Fast Linear Quaternion Attitude Estimator Using Vector Observations

Jin Wu, *Student Member, IEEE,* Zebo Zhou, *Member, IEEE,* Bin Gao, *Senior Member, IEEE,* Rui Li, *Member, IEEE,* Yuhua Cheng, *Senior Member, IEEE* and Hassen Fourati

***Abstract*—As a key problem for multi-sensor attitude determination, Wahba's problem has been studied for almost 50 years. Different from existing methods, this paper presents a novel linear approach to solving this problem. We name the proposed method the Fast Linear Attitude Estimator (FLAE) because it is faster than known representative algorithms. The original Wahba's problem is extracted to several 1-dimensional equations based on quaternions. They are then investigated with pseudo-inverse matrices establishing a linear solution to n-dimensional equations, which are equivalent to the conventional Wahba's problem. To obtain the attitude quaternion in a robust manner, an eigenvalue-based solution is proposed. Symbolic solutions to the corresponding characteristic polynomial is derived showing higher computation speed. Simulations are designed and conducted using test cases evaluated by several classical methods e.g. M. D. Shuster's QUaternion ESTimator (QUEST), F. L. Markley's SVD method, D. Mortari's Second Estimator of the Optimal Quaternion (ESOQ2) and some recent representative methods e.g. Y. Yang's analytical method and Riemannian manifold method. The results show that FLAE generates attitude estimates as accurate as that of several existing methods but consumes much less computation time (about 50% of the known best algorithm). Also, to verify the feasibility in embedded application, an experiment on the accelerometer-magnetometer combination is carried out where the algorithms are compared via C++ programming language. An extreme case is finally studied, revealing a minor improvement shows more effectiveness in this case inspired by Y. Cheng et al.**

***Note to Practitioners*—Attitude determination using vector observations can be applied in many areas. The most frequently involved are the accelerometer-magnetometer combination and star tracker array. Based on the proposed efficient FLAE algorithm, the time consumption of the sensor fusion can be significantly reduced, saving the execution time for fault detection, fail safe and etc.**

***Index Terms*—Wahba's Problem, Vector Observations, Attitude Determination, Attitude Quaternion, Eigenvalue Problem, Pseudo-inverse Matrix.**

This research was supported by National Natural Science Foundation of China under grant No. 41604025 and No. 61450010

Source codes of this paper can be acquired from https://github.com/zarathustr/FLAE and https://github.com/zarathustr/FLAE_cpp.

J. Wu is with School of Aeronautics and Astronautics, School of Automation of University of Electronic Science and Technology of China (UESTC). E-mail: jin_wu_uestc@hotmail.com

B. Gao, R. Li and Y. Cheng are with School of Automation, UESTC. E-mail: bin_gao@uestc.edu.cn; yhcheng@uestc.edu.cn; hitlirui@gmail.com;

H. Fourati is with the University Grenoble Alpes, CNRS, GIPSA-Lab, Grenoble 38400, France, and also with Inria, Grenoble. E-mail: hassen.fourati@gipsa-lab.grenoble-inp.fr

Correspondent: Z. Zhou is with School of Aeronautics and Astronautics, UESTC. E-mail: klinsmann.zhou@gmail.com

I. Introduction

ATTITUDE determination is a vital part of navigation technology [1], [2]. The Wahba's problem was first proposed by G. Wahba in 1965 with the aim to find the optimal attitude matrix (also called the Direction Cosine Matrix, DCM), from multi-sensor vector observations [3]. The Wahba's problem try to minimize the Wahba's loss function defined by

$$L(\mathbf{C}) = \frac{1}{2}\sum_{i=1}^{n} a_i \left\|\mathbf{b}_i - \mathbf{C}\mathbf{r}_i\right\|^2 \tag{1}$$

where $\mathbf{C}$ denotes the optimal DCM, $\|\cdot\|$ stands for the Euclidean norm. $a_i, \mathbf{b}_i, \mathbf{r}_i$ denote the i-th weight, normalized observation vectors in body frame b and reference frame r respectively. a_i is the positive weight of i-th vector observation pair satisfying

$$\sum_{i=1}^{n} a_i = 1 \tag{2}$$

which accounts for the relative precision of various sensors that usually based on the noise characters [4]. Note that commonly $n \geq 2$ since when there is a single vector observation pair, the attitude quaternion could be ambiguous [5]. There are considerable methods for solving this problem:

1) Classical Methods: The Wahba's loss function can be transformed into:

$$L(\mathbf{C}) = 1 - \sum_{i=1}^{n} a_i \mathbf{b}_i^T \mathbf{C}\mathbf{r}_i = 1 - tr(\mathbf{C}\mathbf{B}^T) \tag{3}$$

where $\mathbf{B}$ is defined by

$$\mathbf{B} = \sum_{i=1}^{n} a_i \mathbf{b}_i \mathbf{r}_i^T \tag{4}$$

A polar-decomposition solution is given by J. L. Farrell et al. in 1966 [6] which is not fast and robust [5] since it is actually an explicit pseudo-inverse solution to Wahba's problem. The presented approach consumes plenty of time to compute the generalized inverse and it requires orthogonality correction of the attitude matrix that may induce lack of optimality. Two years later, in 1968, P. B. Davenport [7] converted the Wahba's problem to an eigenvalue problem since the DCM is a homogenous quadratic function of the unit quaternion $\mathbf{q}_{\text{opt}}$ with the vector-wise form of

$$\mathbf{q}_{\text{opt}} = (\mathbf{n}\sin\frac{\theta}{2}, \cos\frac{\theta}{2})^T \tag{5}$$

where $\mathbf{n}$ denotes the vector part (rotation vector) and θ denotes the rotation angle around the rotation vector. By introducing

Lagrange multiplier λ, the Wahba's problem is transformed into [8]

$$\max_{\lambda,\mathbf{q}} \mathbf{q}^T\mathbf{K}\mathbf{q} - \lambda\mathbf{q}^T\mathbf{q} \tag{6}$$

This can be achieved by finding the largest positive eigenvalue and corresponding quaternion of $\mathbf{K}$ [9], [10] satisfying

$$\mathbf{K}\mathbf{q}_{opt} = \lambda_{\max}\mathbf{q}_{\text{opt}} \tag{7}$$

where

$$\begin{aligned} \mathbf{K} &= \begin{bmatrix} \mathbf{S} - \mathbf{I}_{3\times 3}tr(\mathbf{B}) & \mathbf{z} \\ \mathbf{z}^T & tr(\mathbf{B}) \end{bmatrix} \\ \mathbf{S} &= \mathbf{B} + \mathbf{B}^T \\ \mathbf{z} &= \sum_{i=1}^{n} a_i\mathbf{b}_i \times \mathbf{r}_i \end{aligned} \tag{8}$$

M. D. Shuster and et al. proposed their QUEST algorithm [11] in 1981. Firstly, Shuster gave a closed-form characteristic polynomial of $\mathbf{K}$ via Cayley-Hamilton theorem, such that

$$\lambda^4 - (a+b)\lambda^2 - c\lambda + (ab + c\sigma - d) = 0 \tag{9}$$

where

$$\begin{cases} \sigma = tr(\mathbf{B}) \\ \kappa = tr(adj(\mathbf{S})) \\ \Delta = \det(\mathbf{S}) \end{cases}, \begin{cases} a = \sigma^2 - \kappa \\ b = \sigma^2 + \mathbf{z}^T\mathbf{z} \\ c = \Delta + \mathbf{z}^T\mathbf{S}\mathbf{z} \\ d = \mathbf{z}^T\mathbf{S}^2\mathbf{z} \end{cases} \tag{10}$$

Secondly, it is found out that $\lambda_{\max}$ is very close to 1. Hence, starting from the initial value of $\lambda_{init} = 1$, the maximum eigenvalue can be calculated instantly using Newton algorithm within several iterations. Through real-world on-satellite evaluations, the QUEST is proved to be fast, accurate and robust [12], [13], [14]. In order to achieve the same goal of computing the largest eigenvalue of $\mathbf{K}$, F. L. Markley developed the Fast Optimal Attitude Matrix (FOAM) method in 1993 [15] where an equivalent characteristic polynomial of $\mathbf{K}$ is derived. In the same way, Newton algorithm is employed to calculate the maximum eigenvalue. This method is at least as fast as the QUEST algorithm.

Actually, before the FOAM, a robust and accurate Singular Value Decomposition (SVD) based method was also developed by F. L. Markley in 1988 [16] where matrix $\mathbf{B}$ in (8) has been discovered to have the following SVD:

$$\mathbf{B} = \mathbf{M}\Sigma\mathbf{V}^T = \mathbf{M}diag[\Sigma_{11}, \Sigma_{22}, \Sigma_{33}]\mathbf{V}^T \tag{11}$$

where $\mathbf{M}$ and $\mathbf{V}$ are proper orthogonal matrices with determinants of $+1$. Consequently, the optimal attitude matrix can be calculated by

$$\mathbf{C}_{opt} = \mathbf{M}diag[1, 1, (\det\mathbf{M})(\det\mathbf{V})]\mathbf{V}^T \tag{12}$$

This method does not rely on the Davenport's q-method. Since SVD has been widely recorgnized as a very robust numerical matrix operation, Markley's SVD method has the merit of robustness. The only drawback of this method is that SVD involves large time consumption. Besides, D. Mortari developed his two methods ESOQ and ESOQ2 to calculate the optimal quaternion both in 1997 [17], [18] . ESOQ includes another closed-form characteristic polynomial of $\mathbf{K}$ such that

$$\lambda^4 + a\lambda^3 + b\lambda^2 + c\lambda + d = 0 \tag{13}$$

where $a = 0, b = -2\{tr[\mathbf{B}]\}^2 + tr[adj(\mathbf{B} + \mathbf{B}^T)] - \mathbf{z}^T\mathbf{z}, c = -tr[adj(\mathbf{K})], d = \det(\mathbf{K})$. ESOQ2, as a suboptimal estimator, uses vector transformations to calculate the maximum eigenvalue of $\mathbf{K}$ iteratively and it is faster than ESOQ. As has been tested by F. L. Markley and D. Mortari, ESOQ2 is the fastest one among all the methods mentioned above [5]. However, ESOQ2 may not be as robust as methods like SVD, QUEST and FOAM because it employs some geometric approximations which would not be always precise.

To make the estimated attitude more smooth in the time domain, many Kalman Filter (KF) based methods are also developed e.g Filter QUEST, REcursive QUEST (REQUEST) and Q-method Extended Kalman Filter [19], [20], [21]. Yet the generalized Wahba's problem is considered to determine the attitude and spinning rate jointly in least squares. [22], [23].

2) Recently Advances: Above are classical methods for solving Wahba's problem. Recently, Y. Yang proposed his analytic approach to Wahba's problem [8] where an universal root-solving of quartic equation is utilized to give the maximum eigenvalue of $\mathbf{K}$. The algorithm is fast but remains a problem that it may have no real roots for the characteristic polynomial which would then lead to the occurrence of complex quaternions [10].

