УДК 629.78
DOI: 10.17586/0021-3454-2025-68-1-89-97

# СРАВНЕНИЕ АЛГОРИТМОВ ОПРЕДЕЛЕНИЯ ОРИЕНТАЦИИ КОСМИЧЕСКОГО АППАРАТА НА ОСНОВЕ ОДНОМОМЕНТНЫХ ИЗМЕРЕНИЙ

**М. А. Пономарева*, А. В. Крамлих**

*Самарский национальный исследовательский университет имени академика С. П. Королева (Самарский университет), Самара, Россия*

**ponomarevamariaan@gmail.com*

**Аннотация.** Исследована задача определения ориентации космического аппарата по одномоментным измерениям. Представлен ряд алгоритмов определения ориентации космического аппарата по одномоментным измерениям различной физической природы: TRIAD, Optimized TRIAD, q-Method, QUEST, ESOQ, ESOQ2, SVD. Проанализированы следующие свойства алгоритмов, реализованных в математическом пакете MatLab: время работы, среднее значение и среднеквадратическое отклонение меры ошибки определения ориентации, количество операций с плавающей точкой.

***Ключевые слова:*** *космический аппарат, ориентация, алгоритм, одномоментные измерения, матрица поворота, кватернион, быстродействие, точность, вычисления с плавающей точкой*

**Благодарности:** исследование выполнено за счет гранта Российского научного фонда № 23-67-10007, https://rscf.ru/project/23-67-10007/.

**Ссылка для цитирования:** *Пономарева М. А., Крамлих А. В.* Сравнение алгоритмов определения ориентации космического аппарата на основе одномоментных измерений // Изв. вузов. Приборостроение. 2025. Т. 68, № 1. С. 89–97. DOI: 10.17586/0021-3454-2025-68-1-89-97.

## COMPARISON OF ALGORITHMS FOR DETERMINING A SPACE VEHICLE ORIENTATION BASED ON ONE-TIME MEASUREMENTS

**M. A. Ponomareva*, A. V. Kramlikh**

*Samara University, Samara, Russia*

** ponomarevamariaan@gmail.com*

**Abstract.** The problem of determining the orientation of a spacecraft using instantaneous measurements is investigated. A number of algorithms for determining the spacecraft orientation using one-time measurements of various physical nature are presented: TRIAD, Optimized TRIAD, q-Method, QUEST, ESOQ, ESOQ2, SVD. The following properties of the algorithms implemented in the MatLab mathematical package are analyzed: running time, average value and standard deviation of the orientation determination error measure, and the number of floating-point operations.

**Keywords:** spacecraft, orientation, algorithm, one-time measurements, rotation matrix, quaternion, performance, accuracy, floating point calculations

**Acknowledgments:** The study was supported by the Russian Science Foundation grant No. 23-67-10007, https://rscf.ru/project/23-67-10007/.

**For citation:** Ponomareva M. A., Kramlikh A. V. Comparison of algorithms for determining a space vehicle orientation based on one-time measurements. *Journal of Instrument Engineering*. 2025. Vol. 68, N 1. P. 89–97 (in Russian). DOI: 10.17586/0021-3454-2025-68-1-89-97.

**Введение.** Текущее положение космического аппарата (КА) вычисляется системой определения ориентации, включающей набор датчиков и комплекс алгоритмов. Информация об угловом положении КА используется для привязки измерений, выполняемых целевой аппаратурой, и для решения задачи управления угловым движением КА.

В настоящей статье рассматривается ряд основанных на работе с одномоментными измерениями алгоритмов определения ориентации. Одномоментные измерения векторов различной физической природы выполняются одновременно или в достаточно близкие моменты времени, чтобы движением центра масс и угловым движением КА можно было пренебречь. В таком качестве могут выступать [1] измерения от звездного датчика, магнитометра, датчика Солнца, датчика горизонта и т.п., а в качестве моделей измерений используются соответственно модель звездного неба, магнитного поля Земли, направления на Солнце, надир и др. (наиболее точным считается звездный датчик, наименее — датчик горизонта). На малоразмерных КА преимущественно используются магнитометры и датчики направления на Солнце, поскольку они достаточно точные, надежные и недорогие.

Для определения ориентации КА по одномоментным измерениям используются два и более векторов и моделей этих измерений различной физической природы. В алгоритмах TRIAD и Optimized TRIAD используются только два вектора, в q-Method, QUEST, ESOQ, ESOQ2 и SVD можно использовать два вектора и более. Поскольку угол между двумя различными векторами может быть близок к нулю или к 180°, то векторы вырождаются друг в друга и ориентация КА в этом случае определяется с очень большой ошибкой. Во избежание этого используют несколько векторов, чтобы из них выбрать те, угол между которыми не будет равен нулю или 180°.

