Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Star Sensor Simulation with Application of the *k*-NN Method in Star Identification Problem

To cite this article: Danhe Chen *et al* 2022 *J. Phys.: Conf. Ser.* **2235** 012103

Journal of Physics: Conference Series **2235** (2022) 012103 doi:10.1088/1742-6596/2235/1/012103

Star Sensor Simulation with Application of the *k*-NN Method in Star Identification Problem

Danhe Chen, Vitaliya V. Kudlak, Andrey L. Maslennikov*

Nanjing University of Science and Technology, China
Bauman Moscow State Technical University (BMSTU), 2-nd Baumanskaya Street, 5, Moscow, Russian Federation, 105005

Corresponding author's e-mail: amas@bmstu.ru

Abstract. Simulation of the navigation systems is required to develop efficient, reliable, and accurate algorithms for orientation problems. In this paper the astronavigation system is considered, and the star sensor simulation based on algorithmic mathematical model is discussed. The algorithms include the starry sky image obtaining, data processing, and stars identification. Two modifications of the known stars identification algorithm are proposed and related to the use of *k*-NN method in identification and stars selection criteria for increasing computational efficiency. Simulation of the described approach for star sensor is performed in MathWorks MATLAB. The simulation results show adequate results.

1. Introduction

For proper spacecraft control its orientation must be known with high precision. For the CubeSat satellite solution to the orientation problem could be computed using astronavigation system or inertial navigation system or their combination [1]. Inertial navigation is not autonomous and may not be available directly after launch without knowing initial conditions of the spacecraft orientation [2]. Oppositely, the astronavigation system is autonomous and could run without precise knowledge of initial conditions [3].

The main device of the astronavigation system is a star sensor. It consists of photo-sensitive matrix and a microcontroller for data processing. When a star sensor shutter is open the light from celestial objects falls on the photo-sensitive matrix and transformed into digital code and become a digital image. Further, different types of filtering are applied to this image and then the orientation problem is solved by special algorithms. Solution to the orientation problem is obtained by comparing the image obtained from the photo-sensitive matrix and a star catalogue [4, 5]. When the image coincides with some fragment of the starry sky formed from a star catalogue then the direction of the star sensor sight axis is intuitively defined as well as the spacecraft orientation. The algorithms that are intent to determine such coincidence are stars identification algorithms.

To develop efficient and reliable stars identification algorithms the modeling approach should be used [6]. For this purpose, an algorithmic model of starry sky image obtaining is needed as well as imitation of optical defects [7, 8, 9]. There are exists some algorithms for stars identification. In this paper the algorithm presented in work [8] is considered and modified and its practical realization is presented and discussed.

2. Star sensor simulation algorithm

An image of the starry sky could be obtained in different ways. The first one by using virtual planetarium software, for instance Stellarium [10]. The second way is based on computation of the stars projections on the focus plane using initial data from a star catalog. It is important to note that the second approach is considered since it autonomous and could be adjusted to characteristics of a particular star sensor. The whole simulation algorithm consists of several stages: starry sky image creation, optical defects formation via digital filters, and stars identification. Let us describe this algorithm in more details.

2.1. Star sky image creation

To begin with the stars data from a star catalogue is loaded. Hipparcos star catalogue was used in our calculations [11]. At the first step, the projections for the stars from a star catalogue are computed on the virtual focus plane of a star sensor. At the second step, different filters are applied to the image to simulate different optical defects, for instance, spherical aberration, image noise, distortion, or motion aberration. The resulted image is further used in the stars identification algorithm, where its processing starts also with applying digital filters but to remove optical defects if it is possible.

2.2. Stars centers computation

After the image is filtered then the centers of the bright objects on it are computing. For this purpose, a matrix with brightness levels is calculated in a line-at-a-time manner. The boundaries of the bright objects are identified when the value in the matrix is differ from zero. The center of each bright object is then computed as follows

$$x = \frac{\max(n_x) - \min(n_x)}{2}, \quad y = \frac{\max(n_y) - \min(n_y)}{2} \tag{1}$$

where x and y are the corresponding coordinates (pixels) of the bright object center, n_x and n_y are the arrays of the bright object boundaries also stored as a pixels.

