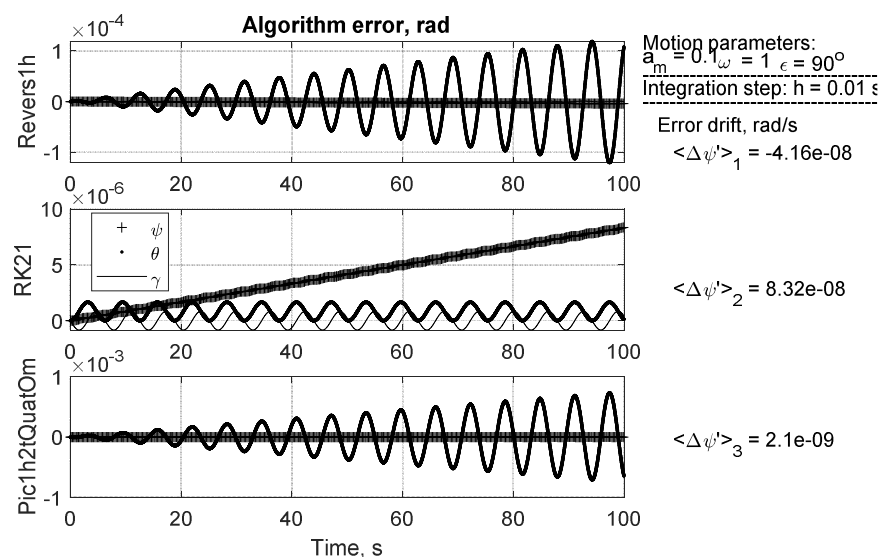


Fig. 2. Integration step 0,01 s

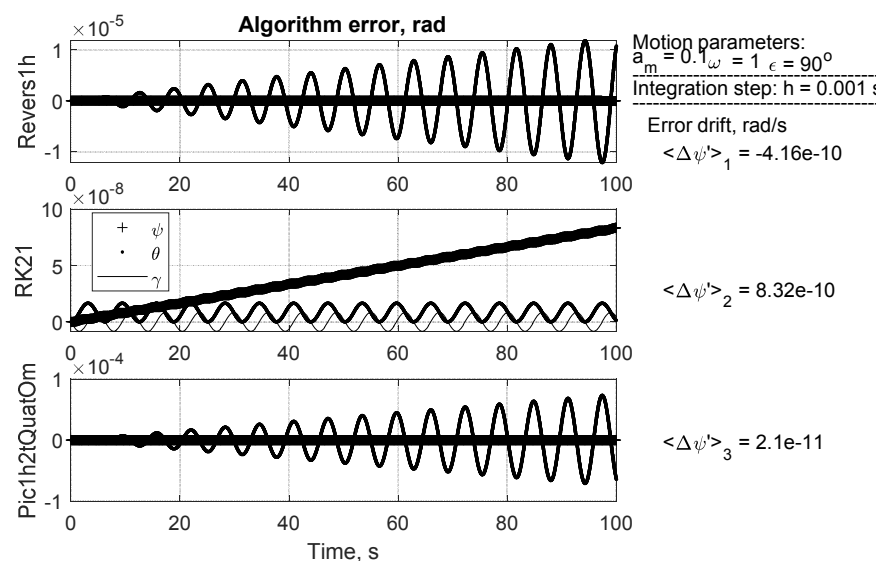


Fig. 3. Integration step 0,001 s

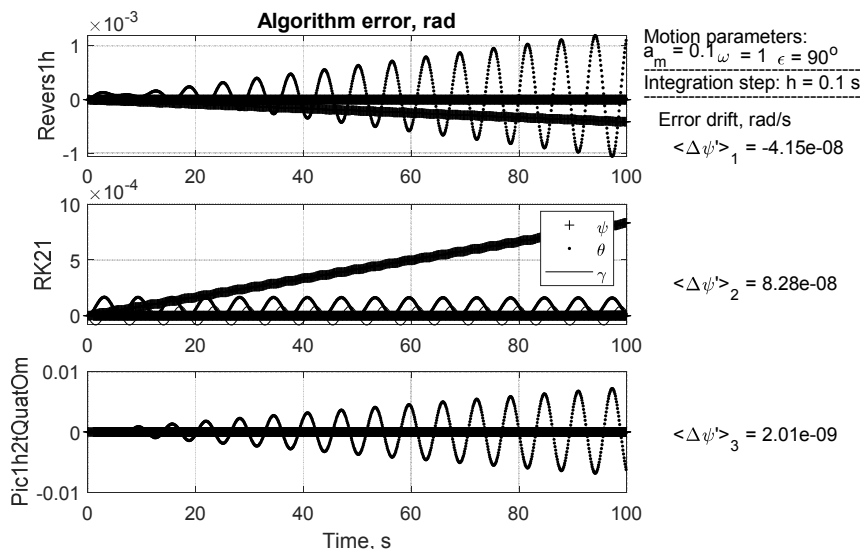


Fig. 4. Integration step 0,1 s. Quaternion rating

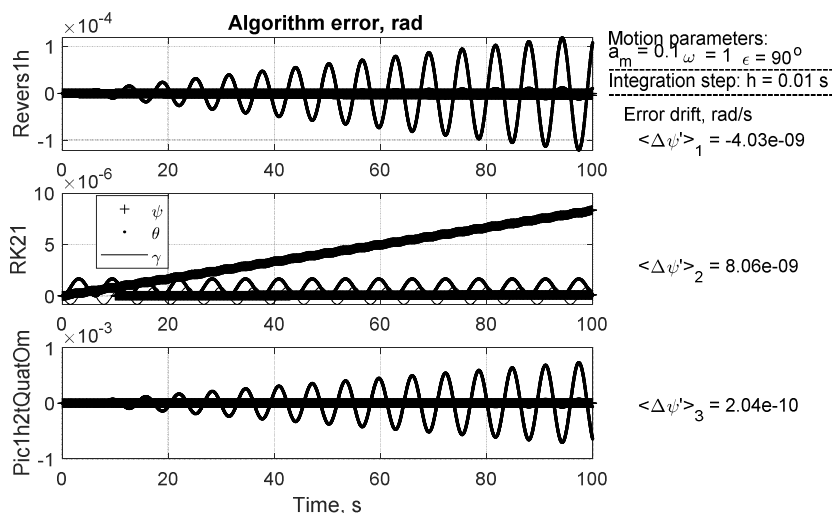


Fig. 5 Integration step 0,01 s. Quaternion rating

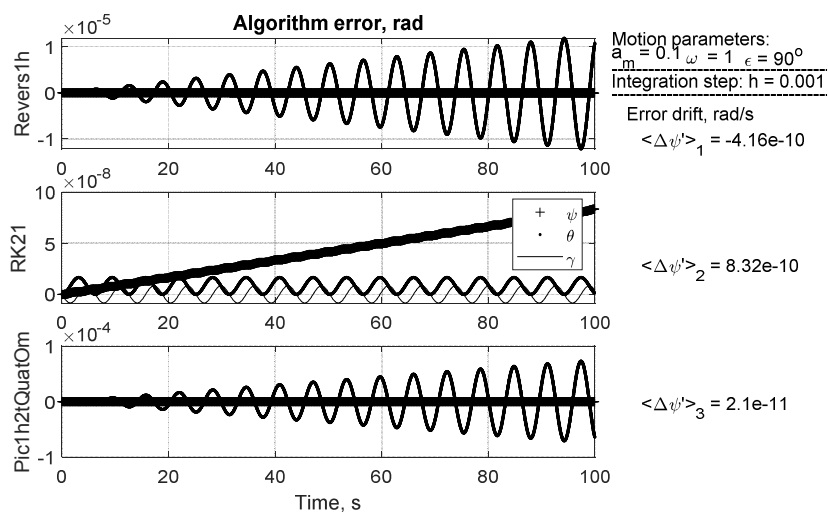


Fig. 6. Integration step 0,001 s. Quaternion rating

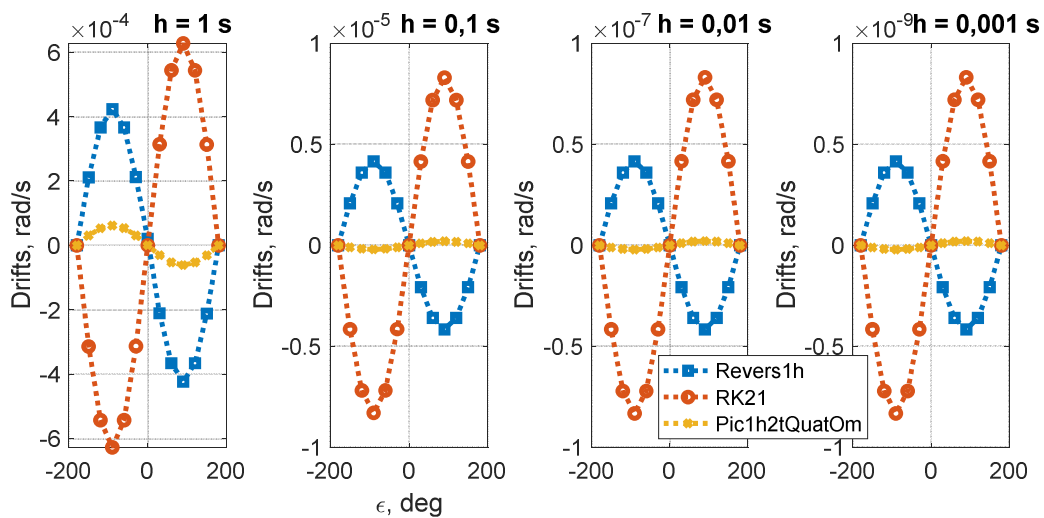


Fig. 7. Dependences of algorithm error drifts and phase shifts

Dependence of maximum error drift and sensors polling rate