Целью настоящей работы является сравнение наиболее популярных алгоритмов определения ориентации КА по следующим критериям: скорость работы, точность, тип выходной информации и сложность реализации.

**Математическая постановка задачи.** В настоящей работе используются:

— абсолютная геоцентрическая система координат $ox_ay_az_a$ (АСК): начало $O$ в центре Земли, ось $ox_a$ направлена в точку весеннего равноденствия, ось $oz_a$ направлена по оси суточного вращения Земли (перпендикулярно плоскости экватора), ось $oy_a$ лежит в плоскости экватора Земли и дополняет систему до правой (рис. 1; $\Omega$ — долгота восходящего узла, $i$ — наклонение орбиты, $u$ — аргумент широты). Изначально модели измерений задаются в АСК с помощью математических моделей магнитного поля Земли, движения Солнца, звезд и т. д., а затем переводятся в орбитальную систему координат;

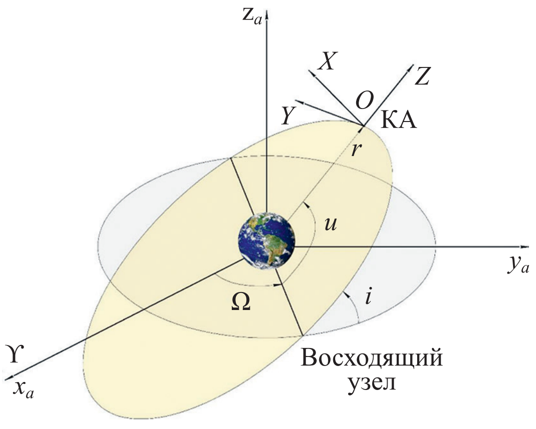


*Рис. 1*

— орбитальная система координат $OXYZ$ (ОСК): начало $O$ в центре масс КА, ось $OZ$ направлена по радиусу-вектору КА, ось $OY$ направлена перпендикулярно плоскости орбиты (по вектору кинетического момента орбитального движения), ось $OX$ направлена по вектору скорости КА (рис. 1). ОСК является опорной, т. е. относительно нее определяется угловое положение КА;

— связанная система координат $Oxyz$ (ССК): начало в центре масс КА, оси сонаправлены с главными центральными осями инерции КА. В ССК векторы определяются с помощью датчиков, находящихся на борту КА.

Задача определения ориентации КА заключается в нахождении углового положения связанной с КА системы координат относительно некоторой опорной системы координат, например орбитальной, по имеющимся измерениям в ССК и моделям этих измерений в ОСК. На рис. 2 проиллюстрирован переход от ОСК к ССК ($\psi$ — угол прецессии, $a_n$ — пространственный угол атаки, $\varphi$ — угол собственного вращения).

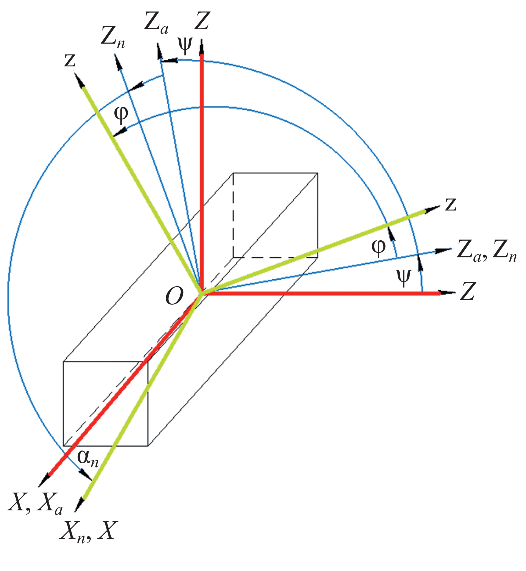


*Рис. 2*

Переход от ОСК к ССК осуществляется тремя последовательными поворотами вокруг осей $x - y - x$ на углы ψ, $\alpha_n$, φ соответственно. Эти углы называются углами Эйлера [2] и задаются следующим образом: угол прецессии $\psi \in [-\pi, \pi]$; пространственный угол атаки $\alpha_n \in [0, \pi]$ и угол собственного вращения $\varphi \in [-\pi, \pi]$.