2.3. Stars identification

Stars identification is performed by comparison the disposition of the of the bright objects on the image with disposition of the stars from star catalogue. The comparison is performed using triangles that are formed by three stars or three bright objects. For this purpose, the projections on the virtual focus plane P of the stars from a star catalogue are computed. In this computation the coordinate transformation between spherical (positions of the stars from a star catalogue) and cartesian coordinates (positions of the bright objects on the image) is used. The transformation of the tangent projection establish relation between spherical (α, δ) and cartesian (ξ, η) coordinates that results in the following equations

$$\begin{aligned} \xi &= \frac{\mathrm{ctg}(\delta)\sin(\alpha - \alpha_0)}{\sin(\delta_0) + \mathrm{ctg}(\delta)\cos(\delta_0)\cos(\alpha - \alpha_0)}; \\ \eta &= \frac{\cos(\delta_0) - \mathrm{ctg}(\delta)\sin(\delta_0)\cos(\alpha - \alpha_0)}{\sin(\delta_0) + \mathrm{ctg}(\delta)\cos(\delta_0)\cos(\alpha - \alpha_0)}, \end{aligned} \tag{2}$$

where (α_0, δ_0) are the average spherical coordinates for the triad of stars. This idea is illustrated on Fig. 1, where S is a star and S^* is projection.

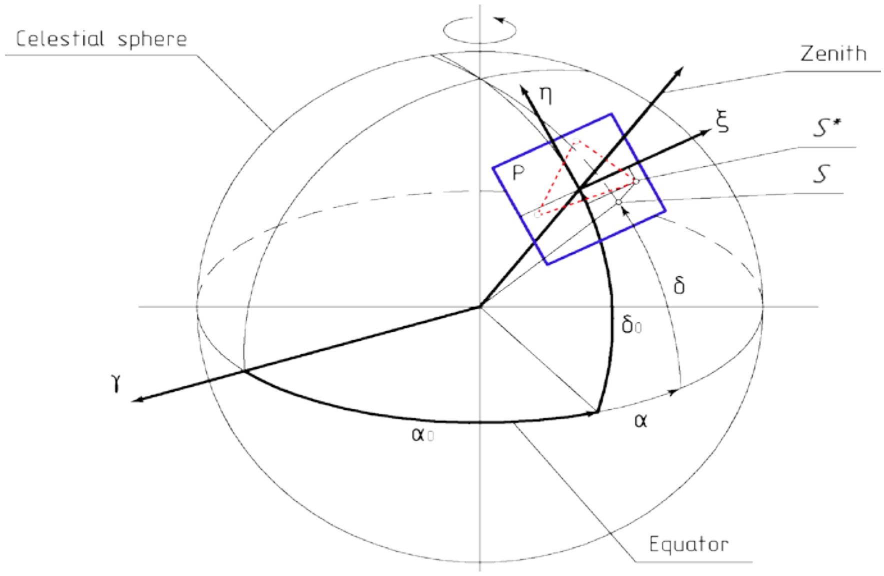

Figure 1. Illustration of the geometry of the stars projections on image plane

Further, all possible triangles are formed for all bright objects centers (x_i, y_i) and for stars projections (ξ_j, η_j). Then those triangles are compared (triangles based on the bright objects centers and triangles based on the stars projections) by side lengths. To determine the triangles likelihood two additional parameters are introduced

$$p = \frac{a}{c}, \quad q = \frac{b}{c}, \tag{3}$$

where a, b, and c are the side lengths of a triangle with a condition that $c > b > a$. The likelihood criterion is defined as follows

$$\begin{cases} |p_1 - p_2| < \varepsilon; \\ |q_1 - q_2| < \varepsilon, \end{cases} \tag{4}$$

where (p_1, q_1) are the corresponding parameters of triangles based on the bright objects centres, and (p_2, q_2) has the same meaning but computed for stars projections case, and ε is the likelihood tolerance.

