Algorithm for Calculating the Orientation of a Satellite in Free Flight

Egor Moschyonok

Abstract. The paper considers the problem of software calculation of the orientation of a freely rotating rigid body. The paper compares algorithms based on numerical methods and exact solutions of Euler equations. Finding the period of angular velocity in the exact solution allows to decrease the orientation calculation time and reduce the accumulated errors compared to numerical algorithms.

Introduction

The relevance of accurate and fast algorithms for spacecraft orientation is constantly growing. Modern missions operate further and further from Earth, and communication delays require complete autonomy. At the same time, scientific tasks, such as astronomical observations or landings on small bodies, require extremely high pointing accuracy. The orientation problem is related to the problem of finding the angular velocity. Mathematically, the problem of finding the angular velocity is represented by the Euler equations. Free rotation of a body with projections p, q, r of the angular velocity $\boldsymbol{\omega}$ and principal moment of inertia A, B, C corresponding to principal axes i, j, k, rigidly connected to the body, is described by the Euler equations.

$$\left.\begin{aligned} \dot{p} &= -\frac{(C-B)}{A}\,qr, \\ \dot{q} &= -\frac{(A-C)}{B}\,rp, \\ \dot{r} &= -\frac{(B-A)}{C}\,pq. \end{aligned}\right\} \qquad \text{Euler equations} \qquad (1)$$

The solution of this system of differential nonlinear equations allows us to find the angular velocity at time t using the given initial data p_0, q_0, r_0.

There are two ways to calculate orientation. The first is by sensors. When the sensors measure the angular velocity with a certain time step, then these raw data obtained from the gyroscopes are filtered to smooth out noise. Then, based on this data, the orientation data is calculated. This method is subject to accumulating errors of the gyroscopes. The second method is based on a model consisting of a system of Euler equations and a kinematic equations. Most often, these equations are solved using numerical methods. In work [1], an exact solution to this system was found.

In the work, algorithms for calculating orientation and angular velocity using numerical methods and an exact solution were implemented in the Python 3 programming language.

To compare and test these algorithms, two data sets were used. The first is the experimental data obtained from [2], and the second is the telemetry satellite data obtained from [3].

A comparison of the execution speed of the numerical and exact solution algorithms on the selected computer is presented in the 1table.1. The explicit Runge-Kutta method of order 8 was chosen for the numerical algorithm. The time for calculating the angular velocity and orientation at the last point t_n of the time interval by the numerical method could not be determined for time intervals $t = [0, 6 \cdot 10^6]$ and longer ones. In these cases, the program did not complete its execution in 15 minutes. For all intervals except $t = [0, 6 \cdot 10^7]$, the angular velocity was calculated 4 times per second, for $t = [0, 6 \cdot 10^7]$ approximately once per second.

Conclusion

The exact algorithm can be used to predict the orientation of a body several hours in advance or more. The exact algorithm is faster and less susceptible to cumulative errors over large time intervals due to the calculations being performed over a period rather than over the entire time interval.

References

[1] S. Adlaj, *An exact calculation of orientation of a freely rotating rigid body*, available at `semjonadlaj.com/SP/ExactFreeRotation.pdf`.

[2] N. P. Kopytov and E. A. Mityushov, *Using wireless sensors to monitor the rotational motion of a "free-flight" rigid body.*, available at `www.mathnet.ru/php/seminars.phtml?presentid=43777`

[3] Ivanov, D.S. and Ovchinnikov, M.Yu. and Pantsyrny, O.A. and Selivanov, A.S. and Fedorov, I.O. and Khromov, O.E. and Yudanov, N.A, *Angular motion of the nanosatellite TNS-0 No. 2*, available at `library.keldysh.ru/preprint.asp?id=2017-126`.

TABLE 1. Execution time for different t

Time interval $t = [t_0, t_n]$	$t = [0, 60]$	$t = [0, 6 \cdot 10^5]$	$t = [0, 6 \cdot 10^6]$	$t = [0, 6 \cdot 10^7]$
Execution time of finding angular velocity using **numerical** algorithm	0.92s	43s	–	–
The execution time of finding the orientation at the last point t_n of the time interval **numerical** algorithm	1.7s	45s	–	–
Execution time of finding angular velocity using **exact** algorithm	1s	7.5s	32s	2m 11s
The execution time of finding the orientation at the last point t_n of the time interval **exact** algorithm	1.8s	9.5s	34.7s	2m 7s

Egor Moschyonok
National University of Science and Technology MISIS
Moscow, Russia
e-mail: moschyonokegor@gmail.com