Угловое положение ССК относительно ОСК описывается матрицей поворота **M**, параметризованной через углы Эйлера [2]:

$$\mathbf{M}(\alpha_n, \varphi, \psi) = \begin{bmatrix} c\alpha_n & s\alpha_n s\psi & -s\alpha_n c\psi \\ s\varphi_n s\alpha_n & c\varphi c\psi - c\varphi s\alpha_n s\psi & c\varphi s\psi + s\varphi c\alpha_n c\psi \\ c\varphi_n s\alpha_n & -s\varphi c\psi - c\varphi s\alpha_n s\psi & -s\varphi s\psi + c\varphi c\alpha_n c\psi \end{bmatrix} \quad (1)$$

или кватернион [3]

$$\mathbf{M}(\mathbf{Q}) = \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_4 q_3) & 2(q_1 q_3 - q_4 q_2) \\ 2(q_1 q_2 - q_4 q_3) & q_4^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 + q_4 q_1) \\ 2(q_1 q_3 + q_4 q_2) & 2(q_2 q_3 - q_4 q_1) & q_4^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}, \quad (2)$$

где $c = \cos$, $s = \sin$; $\mathbf{Q} = (q_1, q_2, q_3, q_4)$ — кватернион; $q_4$ — компоненты векторной части кватерниона, $q_1, q_2, q_3$ — скалярная часть кватерниона.

Связь измерений и моделей этих измерений описывается формулой:

$$\mathbf{b}_i = \mathbf{M}\mathbf{r}_i,$$

где $\mathbf{b}_i$ — вектор измерений ($i = \overline{1, N}$, $N$ — число единичных векторов измерений в ССК); $\mathbf{r}_i$ — вектор моделей измерений в ОСК; определитель **M** должен быть равен единице — условие ортогональности матрицы.

Решение задачи определения ориентации КА сводится к нахождению матрицы поворота **M**.

*TRIAD* — первый алгоритм, созданный для определения матрицы поворота [4–6]. Этот алгоритм работает только с двумя измеренными векторами: 1 — получен с помощью датчиков освещенности, 2 — вектор магнитной индукции; $\mathbf{r}_1$, $\mathbf{r}_2$ — проекции векторов в ССК, $\mathbf{b}_1$, $\mathbf{b}_2$ — проекции векторов в ОСК, которые удовлетворяют выражению

$$\mathbf{b}_1 = \mathbf{M}\mathbf{r}_1, \ \mathbf{b}_2 = \mathbf{M}\mathbf{r}_2.$$

В алгоритме учитывается только направление вектора, а не его длина, поэтому все векторы нормируются.

TRIAD основан на следующей идее: создается правая тройка векторов $\{\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3\}$ в ОСК и правая тройка векторов $\{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3\}$ в ССК с помощью полученных ранее векторов по следующим формулам:

$$\mathbf{R}_1 = \frac{\mathbf{r}_1}{|\mathbf{r}_1|}, \ \mathbf{R}_2 = \frac{\mathbf{R}_1 \mathbf{r}_2}{|\mathbf{R}_1 \mathbf{r}_2|}, \ \mathbf{R}_3 = \mathbf{R}_1 \mathbf{R}_2, \quad (3)$$

$$\mathbf{R}_1 = \frac{\mathbf{b}_1}{|\mathbf{b}_1|}, \ \mathbf{R}_2 = \frac{\mathbf{W}_1 \mathbf{b}_2}{|\mathbf{W}_1 \mathbf{b}_2|}, \ \mathbf{R}_3 = \mathbf{R}_1 \mathbf{R}_2. \quad (4)$$

Тогда матрицу поворота можно найти по формуле

$$\mathbf{M} = \mathbf{R}_1 \mathbf{W}_1^{\mathrm{T}} + \mathbf{R}_2 \mathbf{W}_2^{\mathrm{T}} + \mathbf{R}_3 \mathbf{W}_3^{\mathrm{T}}. \quad (5)$$

В алгоритме TRIAD вычисления опираются на один из векторов (вектор 1 используется чаще), для повышения точности алгоритма необходимо использовать наиболее точно найденный вектор. При разработке КА решается, какой из используемых датчиков считать более точным, в зависимости от принципа их работы (например, магнитометры и датчики Солнца бывают различной сложности и точности) и полноты математических моделей.

*Optimized TRIAD* [7], в нем учитываются не только ошибки определения вектора 2, но и ошибки определения вектора 1.