By using the Riemannian manifold, a more robust iterative method is also developed by Y. Yang [24]. The solving process is reduced to an iterative Newton-Riemann approach such that

$$(\mathbf{P}_{\mathbf{q}_k}\mathbf{K}\mathbf{P}_{\mathbf{q}_k} - \mathbf{q}_k^T\mathbf{K}\mathbf{q}_k\mathbf{I})\mathbf{y}_k = -\mathbf{P}_{\mathbf{q}_k}\mathbf{K}\mathbf{q}_k \tag{14}$$

where $\mathbf{P}_{\mathbf{q}_k} = (\mathbf{I} - \mathbf{q}_k\mathbf{q}_k^T)$. The searching procedure is finished by

$$\mathbf{q}_{k+1} = \frac{\mathbf{q}_k + \mathbf{y}_k}{\|\mathbf{q}_k + \mathbf{y}_k\|} \tag{15}$$

This method is verified to be at least as accurate as QUEST algorithm [24] but it may consume too many floating-point operations per second (FLOPS) in iterations.

Above are main classical and recently-developed methods that proposed by former researchers. Most of these methods are based on Davenport's q-method which results in the solution to the characteristic polynomial. In fact, some matrix operations such as getting the determinant and the adjoint matrix may be sophisticated for batch processing. We may also observe that it is difficult to achieve a balance between robustness and time consumption in terms of all existing methods because a fast one may not always be robust, vice versa.

Looking back to the Wahba's loss function in (1), it is equal to:

$$L(\mathbf{C}) = \frac{1}{2}\sum_{i=1}^{n} \|\sqrt{a_i}(\mathbf{b}_i - \mathbf{C}\mathbf{r}_i)\|^2 \tag{16}$$

Hence, minimizing (16) is equivalent to solving the following system

$$\begin{cases} \sqrt{a_1}(\mathbf{b}_1 - \mathbf{C}\mathbf{r}_1) = \mathbf{e}_1 \\ \sqrt{a_2}(\mathbf{b}_2 - \mathbf{C}\mathbf{r}_2) = \mathbf{e}_2 \\ \vdots \\ \sqrt{a_n}(\mathbf{b}_n - \mathbf{C}\mathbf{r}_n) = \mathbf{e}_3 \end{cases} \tag{17}$$

where $\mathbf{e}_i$ denotes the ith error item which ideally equals to $\mathbf{0}$. In most cases, it is non-zero and is induced by sensor biases

and stochastic noises [25]. Based on (17), it is able to find one reasonable solution to the system. Identically, the solution belongs to the optimal quaternion of the Wahba's problem. Following such motivation, this paper mainly contributes on the issues below:

1) The linear approach to (17) is derived, where pseudo-inverse matrices are adopted. An eigenvalue-based method is then designed to give the robust solution.
2) A much simplified form of characteristic polynomial is obtained. Then, two algorithms including Newton-iterative one and a symbolic one are given.
3) Simulations and real experiments are conducted to verify the low time consumption of the proposed FLAE. Comparisons with representative algorithms are detailed as well.

Based on the above contributions, this paper is structured with the following arrangement:

1) Section II: The 1-dimensional fusion equation analysis.
2) Section III: Using the similar methodology, n-dimensional equations are formed. Its solution is given and discussed. Error eigenvalue is introduced to enhance the robustness of the FLAE. We give a very simple closed-form of the characteristic polynomial. The symbolic solution to the characteristic polynomial is obtained.
3) Section IV: Simulations and real experiments are conducted to verify the performance of the FLAE.
4) Section VI: Concluding remarks of this paper.
5) Appendix A: Basic quaternion and rotation knowledge.
6) Appendix B: Derivation of $\mathbf{P}_D^{\dagger}$.

II. 1-DIMENSIONAL EQUATION

Assume that the sensor outputs can be rotated with DCM using

$$\mathbf{D}^b = \mathbf{C}\mathbf{D}^r \tag{18}$$

where

$$\begin{cases} \mathbf{D}^b = (D_x^b, D_y^b, D_z^b)^T \\ \mathbf{D}^r = (D_x^r, D_y^r, D_z^r)^T \end{cases} \tag{19}$$

are the observation vectors in body frame and reference frame respectively. $\mathbf{C}$ denotes the DCM. Since DCM is the function of quaternion, we define $\mathbf{f}_D(\mathbf{q})$ as

$$\mathbf{f}_D(\mathbf{q}) \triangleq \mathbf{C}\mathbf{D}^r - \mathbf{D}^b \tag{20}$$

where $\mathbf{q} = (q_0, q_1, q_2, q_3)^T$ with the scalar part of

$$q_0 = \cos\frac{\theta}{2} \tag{21}$$

and vector part of

$$\mathbf{n}\sin\frac{\theta}{2} = (q_1, q_2, q_3)^T \tag{22}$$

hereinafter. Our purpose is to minimize the selected loss function, hence the final condition can be given by

$$\mathbf{f}_D(\mathbf{q}) = \mathbf{0} \tag{23}$$

Now we can expand $\mathbf{f}_D(\mathbf{q})$ by

$$\begin{aligned} &\mathbf{f}_D(\mathbf{q}) = \mathbf{C}\mathbf{D}^r - \mathbf{D}^b \\ &= \begin{pmatrix} -D_x^b + D_x^r C_{11} + D_y^r C_{12} + D_z^r C_{13} \\ -D_y^b + D_x^r C_{21} + D_y^r C_{22} + D_z^r C_{23} \\ -D_z^b + D_x^r C_{31} + D_y^r C_{32} + D_z^r C_{33} \end{pmatrix} \\ &= D_x^r \begin{pmatrix} C_{11} \\ C_{21} \\ C_{31} \end{pmatrix} + D_y^r \begin{pmatrix} C_{12} \\ C_{22} \\ C_{32} \end{pmatrix} + D_z^r \begin{pmatrix} C_{13} \\ C_{23} \\ C_{33} \end{pmatrix} - \mathbf{D}^b \\ &= D_x^r \mathbf{C}_1 + D_y^r \mathbf{C}_2 + D_z^r \mathbf{C}_3 - \mathbf{D}^b \end{aligned} \tag{24}$$

$\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ are columns of the DCM and can be further given by

$$\begin{aligned} \mathbf{C}_1 &= \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^3 \\ 2q_1q_2 - 2q_0q_3 \\ 2q_0q_2 + q_1q_3 \end{pmatrix} \\ &= \begin{pmatrix} q_0 & q_1 & -q_2 & -q_3 \\ -q_3 & q_2 & q_1 & -q_0 \\ q_2 & q_3 & q_0 & q_1 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \mathbf{P}_1\mathbf{q} \end{aligned} \tag{25}$$

$$\begin{aligned} \mathbf{C}_2 &= \begin{pmatrix} 2q_1q_2 + q_0q_3 \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ -2q_0q_1 + 2q_2q_3 \end{pmatrix} \\ &= \begin{pmatrix} q_3 & q_2 & q_1 & q_0 \\ q_0 & -q_1 & q_2 & -q_3 \\ -q_1 & -q_0 & q_3 & q_2 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \mathbf{P}_2\mathbf{q} \end{aligned} \tag{26}$$

$$\begin{aligned} \mathbf{C}_3 &= \begin{pmatrix} -2q_0q_2 + 2q_1q_3 \\ 2q_2q_3 + 2q_0q_1 \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \\ &= \begin{pmatrix} -q_2 & q_3 & -q_0 & q_1 \\ q_1 & q_0 & q_3 & q_2 \\ q_0 & -q_1 & -q_2 & q_3 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \mathbf{P}_3\mathbf{q} \end{aligned} \tag{27}$$

respectively. Letting

$$\mathbf{P}_D = D_x^r \mathbf{P}_1 + D_y^r \mathbf{P}_2 + D_z^r \mathbf{P}_3 \tag{28}$$

, we obtain

$$\mathbf{f}_D(\mathbf{q}) = \mathbf{P}_D\mathbf{q} - \mathbf{D}^b \tag{29}$$

When $\mathbf{q}$ is optimal, it satisfies

$$\mathbf{P}_D\mathbf{q} = \mathbf{D}^b \Rightarrow \mathbf{q} = \mathbf{P}_D^{\dagger}\mathbf{D}^b \tag{30}$$

where $\mathbf{P}_D^{\dagger}$ is the pseudo-inverse of $\mathbf{P}_D$ if and only if it has full row rank. Commonly speaking, the analytic form of the pseudo-inverse of a given matrix is very complicated. However, according to our previous work [26], we have

$$\mathbf{P}_D^{\dagger} = \mathbf{P}_D^T = D_x^r \mathbf{P}_1^T + D_y^r \mathbf{P}_2^T + D_z^r \mathbf{P}_3^T \tag{31}$$

whose derivations are given in the Appendix B. Hence we transform the original quadratic system into a linear matrix equation. We can extract the $\mathbf{q}$ from the (30) by

$$\mathbf{P}_D^{\dagger}\mathbf{D}^b - \mathbf{q} = \mathbf{G}\mathbf{q} \tag{32}$$

Now we can solve the homogeneous matrix equation $\mathbf{G}\mathbf{q} = \mathbf{0}$ if and only if $\det(\mathbf{G}) = 0$. The fundamental solution system of

this equation can be given by elementary row transformations. Via Schmidt orthogonalization, the orthonormal quaternions can be computed.