Computer simulations showed, algorithm error drifts almost do not depend on frequencies and amplitudes, which only changes the scale. If we change considering from absolute drift values to their reality to value $\omega \cdot a^2$, and use as argument not a polling rate but a frequency parameter $\mu = \omega h$, i.e. study relation

$$\delta(\mu), \text{ where } \delta = \frac{1}{\omega \cdot a^2} \left| \left\langle \frac{d\Delta\psi}{dt} \right\rangle_{\max} \right| - \text{dimensionless}$$

maximum drift, this relations stay constant in any base oscillations or polling rate.

For described algorithms, corresponding dimensionless dependencies presented on Fig.8. Approximately, they can be presented in analytical form like:

- for Revers1h — $\delta = 0.042 \cdot \mu^2$;
- for RK21 — $\delta = 0.083 \cdot \mu^2$;
- for Pic1h2tQuatOm — $\delta = 0.0021 \cdot \mu^2$.

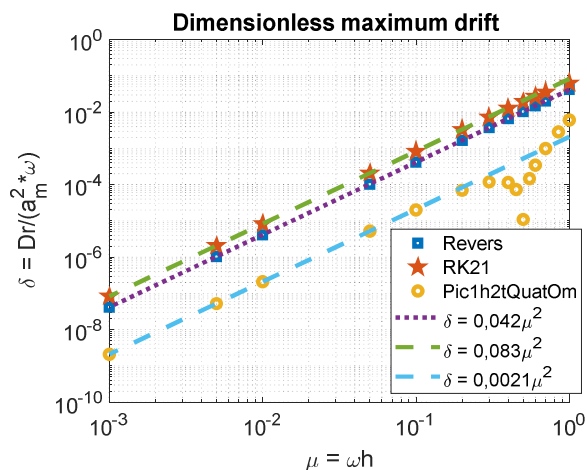


Fig. 8. Dependence of dimensionless drift and frequency parameter

We can see from Fig. 8, that Revers algorithm is twice accurate than one-step RK21, and Pic1h2tQuatOm algorithm is 20 times more accurate than Revers.

High-accurate algorithm

Studying phase characteristics (Fig. 7) of presented algorithms, it can be noticed, that reverse algorithm and Runge-Kutta method both changes harmonically with anti-phase for all polling rate values. It allows us to create combination of these algorithms to significant decrease of maximum error drift.