Для вычисления матрицы поворота по Optimized TRIAD необходимо найти матрицу $\mathbf{M}_1$ согласно формуле (5), считая вектор 1 более точным; затем — по (5) найти матрицу $\mathbf{M}_1$, считая более точным вектор 2. Также нужно получить значения стандартных отклонений: $\sigma_1$ — среднеквадратическое отклонение (СКО) измерений вектора 1, а $\sigma_2$ — СКО вектора 2. Тогда по формуле несмещенной оценки с минимальной дисперсией можно найти [7]:

$$\hat{\mathbf{M}} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\mathbf{M}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\mathbf{M}_2. \tag{6}$$

Для выполнения условия ортогональности матрицы $\mathbf{M}$ необходимо преобразовать выражение (6) следующим образом:

$$\mathbf{M} = 0{,}5\left[\hat{\mathbf{M}} + \left(\hat{\mathbf{M}}^{-1}\right)^{\mathrm{T}}\right].$$

*Задача Вахбы.* Поиск матрицы поворота между двумя системами координат может сводиться к так называемой задаче Вахбы [8]: найти ортогональную матрицу $\mathbf{M}$, доставляющую минимум функции потерь, представляющую собой разность между измеренными векторами и их моделями

$$L(\mathbf{M}) = \sum_{i=1}^{N} a_i \|\mathbf{b}_i - \mathbf{M}\mathbf{r}_i\|^2 \to \min,$$

где $\{\mathbf{b}_i\}$ — совокупность единичных векторов измерений в ССК, $\{\mathbf{r}_i\}$ — совокупность единичных векторов моделей измерений в ОСК, $a_i$ — весовой коэффициент соответствующих измерений.

Функцию потерь можно свести к выражению [8]:

$$L_1 = \mathrm{tr}(\mathbf{M}\mathbf{B}^{\mathrm{T}}) \to \max, \tag{7}$$

где $\mathbf{B} = \sum_{i=1}^{N} a_i \mathbf{b}_i \mathbf{r}_i^{\mathrm{T}}$, tr() — след матрицы.

В дальнейшем вместо функции потерь $L$ будем работать с функцией $L_1$.

Рассматриваемые далее алгоритмы предназначены для решения задачи Вахбы.

*q-Method* — один из первых, созданных для решения задачи Вахбы [6, 9–11]: вводится матрица поворота, параметризованная через кватернион

$$\mathbf{M} = (q_4^2 - \mathbf{q}^{\mathrm{T}}\mathbf{q})\mathbf{I} + 2\mathbf{q}\mathbf{q}^{\mathrm{T}} - 2q_4[\mathbf{q}\times], \tag{8}$$

где $\mathbf{Q} = [\mathbf{q} \;\; q_4]^{\mathrm{T}}$ — кватернион, $[\mathbf{q}\times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$, $\mathbf{I}$ — единичная матрица.

Подставив (8) в уравнение (7) и преобразовав его, получим

$$L_1 = \mathrm{tr}(\mathbf{M}\mathbf{B}^{\mathrm{T}}) = \mathbf{Q}^{\mathrm{T}}\mathbf{K}\mathbf{Q} \to \max, \tag{9}$$

где $\mathbf{K} = \begin{bmatrix} \mathbf{S} - \mathbf{I}\mathrm{tr}(\mathbf{B}) & \mathbf{z} \\ \mathbf{z}^{\mathrm{T}} - \mathbf{B}_{13} & \mathrm{tr}(\mathbf{B}) \end{bmatrix}$, $\mathbf{S} = \mathbf{B} + \mathbf{B}^{\mathrm{T}}$, $\mathbf{z} = \begin{bmatrix} \mathbf{B}_{23} - \mathbf{B}_{32} \\ \mathbf{B}_{31} - \mathbf{B}_{13} \\ \mathbf{B}_{12} - \mathbf{B}_{21} \end{bmatrix}$.

Рассмотрев правую часть уравнения (9), можно заметить, что матрица  постоянна. Тогда для того, чтобы значение функции (9) было максимальным, необходимо, чтобы значение $\mathbf{Q}$ было максимальным. Можно также доказать [10], что **Q** является собственным вектором матрицы **K**. Таким образом, задача сводится к нахождению максимального значения вектора **Q**, который также называют оптимальным кватернионом, или к решению следующего уравнения:

$$\mathbf{K}\mathbf{Q}_{\mathrm{opt}} = \lambda_{\max}\mathbf{Q}_{\mathrm{opt}}, \tag{10}$$

где $\lambda_{\max}$ — максимальное собственное число матрицы **K**.

q-Method не предлагает конкретного способа отыскания максимального собственного вектора матрицы **K**, являющегося оптимальным кватернионом $\mathbf{Q}_{\mathrm{opt}}$. Алгоритмы QUEST [5, 6, 10–12], ESOQ [6, 10, 13] и ESOQ2 [6, 10, 14] предлагают различные способы нахождения этого собственного вектора. Рассмотрим эти алгоритмы подробнее.