III. n-DIMENSIONAL EQUATIONS

To be consistent with (17), assume that we have n observation equations, such that

$$\begin{cases} \sqrt{a_1}\mathbf{D}_1^b = \sqrt{a_1}\mathbf{C}\mathbf{D}_1^r \\ \sqrt{a_2}\mathbf{D}_2^b = \sqrt{a_2}\mathbf{C}\mathbf{D}_2^r \\ \vdots \\ \sqrt{a_n}\mathbf{D}_n^b = \sqrt{a_n}\mathbf{C}\mathbf{D}_n^r \end{cases} \tag{33}$$

Its error residual vector can be given by augmenting $\mathbf{f_D}(\mathbf{q})$

$$\mathbf{f}_{\sum \mathbf{D}}(\mathbf{q}) \triangleq \begin{pmatrix} \sqrt{a_1}(\mathbf{P}_{D_1}\mathbf{q} - \mathbf{D}_1^b) \\ \sqrt{a_2}(\mathbf{P}_{D_2}\mathbf{q} - \mathbf{D}_2^b) \\ \vdots \\ \sqrt{a_n}(\mathbf{P}_{D_n}\mathbf{q} - \mathbf{D}_n^b) \end{pmatrix} \tag{34}$$

When $\mathbf{f}_{\sum \mathbf{D}}(\mathbf{q}) = \mathbf{0}$, the equation satisfies

$$\begin{pmatrix} \sqrt{a_1}\mathbf{P}_{D_1} \\ \sqrt{a_2}\mathbf{P}_{D_2} \\ \vdots \\ \sqrt{a_n}\mathbf{P}_{D_n} \end{pmatrix} \mathbf{q} = \mathbf{P}_{\sum D}\mathbf{q} = \mathbf{D}_\Sigma^b = \begin{pmatrix} \sqrt{a_1}\mathbf{D}_1^b \\ \sqrt{a_2}\mathbf{D}_2^b \\ \vdots \\ \sqrt{a_n}\mathbf{D}_n^b \end{pmatrix} \tag{35}$$

Under this circumstance, the Moore-Penrose pseudo-inverse of $\mathbf{P}_{\sum D}$ is very difficult to compute. The reason is that the projections of each block of the matrix are required, which would significantly enlarge the space and time consumption [27]. However, we can transform the equation by $\mathbf{q}$ and $\mathbf{D}_\Sigma^b$'s pseudo-inverse matrices:

$$\mathbf{P}_{\sum D}\mathbf{q} = \mathbf{D}_{\sum}^b \Leftrightarrow \mathbf{P}_{\sum D} = \mathbf{D}_{\sum}^b \mathbf{q}^\dagger \Leftrightarrow (\mathbf{D}_{\sum}^b)^\dagger \mathbf{P}_{\sum D} = \mathbf{q}^\dagger \tag{36}$$

where $\dagger$ represents the Moore-Penrose pseudo-inverse. We use 'pseudo-inverse' as the abbreviation of Moore-Penrose pseudo-inverse hereinafter. $(\mathbf{D}_{\sum}^b)^\dagger$ and $\mathbf{q}^\dagger$ can be easily given by

$$\begin{aligned} &(\mathbf{D}_\Sigma^b)^\dagger = \begin{bmatrix} \sqrt{a_1}(\mathbf{D}_1^b)^T & \sqrt{a_2}(\mathbf{D}_2^b)^T & \cdots & \sqrt{a_n}(\mathbf{D}_n^b)^T \end{bmatrix} \\ &= (\mathbf{D}_\Sigma^b)^T = \begin{pmatrix} \sqrt{a_1}D_{x,1}^b, \sqrt{a_1}D_{y,1}^b, \sqrt{a_1}D_{z,1}^b, \\ \cdots, \sqrt{a_n}D_{x,n}^b, \sqrt{a_n}D_{y,n}^b, \sqrt{a_n}D_{z,n}^b \end{pmatrix} \\ &\mathbf{q}^\dagger = \mathbf{q}^T = (q_0, q_1, q_2, q_3) \end{aligned} \tag{37}$$

where the criteria of pseudo-inverse is involved, such that for arbitrarily given real matrix $\mathbf{p}$, its pseudo-inverse satisfies [28]

$$\begin{cases} \mathbf{p}\mathbf{p}^\dagger\mathbf{p} = \mathbf{p} \\ \mathbf{p}^\dagger\mathbf{p}\mathbf{p}^\dagger = \mathbf{p}^\dagger \\ (\mathbf{p}\mathbf{p}^\dagger)^T = \mathbf{p}\mathbf{p}^\dagger \\ (\mathbf{p}^\dagger\mathbf{p})^T = \mathbf{p}^\dagger\mathbf{p} \end{cases} \tag{38}$$

This leads to the fact that the pseudo-inverse of any normalized vector is its transpose. $\mathbf{P}_{\sum D}$ is derived by

$$\begin{aligned} &\mathbf{P}_{\sum D} = \begin{pmatrix} \sqrt{a_1}\mathbf{P}_{D_1} \\ \sqrt{a_2}\mathbf{P}_{D_2} \\ \vdots \\ \sqrt{a_n}\mathbf{P}_{D_n} \end{pmatrix} = \begin{pmatrix} \sqrt{a_1}(D_{x,1}^r\mathbf{P}_1 + D_{y,1}^r\mathbf{J}_2 + D_{z,1}^r\mathbf{P}_3) \\ \sqrt{a_2}(D_{x,2}^r\mathbf{P}_1 + D_{y,2}^r\mathbf{J}_2 + D_{z,2}^r\mathbf{P}_3) \\ \vdots \\ \sqrt{a_n}(D_{x,n}^r\mathbf{P}_1 + D_{y,n}^r\mathbf{J}_2 + D_{z,n}^r\mathbf{P}_3) \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{a_1}D_{x,1}^r\mathbf{I} \\ \sqrt{a_2}D_{x,2}^r\mathbf{I} \\ \vdots \\ \sqrt{a_n}D_{x,n}^r\mathbf{I} \end{pmatrix}\mathbf{P}_1 + \begin{pmatrix} \sqrt{a_1}D_{y,1}^r\mathbf{I} \\ \sqrt{a_2}D_{y,2}^r\mathbf{I} \\ \vdots \\ \sqrt{a_n}D_{y,n}^r\mathbf{I} \end{pmatrix}\mathbf{P}_2 + \begin{pmatrix} \sqrt{a_1}D_{z,1}^r\mathbf{I} \\ \sqrt{a_2}D_{z,2}^r\mathbf{I} \\ \vdots \\ \sqrt{a_n}D_{z,n}^r\mathbf{I} \end{pmatrix}\mathbf{P}_3 \\ &= \mathbf{U}_{D_x}\mathbf{P}_1 + \mathbf{U}_{D_y}\mathbf{P}_2 + \mathbf{U}_{D_z}\mathbf{P}_3 \end{aligned} \tag{39}$$

where $\mathbf{U}_{D_x}, \mathbf{U}_{D_y}, \mathbf{U}_{D_z}$ are $3n \times 3$ matrices. Hence

$$(\mathbf{D}_{\sum}^b)^\dagger \mathbf{P}_{\sum D} = (\mathbf{D}_{\sum}^b)^\dagger \mathbf{U}_{D_x}\mathbf{P}_1 + (\mathbf{D}_{\sum}^b)^\dagger \mathbf{U}_{D_y}\mathbf{P}_2 + (\mathbf{D}_{\sum}^b)^\dagger \mathbf{U}_{D_z}\mathbf{P}_3 \tag{40}$$

Let

$$\begin{aligned} \mathbf{H}_x &= (\mathbf{D}_{\sum}^b)^\dagger \mathbf{U}_{D_x} = (H_{x1}, H_{x2}, H_{x3}) \\ \mathbf{H}_y &= (\mathbf{D}_{\sum}^b)^\dagger \mathbf{U}_{D_y} = (H_{y1}, H_{y2}, H_{y3}) \\ \mathbf{H}_z &= (\mathbf{D}_{\sum}^b)^\dagger \mathbf{U}_{D_z} = (H_{z1}, H_{z2}, H_{z3}) \end{aligned} \tag{41}$$

It is noted that $\mathbf{H}_x, \mathbf{H}_y, \mathbf{H}_z$ are 1×3 matrices and can be calculated by

$$\begin{aligned} \mathbf{H}_x &= \begin{pmatrix} \sum_{i=1}^n a_i D_{x,i}^r D_{x,i}^b & \sum_{i=1}^n a_i D_{x,i}^r D_{y,i}^b & \sum_{i=1}^n a_i D_{x,i}^r D_{z,i}^b \end{pmatrix} \\ \mathbf{H}_y &= \begin{pmatrix} \sum_{i=1}^n a_i D_{y,i}^r D_{x,i}^b & \sum_{i=1}^n a_i D_{y,i}^r D_{y,i}^b & \sum_{i=1}^n a_i D_{y,i}^r D_{z,i}^b \end{pmatrix} \\ \mathbf{H}_z &= \begin{pmatrix} \sum_{i=1}^n a_i D_{z,i}^r D_{x,i}^b & \sum_{i=1}^n a_i D_{z,i}^r D_{y,i}^b & \sum_{i=1}^n a_i D_{z,i}^r D_{z,i}^b \end{pmatrix} \end{aligned} \tag{42}$$

The fusion equation finally arrives at

$$\mathbf{H}_x\mathbf{P}_1 + \mathbf{H}_y\mathbf{P}_2 + \mathbf{H}_z\mathbf{P}_3 - \mathbf{q}^\dagger = \mathbf{0}_{1\times 4} \tag{43}$$

Expand the equation with transpose operation, we obtain

$$\begin{aligned} &\mathbf{P}_1^T\mathbf{H}_x^T + \mathbf{P}_2^T\mathbf{H}_y^T + \mathbf{P}_3^T\mathbf{H}_z^T - \mathbf{q} = \mathbf{0} \\ &\Leftrightarrow (\mathbf{W} - \mathbf{I})\mathbf{q} = \mathbf{0} \end{aligned} \tag{44}$$

where $\mathbf{W}$'s elements are given by

$$\begin{aligned} \mathbf{W}_{1,1} &= H_{x1} + H_{y2} + H_{z3} \\ \mathbf{W}_{1,2} &= -H_{y3} + H_{z2} \\ \mathbf{W}_{1,3} &= -H_{z1} + H_{x3} \\ \mathbf{W}_{1,4} &= -H_{x2} + H_{y1} \\ \mathbf{W}_{2,1} &= -H_{y3} + H_{z2} \\ \mathbf{W}_{2,2} &= H_{x1} - H_{y2} - H_{z3} \\ \mathbf{W}_{2,3} &= H_{x2} + H_{y1} \\ \mathbf{W}_{2,4} &= H_{x3} + H_{z1} \\ \mathbf{W}_{3,1} &= -H_{z1} + H_{x3} \\ \mathbf{W}_{3,2} &= H_{x2} + H_{y1} \\ \mathbf{W}_{3,3} &= H_{y2} - H_{x1} - H_{z3} \\ \mathbf{W}_{3,4} &= H_{y3} + H_{z2} \\ \mathbf{W}_{4,1} &= -H_{x2} + H_{y1} \\ \mathbf{W}_{4,2} &= H_{x3} + H_{z1} \\ \mathbf{W}_{4,3} &= H_{y3} + H_{z2} \\ \mathbf{W}_{4,4} &= H_{z3} - H_{y2} - H_{x1} \end{aligned} \tag{45}$$

However, what needs to be pointed out is that $\mathbf{W} - \mathbf{I}$ may suffer from rank-deficient problem in terms of the uncertainties

inside the sensor outputs e.g. biases and stochastic noises. Consequently, this problem should be solved in a more robust way. Note that (44) can be transformed by adding a small quaternion error $\varepsilon\mathbf{q}$:

$$\mathbf{W}\mathbf{q} = (1+\varepsilon)\mathbf{q} \tag{46}$$

where ε denotes the error factor. Apparently, $1+\varepsilon$ is an eigenvalue of $\mathbf{W}$. Therefore the problem is shifted to finding the eigenvalue that is closest to 1. (46) has very similar form with QUEST but in his method the eigenvalue has no obvious meaning yet the quaternion implemented here is in the standard form which is quite different from that in [11].