The formula of corresponding algorithm is

$$\Lambda_1 = \Lambda_0 + \frac{h}{4} \Lambda_0 \circ (\omega_0 + \omega_1) + \frac{h^2}{12} (\Lambda_0 \circ \omega_0) \circ \omega_1 \cdot (7)$$

A Matlab code of ExpRvrs1h function which implements Eq.7 presented below

```
function Q1 = ExpRvrs1h(t,h,Q0)
[psi,tet,ga,ux0,uy0,uz0]=UgDvObj_laz(t);
om0=[0 ux0 uy0 uz0];
psi,tet,ga,ux1,uy1,uz1]=UgDvObj_laz(t+h);
om1=[0 ux1 uy1 uz1];
Qp=Q0+h/4*multQuat(Q0,om0+om1);
Q1=Qp+h^2/12*multQuat(multQuat(Q0,om0),om1);
```

The results of simulation algorithm (7) by function ExpRvrs1h presented on Fig. 9.

After simulations for couples polling rate values and changing the phase shift values from -180° to 180° , we found relation between drift of new algorithm and phase shift.

On Fig. 10 presented this dependence in four polling rates. They show:

- dependence is harmonic in all cases;
- maximum drift not always observed in phase shift 90° (i.e. in case of conning movement); in range

0.01 s > h > 0.001 s, maximum drift is on cases of phase shift $\epsilon \approx -60^\circ$ and $\epsilon \approx 120^\circ$.

More detail results of new algorithm maximum drift measuring presented on Fig. 10, 11.

Dependence of error drifts and polling rates presented on Fig. 11.

