



Optimal, Recursive and Suboptimal Linear Solutions to Attitude Determination from Vector Observations

Jin Wu, Zebo Zhou, Jinling Wang, Hassen Fourati, Chang Lubin

► To cite this version:

Jin Wu, Zebo Zhou, Jinling Wang, Hassen Fourati, Chang Lubin. Optimal, Recursive and Suboptimal Linear Solutions to Attitude Determination from Vector Observations. 2017. hal-01525603v2

HAL Id: hal-01525603

<https://inria.hal.science/hal-01525603v2>

Preprint submitted on 20 Jul 2017 (v2), last revised 11 Dec 2017 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Optimal, Recursive and Suboptimal Linear Solutions to Attitude Determination from Vector Observations

Jin Wu, *Member, IEEE*, Zebo Zhou, Jinling Wang, Hassen Fourati and Lubin Chang, *Member, IEEE*

Abstract—This paper deals with the optimal attitude determination problem and its sub-optimal and time-varying recursive variants. The developed methods are named as the Optimal Linear Estimator of Quaternion (OLEQ), Suboptimal-OLEQ (SOLEQ) and Recursive-OLEQ (ROLEQ). The theory is established based on our previous contributions and the multi-vector matrix multiplications are decomposed with the eigenvalue factorization. Some analytical results are proved and given which provides the audience with a brand new viewpoint of the attitude determination and its evolution inside. With the derivations of two-vector case, the n -vector case is then naturally formed. Simulations are carried out showing the advantages of accuracy, robustness and time consumption of the proposed OLEQs, compared with representative methods. The algorithms are then implemented using C++ programming language on the designed hardware with 3-axis accelerometer, 3-axis magnetometer, giving the effectiveness validation of them in real-world applications.

Index Terms—Attitude Determination, Wahba's Problem, Quaternion Transformation, Vector Observations.

I. INTRODUCTION

ATTITUDE determination (or estimation) from vector observation pairs is a significant technology in aerospace engineering and related geodetic applications [1], [2], [3]. For instance, the inertial navigation, as an important role in modern military applications, has a high demand on attitude determination accuracy for initial alignment [4], [5], [6], [7].

Representative methods are mainly about the solutions to the famous Wahba's problem [8], which aims to obtain the optimal attitude determination results using weighted least squares. Among these algorithms, the Shuster's QUaternion ESTimator (QUEST, [9]), Markley's Singular Value Decomposition (SVD, [10]) and Mortari's (ESOQ, [11]) are the most frequently used ones, which are mostly inspired by Davenport's q-method [12], [13]. Our newly developed Fast Linear Attitude Estimator (FLAE, [14]) obtains a fastest Wahba's solution to

our existing knowledge. Some other interesting approaches are proposed as well, investigating the other internals of this problem e.g. Yang's analytical method, Riemannian-manifold algorithm and Forbes' Linear-Matrix-Inequality (LMI) solution. [15], [16], [17].

There are still some weight-less algorithms for multi-sensor attitude determination. They are usually used on applications like vision attitude determination where the a priori information of weights can hardly be accurately determined. For example, using the nonlinear special orthogonal group on SO3 [18], it is able to obtain the attitude quaternion from arbitrary pairs of vectors. Via optimization approaches like gradient-descent algorithm (GDA, [19]), Gauss-Newton algorithm (GNA, [20]), we may also compute stable solutions. Apart from these, a famous analytical method was proposed by Arun et al. where the Singular Value Decomposition (SVD) is employed to calculate the attitude matrix [21]. Invoking similar commitment, it is then introduced in computing both the attitude and translation vector in machine vision systems [22].