To calculate the eigenvalue of the matrix $\mathbf{W}$, the characteristic polynomial of $\mathbf{W}$ is defined as

$$f(\lambda) \triangleq \det(\mathbf{W} - \lambda\mathbf{I}_{4\times4}) \tag{47}$$

This polynomial can be then derived to

$$f(\lambda) = \lambda^4 + \tau_1\lambda^2 + \tau_2\lambda + \tau_3 \tag{48}$$

where τ_1, τ_2, τ_3 can be given by

$$\begin{aligned}
\tau_1 =& -2\left(\begin{array}{l} H_{x1}^2 + H_{x2}^2 + H_{x3}^2 + H_{y1}^2 \\ +H_{y2}^2 + H_{y3}^2 + H_{z1}^2 + H_{z2}^2 + H_{z3}^2 \end{array}\right) \\
\tau_2 =& 8(H_{x3}H_{y2}H_{z1} - H_{x2}H_{y3}H_{z1} - H_{x3}H_{y1}H_{z2} \\
&+H_{x1}H_{y3}H_{z2} + H_{x2}H_{y1}H_{z3} - H_{x1}H_{y2}H_{z3}) \\
\tau_3 =& \det(\mathbf{W})
\end{aligned} \tag{49}$$

After τ_1, τ_2, τ_3 is computed, the four-order characteristic polynomial can be solved with Newton iterative method. The derivative of $f(\lambda)$ with respect to the eigenvalue λ can be calculated by

$$f'(\lambda) = 4\lambda^3 + 2\tau_1\lambda + \tau_2 \tag{50}$$

Since λ is very close to 1, its initial value can be set to $\lambda_{\text{init}} = 1$. Then the Newton iteration can be conducted using

$$\lambda(n+1) = \lambda(n) - \frac{f[\lambda(n)]}{f'[\lambda(n)]} \tag{51}$$

Commonly, λ could be very accurate after several iterations. When the eigenvalue is obtained, the eigenvector can be then calculated using elementary row operations.

However, as the accuracy is not linear with iteration times, fixed iteration times will not always achieve good results. Hereby, a novel symbolic approach is investigated. The solutions to (47) are given as follows:

$$\begin{aligned}
\lambda_1 &= \tfrac{1}{2\sqrt{6}}\left(T_2 - \sqrt{-T_2^2 - 12\tau_1 - \tfrac{12\sqrt{6}\tau_2}{T_2}}\right) \\
\lambda_2 &= \tfrac{1}{2\sqrt{6}}\left(T_2 + \sqrt{-T_2^2 - 12\tau_1 - \tfrac{12\sqrt{6}\tau_2}{T_2}}\right) \\
\lambda_3 &= -\tfrac{1}{2\sqrt{6}}\left(T_2 + \sqrt{-T_2^2 - 12\tau_1 + \tfrac{12\sqrt{6}\tau_2}{T_2}}\right) \\
\lambda_4 &= -\tfrac{1}{2\sqrt{6}}\left(T_2 - \sqrt{-T_2^2 - 12\tau_1 + \tfrac{12\sqrt{6}\tau_2}{T_2}}\right)
\end{aligned} \tag{52}$$

with the parameters of

$$\begin{aligned}
T_0 &= 2\tau_1^3 + 27\tau_2^2 - 72\tau_1\tau_3 \\
T_1 &= \left(T_0 + \sqrt{-4(\tau_1^2 + 12\tau_3)^3 + T_0^2}\right)^{\frac{1}{3}} \\
T_2 &= \sqrt{-4\tau_1 + \frac{2^{\frac{4}{3}}(\tau_1^2 + 12\tau_3)}{T_1} + 2^{\frac{2}{3}}T_1}
\end{aligned} \tag{53}$$

The analytic computation and validation script can be acquired from **https://github.com/zarathustr/FLAE_mathematica**.

Then, λ is chosen by the value which is nearest to 1. In this way, the solving process of λ is significantly shortened. Let $\mathbf{N}$ be

$$\mathbf{N} = \mathbf{W} - \lambda\mathbf{I}_{4\times4} \tag{54}$$

Via elementary row operations, $\mathbf{N}$ can be transformed to

$$\mathbf{N} \rightarrow \mathbf{N}' = \begin{pmatrix} 1 & 0 & 0 & \chi \\ 0 & 1 & 0 & \rho \\ 0 & 0 & 1 & \upsilon \\ 0 & 0 & 0 & \zeta \end{pmatrix} \tag{55}$$

where ζ is a very tiny number (usually at the level of 1×10^{-15}) for the rank of matrix $\mathbf{W} - \lambda\mathbf{I}_{4\times4}$ is not full. To ensure such equation

$$(\mathbf{W} - \lambda\mathbf{I}_{4\times4})\mathbf{q} = \mathbf{0} \tag{56}$$

has non-zero and unique solution, ζ is chosen to be 0 because it is too small to take any effect. Hence, $\mathbf{N}$ arrives at

$$\mathbf{N}' = \begin{pmatrix} 1 & 0 & 0 & \chi \\ 0 & 1 & 0 & \rho \\ 0 & 0 & 1 & \upsilon \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{57}$$

Let $q_3 = -1$, we obtain the following fundamental solution system

$$\tilde{\mathbf{q}} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \chi \\ \rho \\ \upsilon \\ -1 \end{pmatrix} \tag{58}$$

which can be normalized with the norm of

$$\|\tilde{\mathbf{q}}\| = \sqrt{\chi^2 + \rho^2 + \upsilon^2 + 1} \tag{59}$$

The calculation step are listed by:

Algorithm 1 Calculation procedure of FLAE with symbolic solutions to characteristic polynomial

1: Initialize the estimator with pre-determined weights.
2: Obtain sensor outputs and normalize them into vector observations.
3: Calculate $\mathbf{H}_x, \mathbf{H}_y, \mathbf{H}_z$ with (42).
4: Calculate τ_1, τ_2, τ_3 with (49).
5: Compute eigenvalue that is closest to 1 with (52).
6: Perform elementary row operations on $\mathbf{N}$ to the final form as denoted in (55).
7: Calculate the optimal unit quaternion using (58) and (59).
8: If possible, calculate the Euler angles.

The overall flowchart of the FLAE is given in Figure 1.

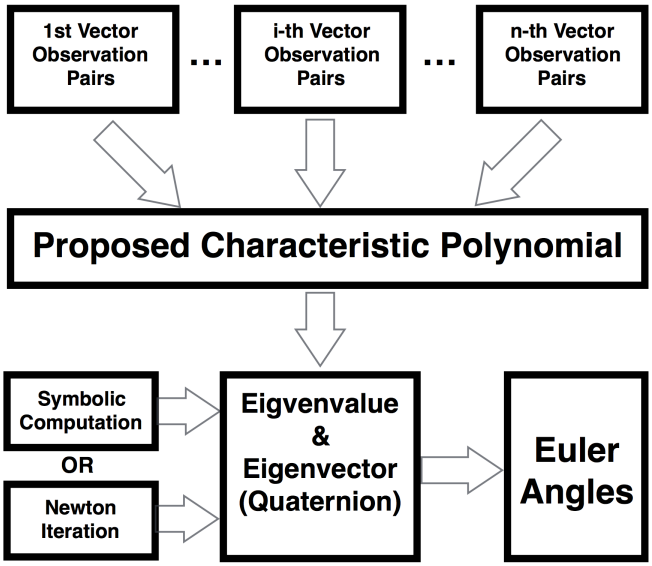


Fig. 1: Flowchart of the FLAE's calculation procedure.

IV. Simulations and Experiments

A. Monte-Carlo Simulation Results

1) Studied Cases: F. L. Markley designed 12 test cases in his famous publication [15], which have also been evaluated by several other existing methods e.g. ESOQ, ESOQ2, Y. Yang's analytical method and Riemannian manifold method. Hence, to create the simulational samples of the proposed method, we also choose these cases. The 12 cases with their details are listed in Table (III). Monte-Carlo simulation samples are created using

$$\mathbf{b}_i = \mathbf{C}_{true}\mathbf{r}_i + \mathbf{n}_i \tag{60}$$

where $\mathbf{n}_i$ denotes the Gaussian measurement noise with the standard deviation of σ_i. $\mathbf{C}_{true}$ represents the referenced DCM and is defined by

$$\mathbf{C}_{true} = \begin{pmatrix} 0.352 & 0.864 & 0.360 \\ -0.864 & 0.152 & 0.480 \\ 0.360 & -0.480 & 0.800 \end{pmatrix} \tag{61}$$

whose corresponding true Euler angle vector is calculated by $(-30.97°, -21.10°, -67.83°)^T$,in X-Y-Z rotation sequence. The estimation error is defined by [15]

$$\phi_{err} = 2\arcsin\frac{\|\mathbf{C}_{opt} - \mathbf{C}_{true}\|}{\sqrt{8}} \tag{62}$$

For each case, a test set with 10000 samples is produced for investigation. The MATLAB R2015b scientific computation software is applied for carrying out all the simulations in this section. The computer owns the configuration of a 4-core Intel i7 2.2GHz CPU, 16G RAM and a 512G solid state disk. The time consumption of different algorithms is collected using the internal timer of the MATLAB software.

2) Comparison with Representative Methods: QUEST, ESOQ2, F. L. Markley's SVD method, Y. Yang's analytical method and Riemannian manifold method are compared with the proposed method in this section. Related results of these numerical simulations are given in Table (I) and (II).

Meanwhile, using Case 3, the simulated Euler angles, attitude errors, values of loss function are shown in Figure 2, 3, 4 respectively. In these figures, the blue, red and yellow lines correspond to x, y, z components of the calculated Euler angles respectively.