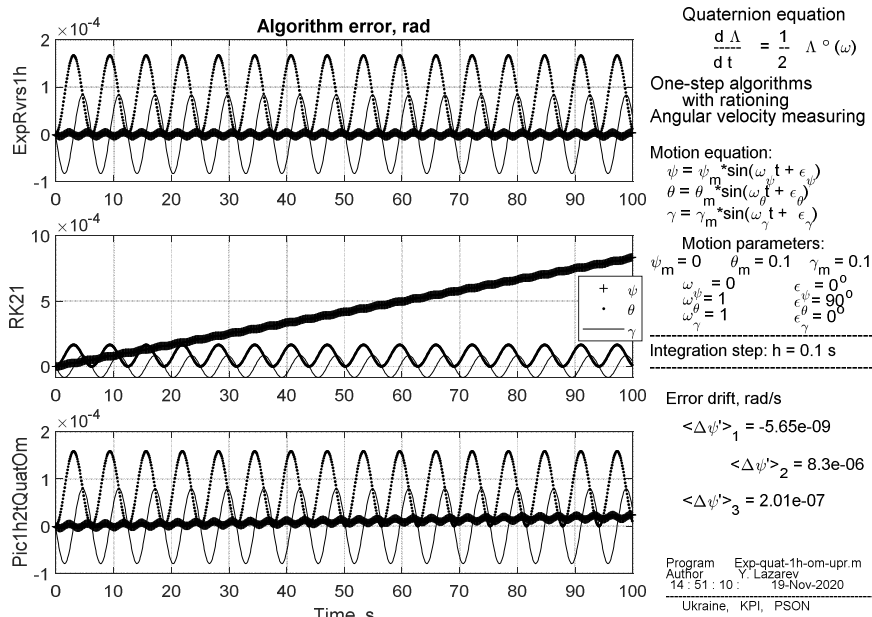


Fig. 9. Simulation of new ExpRvrs1h algorithm. Step $h = 0,1$ s

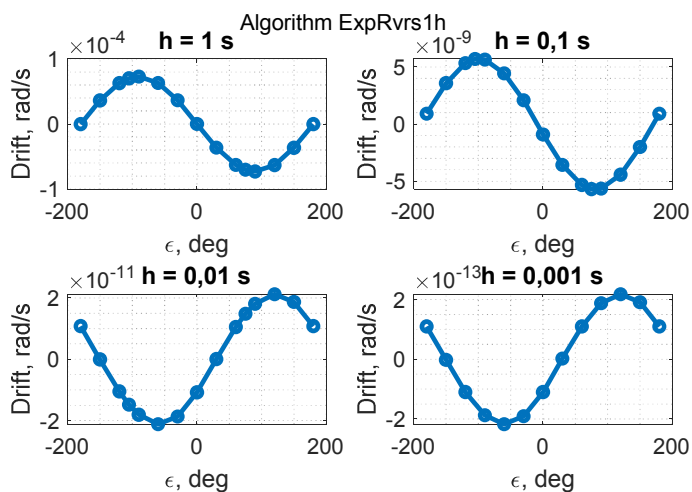


Fig. 10. Dependence of phase shift

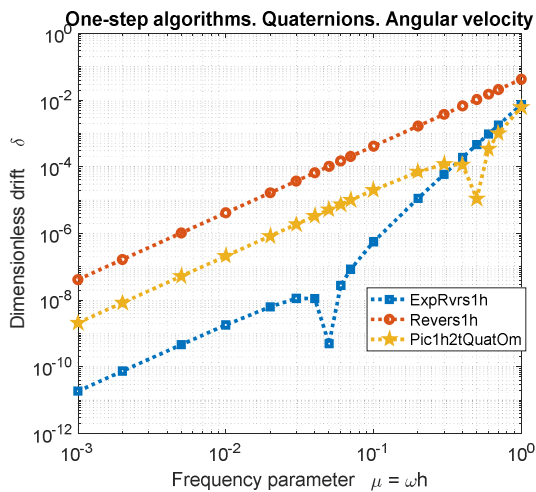


Fig. 11. Dependence of dimensionless drift and frequency parameter

From presented results we assume, that the new algorithm has next features:

- in frequency parameter range $1 > h > 0.05$ it has fourth-order (not second-order like others) accuracy; error drift can be approximately calculated by formula:

$$\delta = 7,3 \cdot 10^{-3} \mu^4;$$
- at $\mu = 1$ maximum drift value is similar to Picard's algorithm;
- in frequency parameter range $0.001 < h < 0.05$ it has (like other single-step algorithms) second-order

accuracy; algorithm drift can be calculated by formula:

$$\delta = 1,6 \cdot 10^{-7} \mu^2;$$

- at $\mu = 0,001$ maximum drift value almost 4 times less then Runge-Kutta method and twice less then Picard's algorithm;
- has "zero" maximum drift value at polling rate $\mu \approx 0,05$ s.

Conclusions

Research [31, 33] shows, if we could find two algorithms with the same error drift accuracy but op-

posite in sign in all faze shift range, it allows to increase accuracy on few orders without significant computation increasing. Computer simulations showed: error drift value of new developed algorithm is 2600 times exceeds similar accuracy of now-used revers algorithm.

Received results allows to expend the using area of strapdown attitude algorithms and to predict their accuracy in different body movement.