For Wahba's problem, it has been shown that most existing algorithms are based on the Davenport's q-method. For a long period, the attitude solving process is fixed on this framework which aims to find the largest eigenvalue of the Davenport matrix $\mathbf{K}$. Is it possible to seek another quite different attitude determination approach? The answer is positive and in this paper, three novel quaternion attitude determination algorithms from pairs of vector measurements are proposed in the sense of optimal, time-recursive and sub-optimal formulations. The main contributions are listed below:

- 1) The main theory is based on our previous research contributions and is extended to arbitrary pairs vector measurements linearly.
- 2) Three estimators i.e. the Optimal Linear Estimator of Quaternion (OLEQ), Recursive OLEQ (ROLEQ) and Sub-optimal OLEQ (SOLEQ) are derived. We also proposed accelerating techniques to make the algorithms faster.
- 3) Simulations and experiments are carried out, which verify the accuracy, flexibility, robustness and time consumption of various algorithms. Detailed comparisons with representative methods are shown to draw the superiority of the proposed OLEQs.

This paper is structured as follows:

- 1) Section II contains the problem formulation and presents that quaternion estimation from a single sensor observation. The two-vector attitude determination theory along

This research is supported by National Natural Science Foundation of China under the grant number of 41604025.

The MATLAB source codes of the proposed algorithms can be acquired from <https://github.com/zarathustr/OLEQs>.

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

Correspondent: Z. Zhou is with School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, China. e-mail: klinsmann.zhou@gmail.com

J. Wang is with School of Civil and Environmental Engineering, The University of New South Wales, Australia. e-mail: jinling.wang@unsw.edu.au

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

L. Chang is with the Naval University of Engineering, Wuhan, China. e-mail: lubin.chang.1987@ieee.org; changlubin@163.com

with the n -vector one are given accordingly.

- 2) Section III involves the numerical examples and simulations where comparisons between the proposed SOLEQ and other representative methods are given.
- 3) Section IV consists of concluding remarks and discussion about the future work.

II. PROBLEM FORMULATION

The conventional Wahba's problem aims to find the optimal attitude matrix from vector observation pairs in the sense of least squares, such that

$$L(\mathbf{C}) = \sum_{i=1}^n a_i \|\mathbf{D}_i^b - \mathbf{C}\mathbf{D}_i^r\|^2, n = 2, 3, \dots \quad (1)$$

in which $\mathbf{C}$ denotes the optimal direction cosine matrix (DCM); $\mathbf{D}_i^b = (D_{x,i}^b, D_{y,i}^b, D_{z,i}^b)^\top$ and $\mathbf{D}_i^r = (D_{x,i}^r, D_{y,i}^r, D_{z,i}^r)^\top$ are the i -th pair of normalized vector observations from the body frame b and the reference frame r respectively; a_i is the weight of the i -th sensor output, which is given by the standard deviations of the input vectors from σ_1 to σ_n :

$$a_i = \frac{\sigma_{tot}^2}{\sigma_i^2} \quad (2)$$

$$\sigma_{tot}^2 = \frac{1}{\sum_{i=1}^n \frac{1}{\sigma_i^2}}$$

Wahba's problem is solved via many representative methods [23]. Many of these algorithms solve the problem via eigenvalue decompositions analytically or numerically [15], [16]. When there are only one pair of vector observations, the Wahba's solutions fail to obtain the optimal quaternion since there will be ambiguous quaternions corresponding to the maximum eigenvalue 1 [24], [25]. In our previous contribution [26], the continuous stable quaternion solution to an accelerometer-based attitude determination system is derived.

A. Quaternion from A Single Sensor Observation

Considering an attitude determination model from a pair of vector observations

$$\mathbf{D}^b = \mathbf{C}\mathbf{D}^r \quad (3)$$

, this section deals with the attitude determination from a single pair of sensor observations. Note that the DCM is decomposed with quaternions $\mathbf{q} = (q_0, q_1, q_2, q_3)^\top$ [26] via:

$$\mathbf{C} = (\mathbf{P}_1\mathbf{q}, \mathbf{P}_2\mathbf{q}, \mathbf{P}_3\mathbf{q}) \quad (4)$$

in which

$$\mathbf{P}_1 = \begin{pmatrix} q_0 & q_1 & -q_2 & -q_3 \\ -q_3 & q_2 & q_1 & -q_0 \\ q_2 & q_3 & q_0 & q_1 \\ q_3 & q_2 & q_1 & q_0 \end{pmatrix} \quad (5)$$

$$\mathbf{P}_2 = \begin{pmatrix} q_0 & -q_1 & q_2 & -q_3 \\ -q_1 & -q_0 & q_3 & q_2 \\ -q_2 & q_3 & -q_0 & q_1 \\ q_1 & q_0 & q_3 & q_2 \end{pmatrix}$$

$$\mathbf{P}_3 = \begin{pmatrix} q_0 & -q_1 & -q_2 & q_3 \\ q_1 & q_0 & -q_2 & -q_3 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{pmatrix}$$