TABLE III: Studied Cases

Case	Observation vectors in the reference frame
1	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [0,1,0]^T, \mathbf{r}_3 = [0,0,1]^T$
2	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [0,1,0]^T$
3	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [0,1,0]^T, \mathbf{r}_3 = [0,0,1]^T$
4	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [0,1,0]^T$
5	$\mathbf{r}_1 = [0.6,0.8,0]^T, \mathbf{r}_2 = [0.8,-0.6,0]^T$
6	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [1,0.01,0]^T, \mathbf{r}_3 = [1,0,0.01]^T$
7	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [1,0.01,0]^T$
8	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [1,0.01,0]^T, \mathbf{r}_3 = [1,0,0.01]^T$
9	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [1,0.01,0]^T$
10	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [0.96,0.28,0]^T, \mathbf{r}_3 = [0.96,0,0.28]^T$
11	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [0.96,0.28,0]^T$
12	$\mathbf{r}_1 = [1,0,0]^T, \mathbf{r}_2 = [0.96,0.28,0]^T$

Case	Measurement standard deviations
1	$\sigma_1 = 10^{-6}, \sigma_2 = 10^{-6}, \sigma_3 = 10^{-6}$
2	$\sigma_1 = 10^{-6}, \sigma_2 = 10^{-6}$
3	$\sigma_1 = 0.01, \sigma_2 = 0.01, \sigma_3 = 0.01$
4	$\sigma_1 = 0.01, \sigma_2 = 0.01$
5	$\sigma_1 = 10^{-6}, \sigma_2 = 0.01$
6	$\sigma_1 = 10^{-6}, \sigma_2 = 10^{-6}, \sigma_3 = 10^{-6}$
7	$\sigma_1 = 10^{-6}, \sigma_2 = 10^{-6}$
8	$\sigma_1 = 0.01, \sigma_2 = 0.01, \sigma_3 = 0.01$
9	$\sigma_1 = 0.01, \sigma_2 = 0.01$
10	$\sigma_1 = 10^{-6}, \sigma_2 = 0.01, \sigma_3 = 0.01$
11	$\sigma_1 = 10^{-6}, \sigma_2 = 0.01$
12	$\sigma_1 = 0.01, \sigma_2 = 10^{-6}$

TABLE IV: Time Consumption and Error Variance (Case 3)

Method	Time Consumption	Attitude Error Variance (deg^2)
QUEST	4.562543×10^{-5}s	0.237242473
ESOQ2	4.475527×10^{-5}s	0.237709058
SVD	4.703876×10^{-5}s	0.237242473
Riemannian Manifold	1.590334×10^{-4}s	0.237242473
Analytical Method	9.569700×10^{-5}s	0.237242473
The Proposed FLAE	2.110286×10^{-5}s	0.237242473

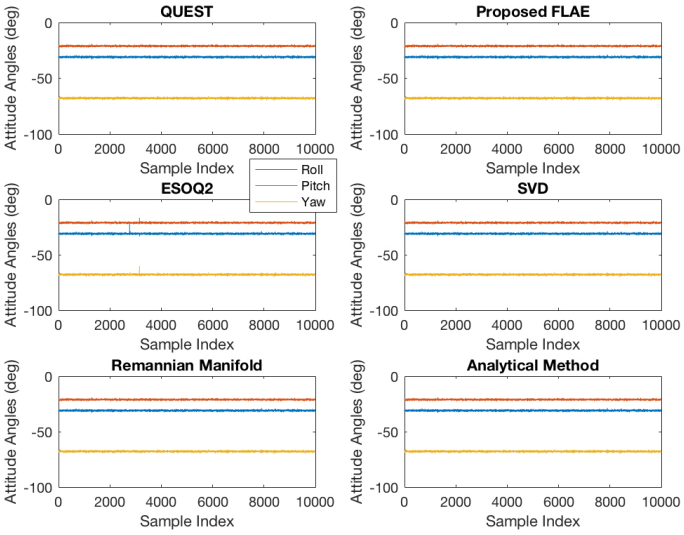


Fig. 2: Euler angles calculated using Case 3.

TABLE I: Estimation Error ϕ_{err}

Case	QUEST	SVD	ESOQ2	Analytical Method	Riemannian Manifold	Proposed FLAE
1	9.101×10^{-7}	9.101×10^{-7}	1.102	9.058×10^{-7}	9.101×10^{-7}	9.101×10^{-7}
2	1.083×10^{-6}	1.083×10^{-6}	1.102	1.083×10^{-6}	1.083×10^{-6}	1.083×10^{-6}
3	8.965×10^{-3}	8.965×10^{-3}	1.102	8.965×10^{-3}	8.965×10^{-3}	8.965×10^{-3}
4	1.067×10^{-2}	1.067×10^{-2}	1.102	1.067×10^{-2}	1.067×10^{-2}	1.067×10^{-2}
5	7.027×10^{-3}	7.027×10^{-3}	1.229	7.027×10^{-3}	7.027×10^{-3}	7.027×10^{-3}
6	5.603×10^{-5}	5.603×10^{-5}	1.509	5.603×10^{-5}	5.603×10^{-5}	5.603×10^{-5}
7	8.450×10^{-5}	8.450×10^{-5}	1.102	8.450×10^{-5}	8.450×10^{-5}	8.450×10^{-5}
8	0.5410	0.5410	1.322	0.5410	0.5410	0.5410
9	0.6757	0.6757	1.230	0.6757	0.6757	0.6757
10	1.789×10^{-2}	1.789×10^{-2}	1.105	1.789×10^{-2}	1.789×10^{-2}	1.789×10^{-2}
11	2.118×10^{-2}	2.118×10^{-2}	1.102	2.118×10^{-2}	2.118×10^{-2}	2.118×10^{-2}
12	2.086×10^{-2}	2.086×10^{-2}	1.102	2.086×10^{-2}	2.086×10^{-2}	2.086×10^{-2}

TABLE II: Loss Function

Case	QUEST	SVD	ESOQ2	Analytical Method	Riemannian Manifold	Proposed FLAE
1	9.592×10^{-13}	9.592×10^{-13}	3.536×10^{-9}	9.592×10^{-13}	9.592×10^{-13}	9.592×10^{-13}
2	7.419×10^{-13}	7.419×10^{-13}	8.730×10^{-9}	7.419×10^{-13}	7.419×10^{-13}	7.419×10^{-13}
3	9.298×10^{-5}	9.298×10^{-5}	2.300×10^{-4}	9.298×10^{-5}	9.298×10^{-5}	9.298×10^{-5}
4	7.179×10^{-5}	7.179×10^{-5}	1.715×10^{-4}	7.179×10^{-5}	7.179×10^{-5}	7.179×10^{-5}
5	3.569×10^{-5}	3.569×10^{-5}	4.592×10^{-5}	3.569×10^{-5}	3.569×10^{-5}	3.569×10^{-5}
6	9.667×10^{-13}	9.667×10^{-13}	1.833×10^{-5}	9.667×10^{-13}	9.667×10^{-13}	9.667×10^{-13}
7	7.556×10^{-13}	7.556×10^{-13}	5.004×10^{-5}	7.556×10^{-13}	7.556×10^{-13}	7.5556×10^{-13}
8	9.228×10^{-5}	9.228×10^{-5}	8.075×10^{-5}	9.228×10^{-5}	9.228×10^{-5}	9.228×10^{-5}
9	7.338×10^{-5}	7.338×10^{-5}	1.469×10^{-4}	7.338×10^{-5}	7.338×10^{-5}	7.338×10^{-5}
10	6.069×10^{-5}	6.069×10^{-5}	5.571×10^{-4}	6.069×10^{-5}	6.069×10^{-5}	6.069×10^{-5}
11	3.607×10^{-5}	3.607×10^{-5}	1.019×10^{-4}	3.607×10^{-5}	3.607×10^{-5}	3.607×10^{-5}
12	3.484×10^{-5}	3.484×10^{-5}	7.213×10^{-5}	3.484×10^{-5}	3.484×10^{-5}	3.484×10^{-5}

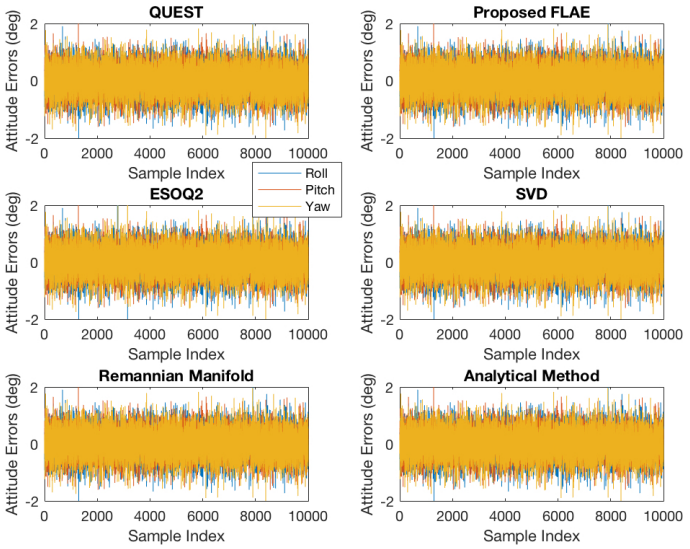


Fig. 3: Attitude error of different algorithms using Case 3.

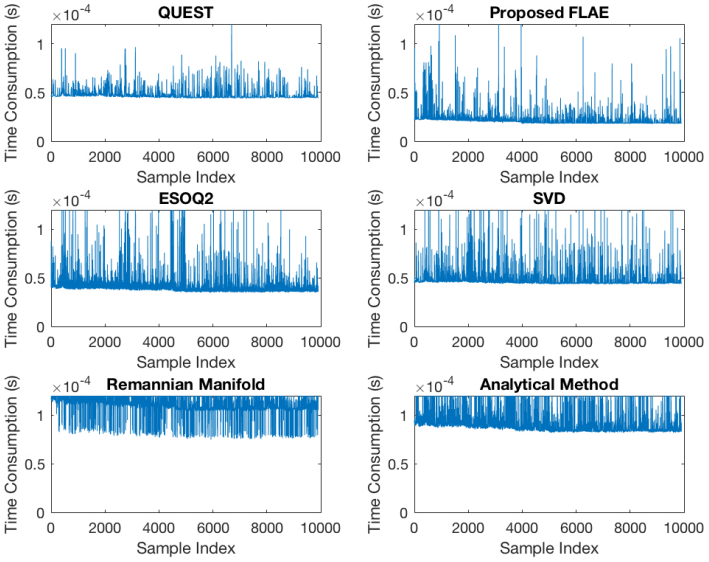


Fig. 5: The time consumption of various algorithms.

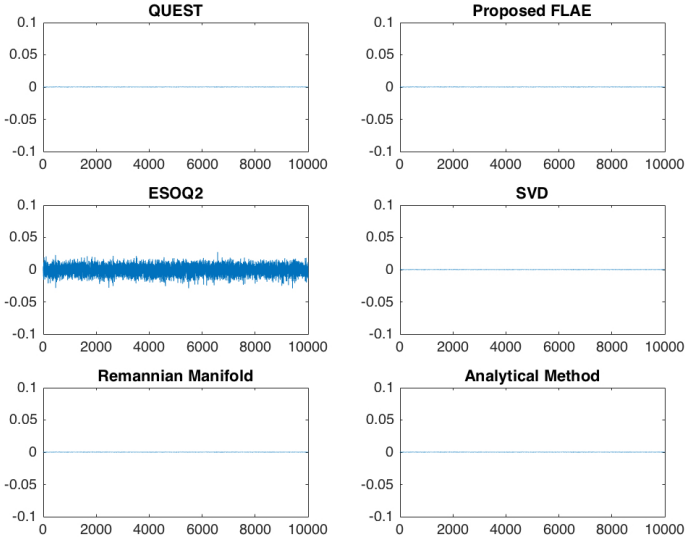


Fig. 4: Loss function in (1) using Case 3.