References

- [1] Ю. Литманович и Дж.Г. Марк, "Прогресс в разработке алгоритмов БИНС на Западе и Востоке в материалах Санкт-Петербургских конференций: обзор за десятилетие", на *X Санкт - Петербургская междунар. конф. по интегрированным навигационным системам*, СПб, 2003, с. 250–260.
- [2] Ю. Лазарев і О. Сапегін, *Алгоритми безплатформових інерціальних систем орієнтації. Прикладні методи синтезу і аналізу точності*, Київ, КПІ ім. Ігоря Сікорського, 2018.
<https://ela.kpi.ua/handle/123456789/25278>
- [3] С. Онищенко, *Применение гиперкомплексных чисел в теории инерциальной навигации. Автономные системы*, Киев, СССР: Наукова думка, 1983.
- [4] J. Bortz "A New Mathematical Formulation for Strapdown Inertial Navigation ", *IEEE Transactions on Aerospace and Electronic Systems*, vol. Aes-7, no. 7, 1971, pp. 61-66.
- [5] В. Бранец и И. Шмыглевский, *Применение кватернионов в задачах ориентации твердого тела*. Москва, СССР: Наука, 1973.
- [6] В. Бранец и И. Шмыглевский, *Введение в теорию безплатформенных инерциальных навигационных систем*. Москва, РФ: Наука, 1992.
- [7] R. Miller, "A new strapdown attitude algorithm", *Journal of Guidance, Control and Dynamics*, vol. 6, no. 4 , pp. 287-291, 1983.
- [8] P. Savage, "Inertial Navigation Integration Algorithms design Part1: Attitude algorithms", *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 19-28, 1998.
- [9] Дж. Марк и Д. Тазартес, "Конические алгоритмы, учитывающие неидеальность частотной характеристики выходных сигналов гироскопов", *Гироскопия и навигация*, № 1(28), с. 65-77, 2000.
- [10] J. Mark and D. Tazartes, "Tuning of coning algorithms to Gyro data frequency response characteristics", *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 4, pp. 641-647, 2001.
- [11] M. Pittelkau, "Rotation vector in attitude estimation", *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 6, pp. 855-860, 2003.
- [12] H. Musoff and J. Murphy, "Study of strapdown navigation attitude algorithms", *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 2, pp. 287-290, 1995.
- [13] M. Ignagni, "Efficient class of optimized coning compensation algorithms", *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 2, pp. 424-429, 1996.
- [14] J. G. Lee, G. Mark, D. Tazartes and Y. Yoon, "Extension of strapdown attitude algorithm for high-frequency base motion", *Journal of Guidance, Control and Dynamics*, vol. 13, no. 4, pp.738-743, 1990.
- [15] Y. Jiang and Y. Lin, "Improved strapdown coning algorithms ", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no.2, pp.486-489, 1992.
- [16] G. Chan and J. Kwang, "Formalized approach to obtaining optimal coefficients for coning algorithms", *Journal of Guidance, Control, and Dynamics*, vol.22, no.1, pp.165-168, 1999.
- [17] J. Crassidis, F. Markley, and Y. Cheng, "Survey of Nonlinear Attitude Estimation Methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, pp. 12-28, 2007.
- [18] F. Markley, "Fast Quaternion Attitude Estimation from Two Vector Measurements," *Journal of Guidance, Control and Dynamics*, vol. 25, pp. 411-414, 2002.
- [19] Д. Лебедев и А. Ткаченко, *Системы инерциального управления. Алгоритмические аспекты*, Киев, Украина: Наукова думка, 1991.
- [20] А. Панов, *Математические основы теории инерциальной ориентации*, Киев, Украина: Наукова думка, 1995.
- [21] В. Слюсарь, "Актуальные вопросы проектирования алгоритмов ориентации БИНС. Ч. 1. Амплитудное расширение области применения алгоритмов", *Гироскопия и навигация*, № 2, с. 61-75, 2006.
- [22] Н. Кробка, "Некоммутативные кинематические эффекты вращения твердого тела вокруг точки и их проявления в особенностях построения безплатформенных систем ориентации на лазерных и волоконно-оптических гироскопах", *Вестник Нижегородского университета им. Н. И. Лобачевского*, № 4(2), с.181-183, 2011.
- [23] Ю. Челноков, С. Переляев и Л. Челнокова, "Исследование алгоритмов определения инерциальной ориентации движущегося объекта", *Известия Саратовского университета. Сер. Математика. Механика. Информатика*, т. 16, вып. 1, с. 80-95, 2016.
- [24] М. Буденный, Я. Калиновский, А. Панов, А. Петренко, Т. Постникова, М. Синьков и Т. Синькова, "Об автоматизированном проектировании системы программно-аппаратных средств на базе гиперкомплексных чисел для задач ориентации твердого тела. Ч. 2.",