In this section, the theory is extended to arbitrary sensor with exactly the similar approach in [26]. Inserting (4) into (3) gives

$$\mathbf{D}^b = (D_x^r\mathbf{P}_1 + D_y^r\mathbf{P}_2 + D_z^r\mathbf{P}_3)\mathbf{q} \quad (6)$$

It has been proved in [26] that

$$\mathbf{P}_1^\top = \mathbf{P}_1^\dagger, \mathbf{P}_2^\top = \mathbf{P}_2^\dagger, \mathbf{P}_3^\top = \mathbf{P}_3^\dagger \quad (7)$$

where $\dagger$ stands for the Moore-Penrose pseudo-inverse. In fact, another property can be shown in the following theorem

Theorem 1. $\mathbf{P}_1\mathbf{P}_2^\top + \mathbf{P}_2\mathbf{P}_1^\top = \mathbf{P}_1\mathbf{P}_3^\top + \mathbf{P}_3\mathbf{P}_1^\top = \mathbf{P}_2\mathbf{P}_3^\top + \mathbf{P}_3\mathbf{P}_2^\top = \mathbf{0}_{3 \times 3}$.

Proof. This can be proved via brute-force calculation:

$$\mathbf{P}_1\mathbf{P}_2^\top = \begin{pmatrix} 0 & -q_0^1 + q_1^1 + q_2^2 - q_3^2 & 2q_0q_1 + 2q_2q_3 \\ q_0^1 - q_1^1 - q_2^2 + q_3^2 & 0 & 2q_0q_2 - 2q_1q_3 \\ -2q_0q_1 - 2q_2q_3 & -2q_0q_2 + 2q_1q_3 & 0 \end{pmatrix}$$

$$\mathbf{P}_2\mathbf{P}_1^\top = \begin{pmatrix} 0 & q_0^1 - q_1^1 - q_2^2 + q_3^2 & -2q_0q_1 - 2q_2q_3 \\ -q_0^1 + q_1^1 + q_2^2 - q_3^2 & 0 & -2q_0q_2 + 2q_1q_3 \\ 2q_0q_1 + 2q_2q_3 & 2q_0q_2 - 2q_1q_3 & 0 \end{pmatrix} \quad (8a)$$

$$\mathbf{P}_1\mathbf{P}_3^\top = \begin{pmatrix} 0 & 2q_0q_1 - 2q_2q_3 & q_0^1 - q_1^1 + q_2^2 - q_3^2 \\ -2q_0q_1 + 2q_2q_3 & 0 & -2q_1q_2 - 2q_0q_3 \\ -q_0^1 + q_1^1 - q_2^2 + q_3^2 & 2q_1q_2 + 2q_0q_3 & 0 \end{pmatrix}$$

$$\mathbf{P}_3\mathbf{P}_1^\top = \begin{pmatrix} 0 & -2q_0q_1 + 2q_2q_3 & -q_0^1 + q_1^1 - q_2^2 + q_3^2 \\ 2q_0q_1 - 2q_2q_3 & 0 & 2q_1q_2 + 2q_0q_3 \\ q_0^1 - q_1^1 + q_2^2 - q_3^2 & -2q_1q_2 - 2q_0q_3 & 0 \end{pmatrix} \quad (8b)$$

$$\mathbf{P}_2\mathbf{P}_3^\top = \begin{pmatrix} 0 & 2q_0q_2 + 2q_1q_3 & -2q_1q_2 + 2q_0q_3 \\ -2q_0q_2 - 2q_1q_3 & 0 & q_0^1 + q_1^1 - q_2^2 - q_3^2 \\ 2q_1q_2 - 2q_0q_3 & -q_0^1 - q_1^1 + q_2^2 + q_3^2 & 0 \end{pmatrix}$$

$$\mathbf{P}_3\mathbf{P}_2^\top = \begin{pmatrix} 0 & -2q_0q_2 - 2q_1q_3 & 2q_1q_2 - 2q_0q_3 \\ 2q_0q_2 + 2q_1q_3 & 0 & -q_0^1 - q_1^1 + q_2^2 + q_3^2 \\ -2q_1q_2 + 2q_0q_3 & q_0^1 + q_1^1 - q_2^2 - q_3^2 & 0 \end{pmatrix} \quad (8c)$$