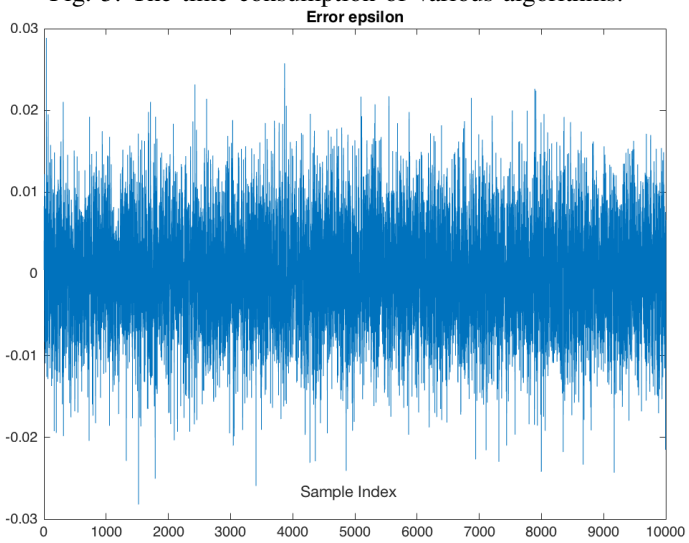


Fig. 6: Error factor ε values for Case 3.

From Table (I) and (II), the following information can be obtained:

1) The estimation errors and values of loss function of QUEST, SVD, Analytical method and Riemannian manifold method are basically the same with the proposed FLAE.
2) D. Mortari's ESOQ2 owns the worst estimation error and values of loss function which coincide with the results in [8].

This indicates that the proposed FLAE has the same estimation performance with existing methods. In fact, these methods have reached the extreme accuracy of the Wahba's problem. With different mathematical tools, although various algorithms have various forms, the results are basically the same.

For Case 3, time consumption of different algorithms can be sorted as: FLAE $<$ ESOQ2 $<$ QUEST $<$ SVD $<$ Analytical Method $<$ Riemannian manifold method. Besides, the average time consumption of the proposed FLAE is almost 47% of that of ESOQ2, an algorithm that is formerly recognized as the fastest one. This proves the proposed FLAE has a big advance. Moreover, the plot of error ε for Case 3 is given in Figure 6. As is consistent with the assumption, they are very small digits around 0.

All the performances tell us the proposed FLAE maintains the same estimation abilities including estimation error and value of Wahba's loss function but it is significantly better than other main existing methods on time consumption.

B. Discussion on FLAE's Efficiency

Above materials show that the proposed FLAE is at least as accurate as existing algorithms for Wahba's solution. The main advance is that it can compute the attitude from vector observations with even less time consumption. Now, the calculation step is categorized into two groups:

1) The calculation of characteristic polynomial.
2) Eigenvalue computation from characteristic polynomial and computation of eigenvector.

In this section, we mainly discuss on the comparisons between QUEST and the proposed FLAE. First, let us see the calculation process of characteristic polynomials from the two algorithms. Using (9) and (48), the polynomials are obtained. We can see that QUEST's polynomial requires much more matrix operations than that of FLAE. This is validated with Monte-Carlo simulation. The results are given in Figure 7.

When the calculation of polynomial is finished, the eigenvalue can be obtained by two means: the Newton iteration and symbolic computation. The proposed FLAE, as described in Algorithm 1, uses the symbolic method to calculate the eigenvalue. Needless to say, symbolic computation is faster than most iterative methods. Monte-Carlo simulation is also conducted containing a large simulation set with 100000 random vector observation pairs. The following figure shows that when the iteration accuracy is set to 1×10^{-5}, denoting a loose accuracy indicator for single precision numbers, the time consumption of FLAE (using symbolic computation) is always less than that of QUEST.

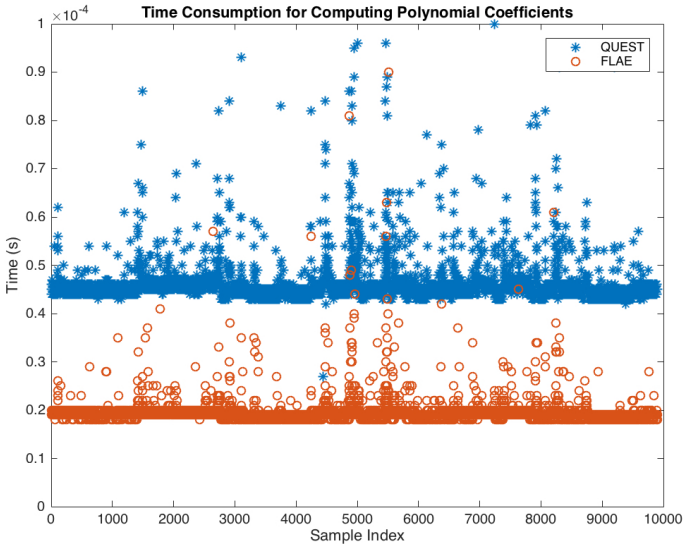


Fig. 7: Time consumption for computing polynomial coefficients.

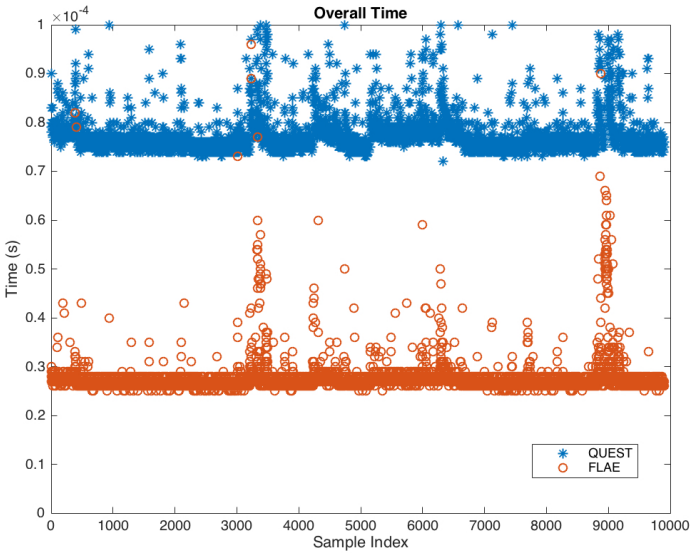


Fig. 8: Time consumption of QUEST and FLAE (using symbolic computation).

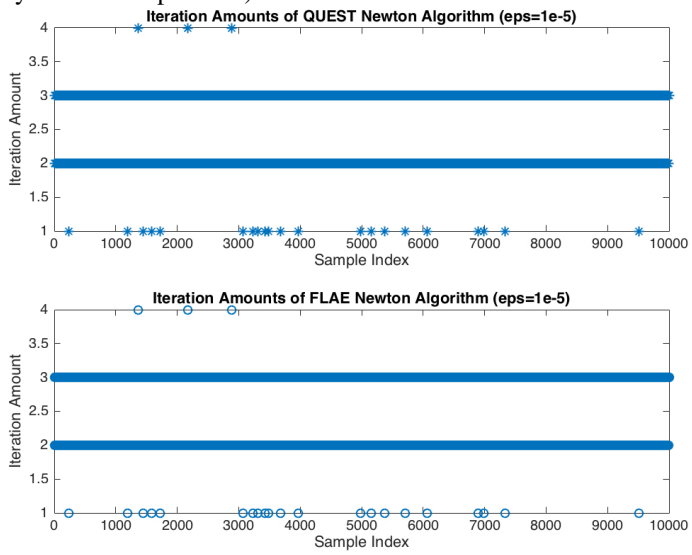


Fig. 9: Iteration amounts of two algorithms when Newton algorithm is applied.

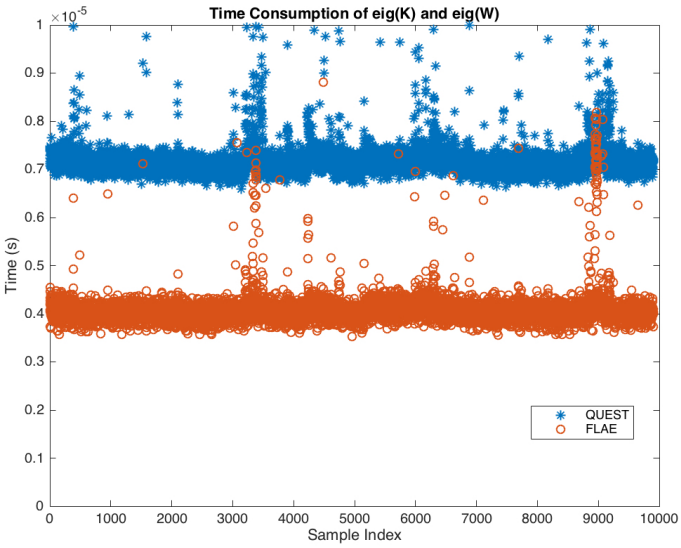


Fig. 10: Time consumption of eigen-decomposition of $\mathbf{K}$ and $\mathbf{W}$.

However, if the users choose Newton iteration for FLAE's computation, there is no doubt that the iteration amount would be similar with that of QUEST. The iteration amounts of the two algorithms are given in Figure 9.

Besides, there are some properties inside the two algorithms. The key to QUEST is the determination of matrix $\mathbf{K}$. For FLAE, $\mathbf{W}$ plays the same role. The target task is to get the eigenvalues from the two matrices. The following figure (Figure 10) shows the time consumption when eigenvalue decomposition is applied to $\mathbf{K}$ and $\mathbf{W}$. According to MATLAB's official document, it is implemented using QR decomposition for most small-matrix cases. We can see that $\mathbf{W}$ can be decomposed with less time consumption i.e. $\mathbf{W}$'s internal makes it more flexible when numerical algorithm is applied on it. The audience can download the source code files with comparisons from **https://github.com/zarathustr/FLAE** for further information.

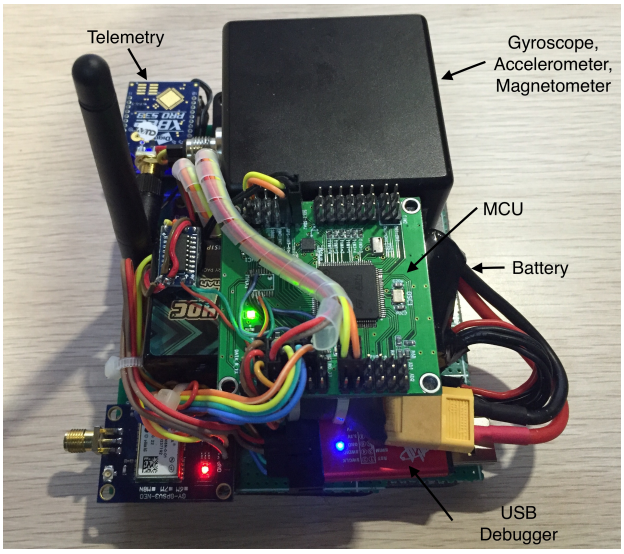


Fig. 11: Experiment platform configurations.