- Реєстрація, зберігання і обробка даних, Т. 4, № 4, с. 77-85, 2002.
- [25] L. Huang, J. Liu, Q. Zeng and Z. Xiong, "A new second-order strap down attitude algorithm", *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 8, pp. 3449-3462, 2013.
- [26] Qin Yong Yuan. Inertial Navigation, Beijing: China Science Publishing & Media Ltd, 2015, 368 p.
- [27] M. Wang, W. Wu, J. Wang, and X. Pan, "High-order attitude compensation in coning and rotation coexisting environment", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, pp. 1178-1190, 2015.
- [28] J. Wu, Z. Zhou, J. Chen, H. Fourati, and R. Li, "Fast Complementary Filter for Attitude Estimation Using Low-Cost MARG Sensors," *IEEE Sensors Journal*, vol. 16, pp. 6997-7007, 2016.
- [29] Z. Wu, Z. Sun, W. Zhang, and Q. Chen, "A Novel Approach for Attitude Estimation Based on MEMS Inertial Sensors Using Nonlinear Complementary Filters," *IEEE Sensors Journal*, vol. 16, pp. 3856-3864, 2016.
- [30] W. Yuanxin, "RodFilter: Attitude Reconstruction From Inertial Measurement by Functional Iteration", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, issue: 5, pp. 2131 – 2142, 2018.
- [31] Ю. Лазарев, П. Аксененко и С. Мураховский, "Исследование точности алгоритмов БИСО", *Вісник НТУУ "КПІ". Серія Приладобудування*, вип. 52(2), с. 10-15, 2016.
DOI: 10.20535/1970.52(2).2016.92739
- [32] Ю. Лазарев и П. Мироненко, "Сравнение точности некоторых алгоритмов бесплатформенных инерциальных систем ориентации", *Вісник НТУУ "КПІ". Серія Приладобудування*, вип. 53(1), с. 5-10, 2017.
DOI: 10.20535/1970.53(1).2017.106547
- [33] Ю. Лазарев и С. Мураховский, "Об одном алгоритме БИСО, минимизирующем дрейф погрешности численного интегрирования", *Вісник НТУУ "КПІ". Серія Приладобудування*, Вип. 50(2), с. 40-46, 2015.

УДК 629.051

¹Ю. Ф. Лазарев, ¹В. В. Аврутов, ¹П. С. Мироненко, ²С. О. Давиденко, ¹О. М. Сапегін¹Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна,²Державне підприємство «Конструкторське бюро «Південне» ім. М. К. Янгеля», Дніпро, Україна

ОДНОКРОКОВИЙ АЛГОРИТМ БИСО ПІДВИЩЕНОЇ ТОЧНОСТІ НА ОСНОВІ КВАТЕРНІОННОГО РІВНЯННЯ ОРІЄНТАЦІЇ І ВИМІРЮВАНЬ КУТОВОЇ ШВИДКОСТІ

Вступ. Обчислювальні алгоритми бесплатформних інерційних навігаційних систем (БІНС) можна розділити на навігаційні алгоритми, які перетворюють вихідні сигнали акселерометрів у шукані координати місця розташування об'єкта та алгоритми орієнтації, які перетворюють вихідні сигнали гіроскопів у кути орієнтації рухомого об'єкта [1]. При цьому для вирішення навігаційної задачі необхідно двічі інтегрувати прискорення, а для вирішення завдання орієнтації – інтегрувати диференціальні кінематичні рівняння орієнтації, що зв'язують виміряну кутову швидкість об'єкта з параметрами орієнтації.

У роботі йдеться про автономні методи позиціонування на основі інформації про кутову швидкість руху об'єкта без використання інформації про лінійне прискорення, тому далі розглядаються обчислювальні алгоритми орієнтації, що складають найважливішу частину БІНС – бесплатформову інерціальну систему орієнтації (БИСО).

У статті розглядаються похибки алгоритмів, побудованих на використанні кватерніонних рівнянь орієнтації, при чому у якості основної характеристики точності алгоритмів БИСО прийнято дрейфи цих похибок.

Основна частина. Дослідження алгоритмів проводиться шляхом моделювання роботи бортового обчислювача відповідно до розглянутих алгоритмів. Кінцевим результатом комп'ютерного моделювання є встановлення залежності дрейфів похибок чисельного інтегрування рівняння орієнтації від кроку опитування датчиків при різних значеннях частоти і амплітуди кутових коливань основи. Розглянуто чотири однокрокових алгоритми - реверсивний, на основі модифікованого методу Ейлера, побудований методом Пікара двома послідовними наближеннями і авторський на основі комбінування формул перших двох алгоритмів. Досліджено залежності дрейфів похибок від різниці фаз між коливаннями об'єкта навколо двох його ортогональних осей. Показано, що найбільші за величиною дрейфи в усіх випадках спостерігаються при конічному русі об'єкта. Проведено модельні дослідження залежностей амплітудних дрейфів від кроків опитування і частоти коливань у безрозмірній формі. Показано суттєве збільшення точності у нового алгоритму порівняно з усіма іншими розглянутими.