Lemma 1. Let $\mathbf{K}(\mathbf{q}) = (D_x^r\mathbf{P}_1 + D_y^r\mathbf{P}_2 + D_z^r\mathbf{P}_3)$, $\mathbf{K}^\top(\mathbf{q}) = \mathbf{K}^\dagger(\mathbf{q})$.

Proof. We have

$$\mathbf{K}(\mathbf{q})\mathbf{K}^\top(\mathbf{q}) = (D_x^r\mathbf{P}_1 + D_y^r\mathbf{P}_2 + D_z^r\mathbf{P}_3)(D_x^r\mathbf{P}_1 + D_y^r\mathbf{P}_2 + D_z^r\mathbf{P}_3)^\top$$

$$= (D_x^r)^2\mathbf{P}_1\mathbf{P}_1^\top + (D_y^r)^2\mathbf{P}_2\mathbf{P}_2^\top + (D_z^r)^2\mathbf{P}_3\mathbf{P}_3^\top$$

$$+ D_x^rD_y^r(\mathbf{P}_1\mathbf{P}_2^\top + \mathbf{P}_2\mathbf{P}_1^\top)$$

$$+ D_x^rD_z^r(\mathbf{P}_1\mathbf{P}_3^\top + \mathbf{P}_3\mathbf{P}_1^\top)$$

$$+ D_y^rD_z^r(\mathbf{P}_2\mathbf{P}_3^\top + \mathbf{P}_3\mathbf{P}_2^\top)$$

$$= [(D_x^r)^2 + (D_y^r)^2 + (D_z^r)^2] \mathbf{I}_{3 \times 3} = \mathbf{I}_{3 \times 3}$$

which gives Lemma 1. ■

Hence (6) can be transformed into

$$\mathbf{D}^b = \mathbf{K}(\mathbf{q})\mathbf{q} \Rightarrow \mathbf{K}^\dagger(\mathbf{q})\mathbf{D}^b = \mathbf{q} \Rightarrow \mathbf{K}^\top(\mathbf{q})\mathbf{D}^b = \mathbf{q} \quad (10)$$

Since $\mathbf{K}(\mathbf{q})$ is in the form of quaternion, $\mathbf{K}^\top(\mathbf{q})\mathbf{D}^b$ can be expanded by

$$\mathbf{K}^\top(\mathbf{q})\mathbf{D}^b = D_x^r\mathbf{P}_1^\top\mathbf{D}^b + D_y^r\mathbf{P}_2^\top\mathbf{D}^b + D_z^r\mathbf{P}_3^\top\mathbf{D}^b \quad (11)$$

Also it can be obtained that

$$\begin{aligned} \mathbf{P}_1^\top \mathbf{D}^b &= \begin{pmatrix} D_x^b q_0 + D_z^b q_2 - D_y^b q_3 \\ D_x^b q_1 + D_z^b q_2 + D_y^b q_3 \\ D_z^b q_0 + D_y^b q_1 - D_x^b q_2 \\ -D_y^b q_0 + D_z^b q_1 - D_x^b q_3 \end{pmatrix} \\ &= \begin{pmatrix} D_x^b & 0 & D_z^b & -D_y^b \\ 0 & D_x^b & D_z^b & D_y^b \\ D_z^b & D_y^b & -D_x^b & 0 \\ -D_y^b & D_z^b & 0 & -D_x^b \end{pmatrix} \mathbf{q} = \mathbf{M}_1 \mathbf{q} \end{aligned} \quad (12a)$$