C. Experiment on eCompass

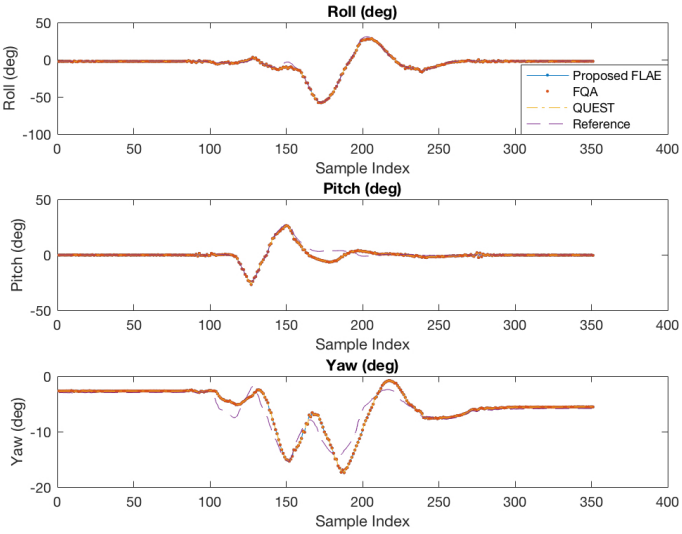


Fig. 12: The Euler angles from the proposed FLAE, FQA, QUEST and golden reference are plotted.

The conventional Wahba's problem is often adopted to get the optimal attitude matrix using star tracker outputs [29], [30], [31]. In the past decade, not only has the Wahba's problem been applied to the satellite attitude determination system, but it has also been used by a variety of applications related to multisensor-based attitude estimation [32]. For instance, the accelerometer-magnetometer combination, also referred to as the eCompass [33], has been widely used [34], [35], [36], [37], [38]. Its basic fusion equations can be given by

$$\begin{cases} \mathbf{A}^b = \mathbf{C}\mathbf{A}^r \\ \mathbf{M}^b = \mathbf{C}\mathbf{M}^r \end{cases} \tag{63}$$

where $\mathbf{A}^b, \mathbf{M}^b$ denote the observation vectors of the accelerometer and magnetometer in the body frame, such that

$$\begin{cases} \mathbf{A}^b = (a_x, a_y, a_z)^T \\ \mathbf{M}^b = (m_x, m_y, m_z)^T \end{cases} \tag{64}$$

while the observation vectors in the reference frame (NED frame) can be given by

$$\begin{cases} \mathbf{A}^r = (0, 0, 1)^T \\ \mathbf{M}^r = (m_N, 0, m_D)^T \end{cases} \tag{65}$$

To verify the feasibility of the proposed FLAE, an eCompass (actually the accelerometer-magnetometer combination) ADIS16405 produced by Analog Devices Inc. is adopted for real experiments. Our hardware is shown in Figure 11. The experiment is conducted at Wuxi, China with longitude of E-120.299° and latitude of N-31.568° with the local magnetometer's reference vector of $\mathbf{M}^r = (0.60311, 0, -0.79766)^T$. The weights of the accelerometer and magnetometer are set as 0.5 and 0.5 respectively due to the noise scales of the sensors. A golden reference Microstrain 3DM-GX3-25 Attitude and Heading Reference System (AHRS) is firmly attached to the

experiment platform to give the reference Euler angles.

Experiment results are given in Figure 12 where the results from FQA, QUEST and golden reference are plotted synchronously. From Figure 12, we can see that the proposed FLAE can estimate the attitude angles at exactly the same accuracy with representative algorithms. Yet the results reflect the motion properly. In this experiment, the mean attitude errors for roll, pitch and yaw are: $-0.3530°$, $-0.3218°$ and $-0.3218°$ respectively.

In engineering practice, the engineers are always tend to use programming languages like C/C++ to implement algorithms. Using the designed hardware, the QUEST and proposed FLAE are executed batchedly. When utilized on STM32F407VET6 microcontroller, the proposed FLAE can run at the frequency of 4000Hz while that of FQA, QUEST and ESOQ2 are 3083Hz, 2214Hz and 2752Hz. FQA is based on geometric relationships and it requires singularity avoidance which is not needed by FLAE. QUEST and ESOQ2 consume more matrix operations than FLAE. These factors lead to the advantage of the proposed FLAE which also prove its extreme low time consumption. C++ codes of FLAE are able to be downloaded on **https://github.com/zarathustr/FLAE_cpp**.

D. Extreme Case

Markley and Mortari carried out a test with an extreme case (scenario 2) in order to give the descriptions of the accuracy and robustness of various attitude determination algorithms [39]. This case is re-studied by Cheng in [4] where the sensor configurations are detailedly given. Such sensor combination makes the Newton iteration much harder to converge. In [4], transforming the original characteristic polynomial generates a more robust iteration approach showing that this improvement to the QUEST can well solve the problem. In the end of this paper, we are going to investigate the performance of the proposed FLAE in the presence of this case as well.

The maximum iteration counter is set to 50 while the accuracy indicator is defined as 1×10^{-15}. The conventional QUEST algorithm is compared with FLAE and the results are shown in Figure 13. It is apparent that the FLAE and QUEST all can not withstand the extreme case, preserving its drawback as there are too many maximum iterations. Then, the improved QUEST is applied to the extreme case whose details with FLAE are provided in Figure 14. We may see that the improved QUEST shows much better performance than FLAE in this case. The reason has been given in [4] describing that it is the precision, or the word-length, of float-point numbers that determines the final available accuracy of λ.

However, we may notice that the proposed FLAE's characteristic polynomial can also be transformed into the similar equation with that in Cheng's literature. Note that

$$\begin{aligned} &f(\lambda) = \lambda^4 + \tau_1\lambda^2 + \tau_2\lambda + \tau_3 \\ &\Rightarrow f(\lambda) = \left(\lambda^2 + \tfrac{\tau_1}{2}\right)^2 + \tau_2\lambda + \tau_3 - \tfrac{\tau_1^2}{4} \end{aligned} \tag{66}$$

whose derivative is

$$f'(\lambda) = 4\lambda\left(\lambda^2 + \frac{\tau_1}{2}\right) + \tau_2 \tag{67}$$

In this way the tested results are shown in Figure 15.

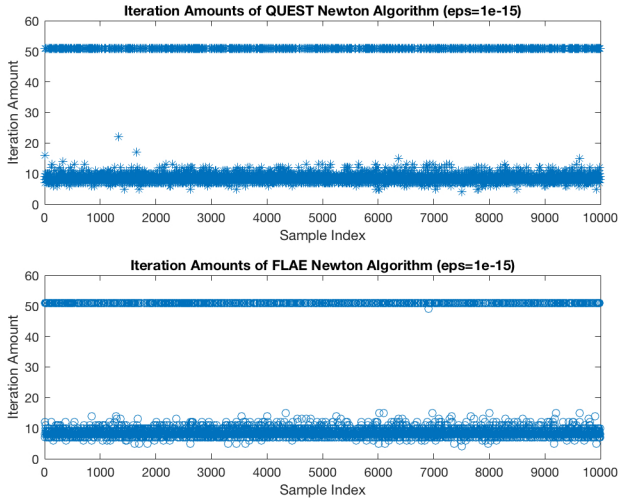


Fig. 13: Iteration amounts of conventional QUEST and FLAE when the extreme case is applied.

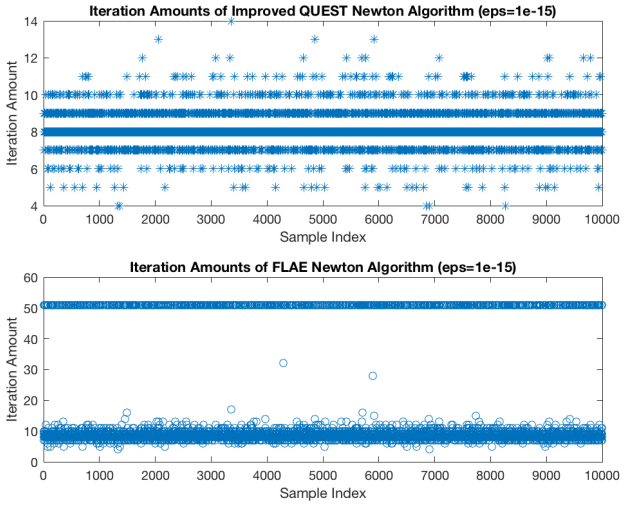


Fig. 14: Iteration amounts of improved QUEST and FLAE when the extreme case is applied.

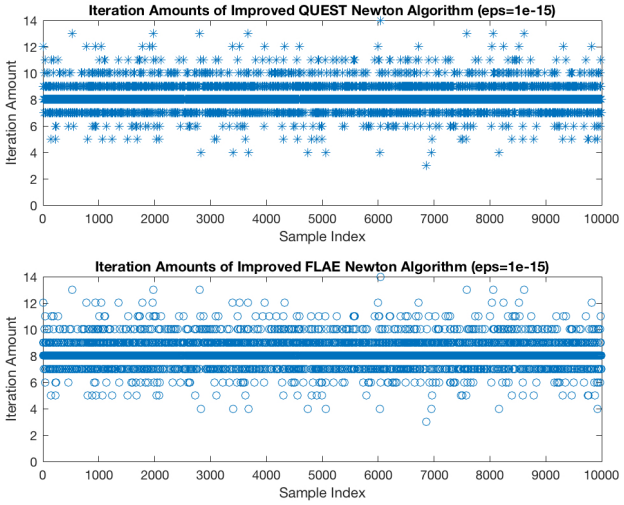


Fig. 15: Iteration amounts of improved QUEST and improved FLAE when the extreme case is applied.

The results show that the performance of FLAE under this case is significantly improved, with basically the same iteration amounts with that of the improved QUEST.

V. Conclusion

With pseudo-inverse matrices, we successfully establish the linear theory for solving Wahba's problem. Pseudo-inverse matrices are used for derivations rather than being used in the final FLAE algorithm. In order to enhance the robustness, the solving process is shifted to solving the eigenvalue of $\mathbf{W}$ that is very close to 1. A Newton iterative method and a symbolic solution method are designed to calculate the eigenvalues. The final FLAE algorithm uses the symbolic approach.

We may find out that the characteristic polynomial for the proposed FLAE, QUEST and ESOQ are very similar. However, as has been given in (49), the characteristic polynomial of the proposed FLAE owns the simplest form which does not contain adjoint matrices. This leads to the simplicity when calculating eigenvalues. Also, the proposed symbolic solutions to the characteristic polynomial would make the solving process even faster.

Numerical simulations are carried out. Comparisons with several representative methods are given. The simulation results show that the proposed FLAE exhibits the same performance for estimation errors and values of loss function but much lower time consumption. Based on the results, we systematically give the analysis of the advantage of FLAE on time consumption. A real experiment on an eCompass is also conducted which verifies the feasibility of the proposed algorithm for real applications. Finally, the study of the performance of FLAE in the presence of one extreme case is conducted revealing that the conventional FLAE faces dilemma in such occasion. An improved transformation is applied to FLAE providing us with almost the same results with improved QUEST which solves the problem properly.