*QUEST*. Уравнение (10) можно переписать как

$$(\lambda_{\max}\mathbf{I} - \mathbf{K})\mathbf{Q}_{\mathrm{opt}} = 0, \tag{11}$$

$\lambda_{\max}$ в случае использования двух векторов измерений можно найти по формуле

$$\lambda_{\max} = \sqrt{a_1^2 + 2a_1a_2c + a_2^2},$$

$$c = (\mathbf{b}_1\mathbf{b}_2)(\mathbf{r}_1\mathbf{r}_2) + \|\mathbf{b}_1\mathbf{b}_2\|\,\|\mathbf{r}_1\mathbf{r}_2\|,$$

где $\|\ \|$ — вторая норма.

Тогда из уравнения (10) можно найти оптимальный кватернион по формуле

$$\mathbf{Q}_{\mathrm{opt}} = \frac{1}{\sqrt{\gamma^2 + |\mathbf{x}|^2}}\begin{bmatrix}\mathbf{x}\\ \gamma\end{bmatrix},$$

где $\mathbf{x} = (\alpha\mathbf{I} + \beta\mathbf{S} + \mathbf{S}^2)\mathbf{z}$, $\gamma = \alpha(\lambda_{\max} + \mathrm{tr}(\mathbf{B})) - \det(\mathbf{S}^2)$, det() — определитель матрицы, $\alpha = \lambda_{\max}^2 - (\mathrm{tr}(\mathbf{B}))^2 + \mathrm{tr}(\mathrm{adj}(\mathbf{S}))$, adj() — присоединенная матрица, $\beta = \lambda_{\max} - \mathrm{tr}(\mathbf{B})$.

В алгоритме ESOQ вводится вспомогательная матрица

$$\mathbf{H} = \lambda_{\max}\mathbf{I} - \mathbf{K}.$$

Необходимо найти присоединенную матрицу adj(**H**) и $k$ — номер столбца с максимальным диагональным элементом матрицы adj(**H**).

Далее находятся ненормированные элементы оптимального кватерниона

$$q_i = (-1)^{k+1}\det(\mathbf{H}_{ki}),$$

где $i = 1, \ldots, 4$, а $\mathbf{H}_{ki}$ получена из матрицы **H** путем вычеркивания $k$-й строчки и $i$-го столбца.

Оптимальный кватернион нормируется по формуле

$$\mathbf{Q}_{\mathrm{opt}} = \frac{1}{\sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}}\begin{bmatrix}q_1\\ q_2\\ q_3\\ q_4\end{bmatrix}.$$

*ESOQ2*. В уравнение (10) подставляется кватернион вида $\mathbf{Q} = \begin{bmatrix}\mathbf{e}\sin\left(\dfrac{\theta}{2}\right)\\ \cos\left(\dfrac{\theta}{2}\right)\end{bmatrix}$, где **e** — ось поворота; $\theta$ — угол поворота.

Затем с помощью математических преобразований можно получить

$$\mathbf{P} = (\lambda_{\max} - \mathrm{tr}(\mathbf{B}))((\lambda_{\max} + \mathrm{tr}(\mathbf{B}))\mathbf{I} - \mathbf{S}) - \mathbf{z}\mathbf{z}^{\mathrm{T}} = [\mathbf{p}_1 \vdots \mathbf{p}_2 \vdots \mathbf{p}_3],$$

далее

$$\mathbf{P}_1 = [\mathbf{p}_2\mathbf{p}_3 \vdots \mathbf{p}_3\mathbf{p}_1 \vdots \mathbf{p}_1\mathbf{p}_2]$$

и вектор $\mathbf{y}$ — столбец $\mathbf{P}_1$ с максимальной второй нормой.

Оптимальный кватернион находится по формуле

$$\mathbf{Q}_{\mathrm{opt}} = \frac{1}{\sqrt{|(\lambda_{\max} - \mathrm{tr}(\mathbf{B}))\mathbf{y}|^2 + |\mathbf{z}\mathbf{y}|^2}} \begin{bmatrix} (\lambda_{\max} + \mathrm{tr}(\mathbf{B}))\mathbf{y} \\ \mathbf{z}\mathbf{y} \end{bmatrix}.$$

*SVD*. Также задачу Вахбы можно решить с помощью сингулярного разложения, используя алгоритм SVD [6, 10, 11, 15].