$$\begin{aligned} \mathbf{P}_2^\top \mathbf{D}^b &= \begin{pmatrix} D_y^b q_0 - D_z^b q_1 + D_x^b q_3 \\ -D_z^b q_0 - D_y^b q_1 + D_x^b q_2 \\ D_x^b q_1 + D_y^b q_2 + D_z^b q_3 \\ D_x^b q_0 + D_z^b q_2 - D_y^b q_3 \end{pmatrix} \\ &= \begin{pmatrix} D_y^b & -D_z^b & 0 & D_x^b \\ -D_z^b & -D_y^b & D_x^b & 0 \\ 0 & D_x^b & D_y^b & D_z^b \\ D_x^b & 0 & D_z^b & -D_y^b \end{pmatrix} \mathbf{q} = \mathbf{M}_2 \mathbf{q} \end{aligned} \quad (12b)$$

$$\begin{aligned} \mathbf{P}_3^\top \mathbf{D}^b &= \begin{pmatrix} D_z^b q_0 + D_y^b q_1 - D_x^b q_2 \\ D_y^b q_0 - D_z^b q_1 + D_x^b q_3 \\ -D_x^b q_0 - D_z^b q_2 + D_y^b q_3 \\ D_x^b q_1 + D_y^b q_2 + D_z^b q_3 \end{pmatrix} \\ &= \begin{pmatrix} D_z^b & D_y^b & -D_x^b & 0 \\ D_y^b & -D_z^b & 0 & D_x^b \\ -D_x^b & 0 & -D_z^b & D_y^b \\ 0 & D_x^b & D_y^b & D_z^b \end{pmatrix} \mathbf{q} = \mathbf{M}_3 \mathbf{q} \end{aligned} \quad (12c)$$

Then we have

$$\mathbf{K}^\top(\mathbf{q})\mathbf{D}^b = D_x^r \mathbf{M}_1 \mathbf{q} + D_y^r \mathbf{M}_2 \mathbf{q} + D_z^r \mathbf{M}_3 \mathbf{q} = \mathbf{W} \mathbf{q} \quad (13)$$

where $\mathbf{W}$ is given by

$$\mathbf{W} = D_x^r \mathbf{M}_1 + D_y^r \mathbf{M}_2 + D_z^r \mathbf{M}_3 \quad (14)$$

Therefore the attitude determination problem is shifted to

$$\mathbf{W} \mathbf{q} = \mathbf{q} \quad (15)$$

Theorem 2. For any normalized vector observation pair $\{\mathbf{D}^r, \mathbf{D}^b\}$, $\mathbf{W}^2 = \mathbf{I}$, where $\mathbf{I}$ is the four-order identity matrix.

Proof. The characteristic polynomial of $\mathbf{W}$ is given by

$$p(\lambda) = (\lambda - 1)^2(\lambda + 1)^2 \quad (16)$$

Then with Cayley-Hamilton Theorem, we substitute $\mathbf{W}$ for its eigenvalues and obtain

$$(\mathbf{W}^2 - \mathbf{I})^2 = \mathbf{0} \quad (17)$$

as $\mathbf{W}^2 - \mathbf{I}$ is real symmetric, it should be $\mathbf{0}$, which finishes the proof. $\blacksquare$

In [26], we showed that (15) can be seen as an iterative dynamical system

$$\mathbf{q}(n) = \mathbf{W} \mathbf{q}(n-1) \quad (18)$$

where $\mathbf{q}(n)$ denotes the quaternion for the n th iteration. Also, as has been proved, the discrete system can be converted to

a corresponding continuous system. The stable solution to the continuous system is calculated as

$$\mathbf{q} = \frac{\mathbf{W} + \mathbf{I}}{2} \mathbf{q}_{rand} \quad (19)$$

if $\mathbf{W}^2 = \mathbf{I}$. Where $\mathbf{q}_{rand}$ denotes an randomly-chosen unit quaternion. This provides us with a new approach to obtaining the measurement quaternion from a single strapdown sensor.