We believe that the research findings will be of great benefit to solving the conventional Wahba's problem and it would accomplish related attitude determination tasks with much higher efficiency in future applications.

Appendix A
Quaternion and DCM

Spatial rotation can be described by direction cosine matrix (DCM). Commonly, a DCM can be defined with three Euler angles by [40]

$$\mathbf{C} = \begin{pmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ -s\psi c\gamma + c\psi s\theta s\gamma & c\psi c\gamma + s\psi s\theta s\gamma & c\theta s\gamma \\ s\psi s\theta + c\psi c\theta c\gamma & -c\psi s\gamma + s\psi s\theta c\gamma & c\theta c\gamma \end{pmatrix} \tag{68}$$

where c and s are the abbreviations of cos and sin functions respectively. θ, ψ, γ stand for the pitch, yaw and roll angles respectively. DCM is a universal tool for representing rotations but it has the disadvantages of singularities e.g the gimbal lock phenomenon[40]. To avoid such problems, quaternions can be introduced. A unit quaternion is defined over the Hamilton space $\mathbb{H}$, which can be written with three imaginary components and one real component [40].

$$\mathbf{q} = q_0 + \mathbf{i} \cdot q_1 + \mathbf{j} \cdot q_2 + \mathbf{k} \cdot q_3 \tag{69}$$

which satisfies

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{70}$$

Conversion from a quaternion to a DCM can be given by [40]

$$\mathbf{C} = \begin{pmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 1 - 2q_1^2 - 2q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 1 - 2q_1^2 - 2q_2^2 \end{pmatrix} \tag{71}$$

Appendix B
Proofs of $\mathbf{P}_D^\dagger$

We have proved the following equations in [26]:

$$\mathbf{P}_1^T = \mathbf{P}_1^\dagger, \mathbf{P}_2^T = \mathbf{P}_2^\dagger, \mathbf{P}_3^T = \mathbf{P}_3^\dagger \tag{72}$$

Let us simplify the multiplication below:

$$\begin{aligned} &\mathbf{P}_D\mathbf{P}_D^T = (D_x^r\mathbf{P}_1 + D_y^r\mathbf{P}_2 + D_z^r\mathbf{P}_3)(D_x^r\mathbf{P}_1^T + D_y^r\mathbf{P}_2^T + D_z^r\mathbf{P}_3^T) \\ &= (D_x^r)^2\mathbf{P}_1\mathbf{P}_1^T + (D_y^r)^2\mathbf{P}_2\mathbf{P}_2^T + (D_z^r)^2\mathbf{P}_3\mathbf{P}_3^T + \\ &D_x^rD_y^r(\mathbf{P}_1\mathbf{P}_2^T + \mathbf{P}_2\mathbf{P}_1^T) + D_x^rD_z^r(\mathbf{P}_1\mathbf{P}_3^T + \mathbf{P}_3\mathbf{P}_1^T) \\ &+D_y^rD_z^r(\mathbf{P}_2\mathbf{P}_3^T + \mathbf{P}_3\mathbf{P}_2^T) \end{aligned} \tag{73}$$

where

$$\mathbf{P}_1\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_2^T = \mathbf{P}_3\mathbf{P}_3^T = \mathbf{I}_{3\times3} \tag{74}$$

The rest ones equal to zero since we have

$$\mathbf{P}_1\mathbf{P}_2^T + \mathbf{P}_2\mathbf{P}_1^T = \mathbf{P}_1\mathbf{P}_3^T + \mathbf{P}_3\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_3^T + \mathbf{P}_3\mathbf{P}_2^T = \mathbf{0}_{3\times3} \tag{75}$$

Hence we have

$$\mathbf{P}_D\mathbf{P}_D^T = \mathbf{I}_{3\times3} \tag{76}$$

which yields

$$\mathbf{P}_D^\dagger = \mathbf{P}_D^T = D_x^r\mathbf{P}_1^T + D_y^r\mathbf{P}_2^T + D_z^r\mathbf{P}_3^T \tag{77}$$

Acknowledgment

This research was supported by National Natural Science Foundation of China under grant of No. 41604025 and No. 61450010. Some part of experiment equipments are supported by Yunkan Technology Inc., Chengdu, China. Dr. Yaguang Yang provided the source MATLAB code for his two algorithms. We genuinely thank them for their help.

References

[1] L. Chang, F. Qin, and A. Li, "A Novel Backtracking Scheme for Attitude Determination-Based Initial Alignment," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 384–390, 2014.

[2] S. Du and Y. Gao, "Inertial aided cycle slip detection and identification for integrated PPP GPS and INS," *Sensors (Switzerland)*, vol. 12, no. 11, pp. 14 344–14 362, 2012.

[3] G. Wahba, "A Least Squares Estimate of Satellite Attitude," p. 409, 1965.

[4] Y. Cheng and M. D. Shuster, "Improvement to the Implementation of the QUEST Algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 301–305, 2014.

[5] F. L. Markley and D. Mortari, "How to estimate attitude from vector observations," *Advances in the Astronautical Sciences*, vol. 103, no. PART III, pp. 1979–1996, 2000.

[6] J. L. Farrell and J. C. Stuelpnagel, "A Least Squares Estimate of Spacecraft Attitude," *SIAM Review*, vol. 8, no. 3, pp. 384–386, 1966.

[7] P. B. Davenport, "A vector approach to the algebra of rotations with applications," Tech. Rep. August, 1968.

[8] Y. Yang and Z. Zhou, "An analytic solution to Wahbas problem," *Aerospace Science and Technology*, vol. 30, no. 1, pp. 46–49, 2013.

[9] J. E. Keat, "Analysis of Least-Squares Attitude Determination Routine DOAOP," *Computer Sciences Corporation*, vol. 6034, 1977.

[10] Y. Yang, "Spacecraft attitude determination and control: Quaternion based method," *Annual Reviews in Control*, vol. 36, no. 2, pp. 198–219, 2012.

[11] M. D. Shuster and S. D. Oh, "Three-axis attitude determination from vector observations," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, pp. 70–77, 1981.

[12] Y. Cheng and M. D. Shuster, "Robustness and accuracy of the quest algorithm," *Advances in the Astronautical Sciences*, vol. 127, pp. 41–61, 2007.

[13] Y. Cheng and M. Shuster, "The speed of attitude estimation (aas 07-105)," *Advances in the Astronautical Sciences*, vol. 127, no. 1, p. 101, 2007.

[14] F. L. Markley, "Humble problems," *Advances in the Astronautical Sciences*, vol. 124 II, pp. 2205–2222, 2006.

[15] ——, "Attitude determination using vector observations - A fast optimal matrix algorithm," *The Journal of the Astronautical Sciences*, vol. 41, no. 2, pp. 261–280, 1993.

[16] ——, "Attitude determination using vector observations and the singular value decomposition," *Journal of the Astronautical Sciences*, vol. 36, no. 3, pp. 245–258, 1988.

[17] Mortari, Daniele, "ESOQ-2 single-point algorithm for fast optimal spacecraft attitude determination," *Advances in the Astronautical Sciences*, vol. 95 PART 2, no. August, pp. 817–825, 1997.

[18] D. Mortari, "ESOQ: A Closed-Form Solution to the Wahba Problem," *The Journal of the Astronautical Sciences*, vol. 45, no. 2, pp. 195–204, 1997.

[19] T. Ainscough, R. Zanetti, J. Christian, and P. D. Spanos, "Q-Method Extended Kalman Filter," *Journal of Guidance, Control, and Dynamics*, vol. Early Edit, no. 4, 2015.

[20] I. Y. Bar-Itzhack, "REQUEST - A recursive QUEST algorithm for sequential attitude determination," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 5, pp. 1034–1038, 1996.

[21] M. D. Shuster, "Filter QUEST or REQUEST," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 643–645, 2009.

[22] M. L. Psiaki and J. C. Hinks, "Numerical Solution of a Generalized Wahba Problem for a Spinning Spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 3, pp. 764–773, 2012.

[23] M. D. Shuster, "The generalized Wahba problem," *The Journal of the Astronautical Sciences*, vol. 54, no. 2, pp. 245–259, 2006.

[24] Y. Yang, "Attitude determination using Newton's method on Riemannian manifold," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 0, no. 0, pp. 1–6, 2015.

[25] D. Choukroun, I. Y. Bar-Itzhack, and Y. Oshman, "Novel quaternion Kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 174–190, 2006.

[26] 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, no. 18, pp. 6997–7007, 2016.

[27] C. A. Rohde, "Generalized inverses of partitioned matrices," *Journal of the Society for Industrial and Applied Mathematics*, vol. 13, no. 4, pp. 1033–1035, 1965.

[28] A. Ben-Israel and T. N. E. Greville, "Generalized inverses," *Cms Books in Mathematics*, vol. 34, no. 3, pp. 406–413, 2003.

[29] T. Delabie, J. D. Schutter, and B. Vandenbussche, "Robustness and Efficiency Improvements for Star Tracker Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 11, pp. 2108–2121, 2015.

[30] D. Gong, X. Shao, W. Li, and D. Duan, "Optimal linear attitude estimators via geometric analysis," *Journal of Zhejiang University SCIENCE A*, vol. 12, no. 11, pp. 873–882, 2011.

[31] N. Madinehi, "Rigid body attitude estimation: An overview and comparative study," 2013.

[32] Z. Zhou, Y. Li, J. Liu, and G. Li, "Equality constrained robust measurement fusion for adaptive kalman-filter-based heterogeneous multi-sensor navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2146–2157, 2013.

[33] W. Li and J. Wang, "Effective Adaptive Kalman Filter for MEMS-IMU/Magnetometers Integrated Attitude and Heading Reference Systems," *Journal of Navigation*, vol. 66, no. 01, pp. 99–113, 2012.

[34] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich, "Complementary Observer for Body Segments Motion Capturing by Inertial and Magnetic Sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 1, pp. 149–157, 2014.

[35] R. Mahony, T. Hamel, and J. M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[36] X. Yun and E. Bachmann, "Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking," *Robotics, IEEE Transactions on*, vol. 22, no. 6, pp. 1216–1227, 2006.

[37] R. G. Valenti, I. Dryanovski, and J. Xiao, "A Linear Kalman Filter for MARG Orientation Estimation Using the Algebraic Quaternion Algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 2, pp. 467–481, 2016.

[38] P. Marantos, Y. Koveos, and K. J. Kyriakopoulos, "UAV State Estimation Using Adaptive Complementary Filters," *IEEE Transactions on Control Systems Technology*, vol. 50, no. 7, pp. 1573–1582, 2015.

[39] F. L. Markley and D. Mortari, "Quaternion attitude estimation using vector observations." *Journal of the Astronautical Sciences*, vol. 48, no. 2, pp. 359–380, 2000.

[40] J. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008.