Матрица $\mathbf{B} = \sum_{i=1}^{N} a_i \mathbf{b}_i \mathbf{r}_i^{\mathrm{T}}$ имеет сингулярное разложение

$$\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}} = \mathbf{U} \begin{bmatrix} \Sigma_{11} & 0 & 0 \\ 0 & \Sigma_{22} & 0 \\ 0 & 0 & \Sigma_{33} \end{bmatrix} \mathbf{V}^{\mathrm{T}},$$

где $\mathbf{U}$ и $\mathbf{V}$ — ортогональные матрицы, состоящие из левых и правых сингулярных векторов соответственно, а собственные сингулярные числа $\Sigma$ подчиняются неравенствам $\Sigma_{11} \ge \Sigma_{22} \ge \Sigma_{33} \ge 0$. Матрица $\mathbf{U}$ состоит из собственных векторов матрицы $\mathbf{B}\mathbf{B}^{\mathrm{T}}$, матрица $\mathbf{V}$ состоит из собственных векторов матрицы $\mathbf{B}^{\mathrm{T}}\mathbf{B}$.

Матрица поворота находится по формуле

$$\mathbf{M} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U})\det(\mathbf{V}) \end{bmatrix} \mathbf{V}^{\mathrm{T}}.$$

**Моделирование.** Для сравнения алгоритмов было проведено математическое моделирование в MatLab при следующих условиях. Углы ориентации выбирались из диапазона $\alpha_n \in [0;\ 180°]$, $\varphi \in [-180°;\ 180°]$, $\psi \in [-180°;\ 180°]$ и вычислялась истинная матрица поворота $\mathbf{M}$, которую можно найти по формуле (1).

Во всех рассмотренных алгоритмах определения ориентации КА используются нормированные векторы. Моделирование векторов $\mathbf{r}_1\mathbf{r}_2$ проводилось следующим образом: компоненты $r_{x_i}, r_{y_i}, r_{z_i}$ ($i = 1, 2$) векторов $\mathbf{r}_i = (r_{x_i}, r_{y_i}, r_{z_i})^{\mathrm{T}}$ генерировались по равновероятному закону в диапазоне $[-1000;\ 1000]$.

Векторы $\mathbf{b}_1$, $\mathbf{b}_2$ (измерения в ССК) получены по формуле

$$\mathbf{b}_i = \mathbf{M}\mathbf{r}_i + \mathbf{w},$$

где $\mathbf{w} = (w_x, w_y, w_z)^{\mathrm{T}}$, $w_x, w_y, w_z \sim N(0, \sigma^2)$, $\sigma = 0{,}01|\mathbf{r}|$ или $\sigma = 0{,}1|\mathbf{r}|$, $i = 1, 2$ (т. е. величина СКО принималась равной 1 или 10 % модуля вектора).

Мера ошибки определения ориентации КА находится по формуле

$$F = \arccos\left(\frac{1}{2}(\mathrm{tr}(\mathbf{B} - 1))\right), \tag{12}$$

где $\tilde{\mathbf{B}} = \mathbf{M}\hat{\mathbf{M}}^{\mathrm{T}}$, $\mathbf{M}$ — истинная матрица, $\hat{\mathbf{M}}$ — найденная матрица.

Для каждого алгоритма подсчитаны: мера ошибки определения ориентации, скорость работы для выборки размером 10 000, при этом число операций с плавающей точкой определялось для одной реализации алгоритма.

Результаты исследования приведены в табл. 1, 2 и на рис. 3.

В табл. 1 представлены результаты вычисления времени работы и числа операций с плавающей точкой. Самыми быстрыми алгоритмами являются TRIAD и SVD, самые медленные — ESOQ и QUEST. Алгоритм с наименьшим числом операций с плавающей точкой — TRIAD, с наибольшим числом — QUEST.

***Таблица 1***

| Алгоритм | Время работы*, с | Число операций с плавающей точкой** | Тип выходной информации |
|---|---|---|---|
| TRIAD | 0,04 | 11 | Матрица поворота |
| Optimized TRIAD | 0,14 | 42 | Матрица поворота |
| q-method | 0,20 | 59 | Кватернион |
| QUEST | 0,29 | 227 | Кватернион |
| ESOQ | 0,54 | 125 | Кватернион |
| ESOQ2 | 0,14 | 79 | Кватернион |
| SVD | 0,13 | 53 | Матрица поворота |

* Указано время работы алгоритмов при обработке выборки размером 10 000 реализаций.
** Для вычислений операций с плавающей точкой использовалась функция, реализованная в MatLab [16].