In the described approach triangles based on the stars projections are computed for every three stars from a star catalogue. However, it is clearly ineffective since there a lot of stars that are located too far away from each other and cannot simultaneously show up on the image obtained by a star sensor. That implies that only computation of triangles that are fit in star sensor field of view make sense. To realize this a choice of the stars projections for which triangles are computed is made based on the angular distance between two stars projections

$$2\,\mathrm{arctg}\left(\frac{\|r_i - r_j\|}{2}\right) \le \gamma \quad i, j = \overline{1...n}, \quad j \ne i \tag{5}$$

where r_k is a radius vector for every star projection, and γ is a constant equals to the angle of a star sensor field of view.

The information about triangles likelihood is stored in a special $n_1 \times n_2$ matrix N , where n_1 is a number of the stars projections, and n_2 is a number of the bright objects on the image. When the likelihood condition (4) is satisfied then the corresponding element of the matrix N is increased by 1. After the matrix N is completed the maximal value in each column is determined as well as the corresponding row number. Eventually, computed pairs (columns and row numbers) of the bright objects and the stars projections are form the set of matching pairs. From this point the orientation of the star sensor sight axis could be easily calculated, for example, as a geometrical center of the stars projections group that match to the bright objects centers.

2.4. Verification of an identified star cluster

In practice, the configuration of the bright objects on the image might corresponds to more than one segments of the starry sky. In this case the described above approach has multiple solutions, say star clusters. That implies that the computed set of matched pairs is not enough to identify starry sky region wither the star sensor sight axis is oriented. To resolve this issue the distance between stars projections must be additionally taken into account.

The selection of the right star cluster could be done by utilizing idea of *k*-NN method. As a criterion for *k*-NN method the estimate of closeness Q_i between every star projection is used. The closeness Q_i is not the actual distance between two stars projections in a Euclidean sense. This is the number of the distances that are less than the star sensor field of view. This amount computed as a sum of satisfactions of the following condition

$$D_d \leq \gamma \tag{6}$$

where D_d is an element of the quadratic matrix D with the values of the angular distances between stars projections. Finally, with *k*-NN method the group (star cluster) of the stars projections is selected as identified stars.

3. Simulation results

The described algorithmic mathematical model for a star sensor simulation including algorithm for the stars identification was realized in MathWorks MATLAB. Fig. 2 illustrates the created starry sky image (on the left) and the one computed from a star catalogue (on the right).