Висновки. Точність за дрейфами похибок запропонованого алгоритму в 2600 разів перевищує аналогічну точність використовуваного реверсивного алгоритму. Незначною видозміною формули однокрокового алгоритму вдалося підвищити точність на кілька порядків, практично не змінюючи обсяг обчислень. Отримані ре-

зультати дозволяють розширити область використання алгоритмів БІСО і прогнозувати їх точність при різних рухах основи.

Ключові слова: безплатформна інерціальна система орієнтації; однокроковий алгоритм; дрейф похибки.

¹⁾Ю. Ф. Лазарев, ¹⁾В. В. Аврутов, ¹⁾П. С. Мироненко, ²⁾С. А. Давиденко, ¹⁾А. Н. Сапегин

¹⁾Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Киев, Украина,

²⁾Государственное предприятие «Конструкторское бюро «Южное» им. М. К. Янгеля», Днепро, Украина

ОДНОШАГОВЫЙ АЛГОРИТМ БИСО ПОВЫШЕННОЙ ТОЧНОСТИ НА ОСНОВЕ КВАТЕРНИОННОГО УРАВНЕНИЯ ОРИЕНТАЦИИ И ИЗМЕРЕНИЙ УГЛОВОЙ СКОРОСТИ

Введение. Вычислительные алгоритмы безплатформенных инерциальных навигационных систем (БИНС) можно разделить на навигационные алгоритмы, которые преобразуют выходные сигналы акселерометров в искомые координаты местоположения объекта, и алгоритмы ориентации, которые преобразуют выходные сигналы гироскопов в углы ориентации подвижного объекта [1]. При этом для решения навигационной задачи необходимо дважды интегрировать ускорение, а для решения задач ориентации – интегрировать дифференциальные кинематические уравнения ориентации, связывающие измеренную угловую скорость объекта с параметрами ориентации.

В работе речь идёт об автономных методах определения углового положения объекта на основе информации об угловой скорости его движения, поэтому далее рассматриваются вычислительные алгоритмы ориентации, составляющие важнейшую часть БИНС – безплатформенную инерциальную систему ориентации (БИСО).

В статье рассматриваются погрешности алгоритмов, построенных на использовании кватернионных уравнений ориентации, причём в качестве основной характеристики точности алгоритмов БИСО приняты дрейфы этих погрешностей.

Основная часть. Исследование алгоритмов проводится путём моделирования работы бортового вычислителя в соответствии с рассматриваемыми алгоритмами. Конечным результатом компьютерного моделирования является установление зависимости дрейфов погрешностей численного интегрирования уравнения ориентации от шага опроса измерителей при различных значениях частоты и амплитуды угловых колебаний основания.

Рассмотрены четыре одношаговых алгоритма – реверсивный, на основе модифицированного метода Эйлера, построенный методом Пикара двумя последовательными приближениями и авторский на основе комбинирования формул первых двух алгоритмов.

Изучены зависимости дрейфов погрешностей от сдвигов фаз между колебаниями объекта вокруг двух его ортогональных осей. Показано, что наибольшие по величине дрейфы во всех случаях наблюдаются при коническом движении объекта.

Проведены модельные исследования зависимостей амплитудных дрейфов от шагов опроса и частоты колебаний в безразмерной форме. Показано существенное увеличение точности у нового алгоритма по сравнению со всеми другими рассмотренными.

Выводы. Точность по дрейфам предложенного алгоритма в 2600 раз превышает аналогичную точность используемого реверсивного алгоритма. Незначительным видоизменением формулы одношагового алгоритма удалось повысить точность на несколько порядков, практически не изменяя объем вычислений. Полученные результаты позволяют расширить область использования алгоритмов БИСО и прогнозировать их точность при различных движениях основания.

Ключевые слова: безплатформенная инерциальная система ориентации; одношаговый алгоритм; дрейф погрешности.

Надійшла до редакції
10 листопада 2020 року

Рецензовано
20 листопада 2020 року