В табл. 2 представлены результаты вычисления среднего значения и СКО для меры ошибки определения ориентации КА (см. формулу (12)). Можно увидеть, что наименее точным является TRIAD, остальные алгоритмы работают примерно с одинаковой точностью. Также можно увидеть, что при увеличении ошибки измерения вектора в десять раз погрешность определения ориентации тоже увеличивается примерно в десять раз.

***Таблица 2***

| Алгоритм | Среднее значение ошибки, …° | | СКО, …° | |
|---|---|---|---|---|
| | $\sigma = 0{,}1\lvert\mathbf{r}\rvert$ | $\sigma = 0{,}01\lvert\mathbf{r}\rvert$ | $\sigma = 0{,}1\lvert\mathbf{r}\rvert$ | $\sigma = 0{,}01\lvert\mathbf{r}\rvert$ |
| TRIAD | 8,58 | 0,86 | 5,78 | 0,57 |
| Optimized TRIAD | 7,84 | 0,79 | 5,35 | 0,53 |
| q-method | 7,84 | 0,79 | 5,35 | 0,53 |
| QUEST | 7,84 | 0,79 | 5,35 | 0,53 |
| ESOQ | 7,84 | 0,79 | 5,35 | 0,53 |
| ESOQ2 | 7,84 | 0,79 | 5,35 | 0,53 |
| SVD | 7,84 | 0,79 | 5,35 | 0,53 |

На рис. 3 представлены результаты моделирования алгоритмов при ошибке в 1 % от модуля вектора ($\sigma = 0{,}01\lvert\mathbf{r}\rvert$). Можно увидеть, что алгоритм TRIAD — наименее точный, а графики остальных алгоритмов сливаются. На границах графика можно увидеть значительное увеличение погрешности при приближении к особым точкам. Это происходит, потому что при углах между векторами $\delta = 0$ или 180° векторы вырождаются друг в друга.

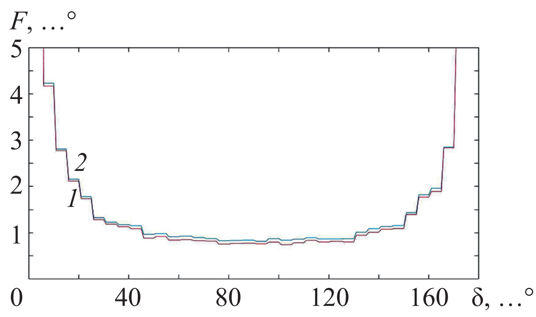


*Рис. 3*

**Заключение.** Моделирование показало, что наименее точным алгоритмом является TRIAD, остальные работают примерно с одинаковой точностью. Самыми быстрыми являются TRIAD и SVD, самые медленные — ESOQ и QUEST. Алгоритм с наименьшим числом операций с плавающей точкой — TRIAD, с наибольшим — QUEST.

## СПИСОК ЛИТЕРАТУРЫ


1. *Mohd Zamri H., Amran A., Abu Hassan A.* et al. Review on attitude estimation algorithm of attitude determination system // ARPN Journal of Engineering and Applied Sciences. 2016. Vol. 11, N 7. P.4455–4460.
2. *Белоконов И. В., Тимбай И. А*. Движение наноспутника относительно центра масс на околоземных орбитах: учеб. пос. Самара: Изд-во Самарского университета, 2020. 128 с.
3. *Бранец В. Н., Шмыглевский И. П*. Применение кватернионов в задачах ориентации твердого тела. М.: Наука, 1973. 320 с.
4. *Black H. D*. A Passive System for Determining the Attitude of a Satellite // AIAA Journal. 1964. Vol. 2, N 7. P. 1350–1351.
5. *Shuster M. D., Oh S. D*. Three-Axis Attitude Determination from Vector Observations // Journal of Guidance and Control. 1981. Vol. 4, N 1. P. 70–77.
6. *Markley F. L., Crassidis J. L.* Fundamentals of Spacecraft Attitude Determination and Control. Microcosm Press and Springer, 2014.
7. *Bar-Itzhack Y. I., Harman R. R*. Optimized TRIAD algorithm for attitude determination // Collection of Technical papers (A96-34712 09-12). 1996. P. 422–427.
8. *Wahba G. A*. Least-squares estimate of satellite attitude // SIAM Review. 1965. Vol. 8, N 3. P. 384–386.
9. *Davenport P.* A vector approach to the algebra of rotations with applications // NASA TN D-4696. 1968.
10. *Markley F. L., Mortari D.* How to estimate attitude from vector observations // AAS 99-427, AAS/AIAA. Astrodynamics Specialist Conference. Girdwood, Alaska, 1999.
11. *Hajiyev C., Soken H. E*. Fault Tolerant Attitude Estimation for Small Satellites. CRC Press, 2021.
12. *Shuster M. D*. Approximate Algorithms for Fast Optimal Attitude Computation // AIAA Paper 78-1249, AlAA. Guidance and Control Conference. Palo Alto, 7–9 August 1978.
13. *Mortari D*. ESOQ: A closed-form solution of the Wahba problem // Journal of Astronautical Sciences. 1997. Vol. 45, N 2. P. 195–205.
14. *Mortari D*. ESOQ2 Single-Point Algorithm for Fast Optimal Attitude Determination // Paper AAS 97–167, AAS/AIAA. Space Flight Mechanics Meeting. Huntsville, 10–12 February 1997.
15. *Markley F. L*. Attitude Determination Using Vector Observations and the Singular Value Decomposition // Journal of the Astronautical Sciences. 1988. Vol. 36, N 3. P. 245–258.
16. Counting the Floating Point Operations (FLOPS) [Электронный ресурс]: <https://www.mathworks.com/matlabcentral/fileexchange/50608-counting-the-floating-point-operations-flops>. (дата обращения 22.05.2024).