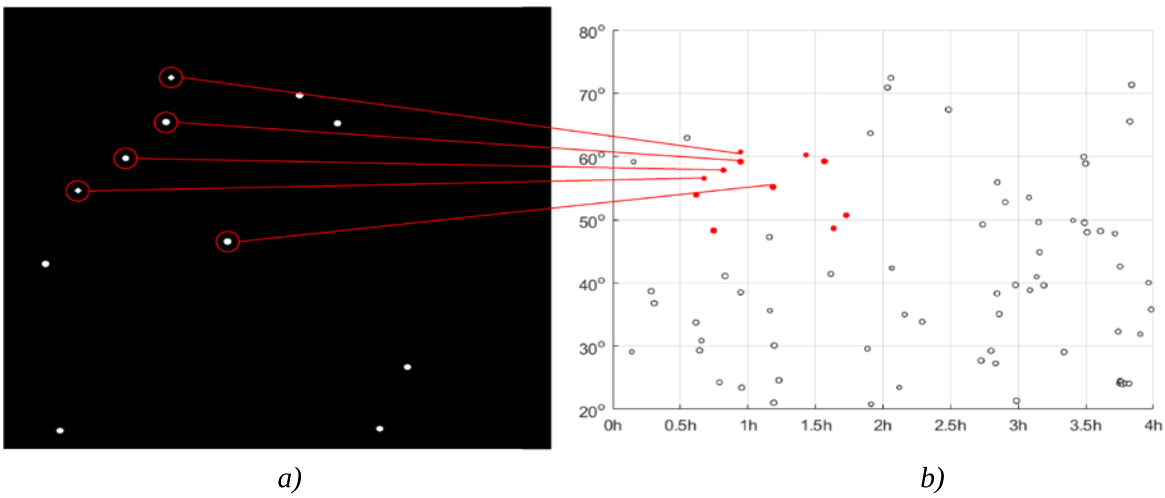

Figure 2. Simulation results: a) – created image of the starry sky, б) – starry sky image based on star catalogue and the corresponding identified stars

Red dots on the right represents the stars projections – candidates to be identified stars. However, the only few of them are forming group that satisfy criterion for the *k*-NN method. This group is selected as a set of stars corresponds to the bright objects on the image.

4. Conclusions and Discussions

In this paper the algorithmic model for simulation of a star sensor was described. To exclude some identification error, the *k*-NN method with the criterion related to the angular distance between stars projections was used. To increase the numerical efficiency of the algorithm stars projections are computed only for the combination of stars that are fit in the area related to the size of a star sensor field of view. The simulation was performed in MathWorks MATLAB and showed good identification results.

References

[1] Fokin, L.A. Multisensor Data Fusion in Adaptive Astro-Satellite-Inertial Navigation System. 2007 Siberian Conference on Control and Communications, 2007. DOI: 10.1109/SIBCON.2007.371294

[2] Salychev, O.S. Inertial systems in navigation and geophysics. Moscow, Bauman MSTU Press, 1998, 352 p.

[3] Ali, J. Astronavigation System as an Autonomous Enhancement Suite for a Strapdown Inertial Navigation System: An Evaluation. 2009, vol. 9, no. 2, pp. 42-48

[4] Gayvoronskiy, S.V., Rusin, E.V., Tsodokova, V.V. Stars identification at the astronomical coordinates determination by means of an automated zenith telescope // Scientific and technical journal of information technologies, mechanics, and optics. 2015, Vol. 15, no 1, pp.22–29.

[5] Ali, J., Changyun, Z., Jiancheng, F., An algorithm for astro-inertial navigation using CCD star sensors. Aerospace Science and Technology. 2006, vol. 10, pp.449-454

[6] Ryabogin N.V., Erokhin G.A., Pronina E.B., Koshelev A. Yu. Model-based systems engineering as a basis for development and creation of promising space systems and complexes // Rocket-Space Device Engineering and Information Systems. 2020, Vol. 7, no 3, pp. 51-60.

[7] Kudlak, V.V., Maslennikov, A.L. Starry sky image forming utilizing Gauss filter in modelling optical defects. Aviakosmicheskoe Proborostroenie, 2021, no. 12 (accepted for publications)

[8] Qiaoyun, F., Xiaojuan, L. and Guangjun, Z., 2012. Selection of star sensor lens aberration model. Infrared and Laser Engineering, 41(3), pp.665-670.

[9] Asadnezhad, M., Eslamimajd, A. and Hajghassem, H., 2018. Optical system design of star sensor and stray light analysis. Journal of the European Optical Society-Rapid Publications, 14(1), pp.1-11.

[10] Zotti, G. and Wolf, A., 2019. Stellarium 0.19. 0 User Guide. Technical report. Available online at github. com/Stellarium/stellarium/releases/download/v0. 19.0/stellarium user guide-0.19. 0-1. pdf. Accessed September 28, 2021.

[11] The Hipparcos and Tycho Catalogues, ESA Publications Division, c/o ESTEC, Noordwijk, The Netherlands, 1997, ISSN 0379–6566, ISBN 92–9092–399-7 (Volumes 1–17)