## СВЕДЕНИЯ ОБ АВТОРАХ


***Мария Андреевна Пономарева*** — студентка; Самарский национальный исследовательский университет имени академика С.П. Королева, межвузовская кафедра космических исследований; E-mail: ponomarevamariaan@gmail.com

***Крамлих Андрей Васильевич*** — канд. техн. наук, доцент; Самарский национальный исследовательский университет имени академика С.П. Королева, межвузовская кафедра космических исследований; E-mail: kramlikh@mail.ru


Поступила в редакцию 10.07.24; одобрена после рецензирования 09.09.24; принята к публикации 21.11.24.

**REFERENCES**


1. Mohd Zamri H., Amran A., Abu Hassan A. et al. *ARPN Journal of Engineering and Applied Sciences*, 2016, no. 7(11), pp. 4455–4460.
2. Belokonov I.V., Timbay I.A. *Dvizheniye nanosputnika otnositel'no tsentra mass na okolozemnykh orbitakh* (Motion of a Nanosatellite Relative to the Center of Mass in Near-earth Orbits), Samara, 2020, 128 p. (in Russ.)
3. Branets V.N., Shmyglevsky I.P. *Primeneniye kvaternionov v zadachakh oriyentatsii tverdogo tela* (Application of Quaternions in Problems of Rigid Body Orientation), Moscow, 1973, 320 p. (in Russ.)
4. Black H.D. *AIAA Journal*, 1964, no. 7(2), pp. 1350–1351.
5. Shuster M.D., Oh S.D. *Journal of Guidance and Control*, 1981, no. 1(4), pp. 70–77.
6. Markley F.L., Crassidis J.L. *Fundamentals of Spacecraft Attitude Determination and Control*, Microcosm Press and Springer, 2014.
7. Bar-Itzhack Y.I., Harman R.R. *Collection of Technical papers* (A96-34712 09-12), 1996, pp. 422–427.
8. Wahba G.A. *SIAM Review*, 1965, no. 3(8), pp. 384–386.
9. Davenport P. *NASA TN D-4696*, 1968.

10. Markley F.L., Mortari D. *AAS 99-427, AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Alaska, 1999.
11. Hajiyev C., Soken H.E. *Fault Tolerant Attitude Estimation for Small Satellites*, CRC Press, 2021.
12. Shuster M.D. AIAA Paper 78-1249, *AIAA Guidance and Control Conference*, Palo Alto, CA, August 7–9, 1978.
13. Mortari D. *Journal of Astronautical Sciences*, 1997, no. 2(45), pp. 195–205.
14. Mortari D. *Paper AAS 97–167, AAS/AIAA Space Flight Mechanics Meeting*, Huntsville, AL, February 10–12, 1997.
15. Markley F.L. *Journal of the Astronautical Sciences*, 1988, no. 3(36), pp. 245–258.
16. *Counting the Floating Point Operations (FLOPS)*, https://www.mathworks.com/matlabcentral/fileexchange/50608-counting-the-floating-point-operations-flops.

**DATA ON AUTHORS**

***Maria A. Ponomareva*** — Student, Samara University, Inter-University Department of Space Research; E-mail: ponomarevamariaan@gmail.com

***Andrey V. Kramlikh*** — PhD, Associate Professor; Student, Samara University, Inter-University Department of Space Research; E-mail: kramlikh@mail.ru

Received 10.07.24; approved after reviewing 09.09.24; accepted for publication 21.11.24.