III. OPTIMAL LINEAR ESTIMATOR OF QUATERNION

With (3), it is able to list all the single rotation equations together as follows:

$$\begin{cases} \mathbf{D}_1^b = \mathbf{C} \mathbf{D}_1^r \\ \mathbf{D}_2^b = \mathbf{C} \mathbf{D}_2^r \\ \vdots \\ \mathbf{D}_n^b = \mathbf{C} \mathbf{D}_n^r \end{cases} \Rightarrow \begin{cases} \sqrt{a_1} \mathbf{D}_1^b = \sqrt{a_1} \mathbf{C} \mathbf{D}_1^r \\ \sqrt{a_2} \mathbf{D}_2^b = \sqrt{a_2} \mathbf{C} \mathbf{D}_2^r \\ \vdots \\ \sqrt{a_n} \mathbf{D}_n^b = \sqrt{a_n} \mathbf{C} \mathbf{D}_n^r \end{cases} \quad (20)$$

which can in turn be converted to

$$\begin{cases} \sqrt{a_1} \mathbf{q} = \sqrt{a_1} \mathbf{W}_1 \mathbf{q} \\ \sqrt{a_2} \mathbf{q} = \sqrt{a_2} \mathbf{W}_2 \mathbf{q} \\ \vdots \\ \sqrt{a_n} \mathbf{q} = \sqrt{a_n} \mathbf{W}_n \mathbf{q} \end{cases} \quad (21)$$

This system is rewritten as

$$\begin{pmatrix} \sqrt{a_1} \mathbf{I} \\ \sqrt{a_2} \mathbf{I} \\ \vdots \\ \sqrt{a_n} \mathbf{I} \end{pmatrix} \mathbf{q} = \begin{pmatrix} \sqrt{a_1} \mathbf{W}_1 \\ \sqrt{a_2} \mathbf{W}_2 \\ \vdots \\ \sqrt{a_n} \mathbf{W}_n \end{pmatrix} \mathbf{q} \quad (22)$$

Notice that

$$(\sqrt{a_1} \mathbf{I}, \sqrt{a_2} \mathbf{I}, \dots, \sqrt{a_n} \mathbf{I}) \begin{pmatrix} \sqrt{a_1} \mathbf{I} \\ \sqrt{a_2} \mathbf{I} \\ \vdots \\ \sqrt{a_n} \mathbf{I} \end{pmatrix} = \left(\sum_{i=1}^n a_i \right) \mathbf{I} = \mathbf{I} \quad (23)$$

Hence the pre-multiplication leads to

$$\mathbf{q} = \left(\sum_{i=1}^n a_i \mathbf{W}_i \right) \mathbf{q} \quad (24)$$

It should be then noted that (22) is free of the existence of noises. In fact, the optimal attitude determination i.e. the Wahba's problem, has been proved as a total least-square problem in which both the reference and observation models are corrupted by the stochastic items. In this case, the maximum eigenvalue of $\sum_{i=1}^n a_i \mathbf{W}_i$ in engineering practice is very close to the noise-free theoretical result of 1. Corresponding with (18), based on the Brouwer's fixed-point theorem, it is able to iteratively obtain the normalized optimal quaternion by rotating a randomly given initial quaternion over and over again till infinity. This is something similar with the power

method of the numerical eigenvalue factorization but can be accelerated with geometric-series-like iterations as follows

$$\begin{aligned}\mathcal{R} &= \left(\sum_{i=1}^n a_i \mathbf{W}_i \right) \\ \mathcal{R}^2 &= \mathcal{R} \cdot \mathcal{R} \\ &\vdots \\ \mathcal{R}^{2^j} &= \mathcal{R}^{2^{j-1}} \cdot \mathcal{R}^{2^{j-1}}\end{aligned}\quad (25)$$

where $\mathcal{R}$ denotes the rotation operator over the Hamilton space $\mathbb{H}$. In fact, the above iterations can hardly be achieved when the maximum eigenvalue approaches 1. The reason is that at this time the power $\mathcal{R}^2$ approaches $\mathbf{I}$ as well. A more robust way is shown to solve this problem. Considering the both sides of (22), we may find out that the right side is in fact the mixture of solutions to single vector observation pairs. As mentioned in (19), a stable, continuous solution to each single equation can be done by pre-multiplying $\frac{1}{2}(\mathbf{W}_i + \mathbf{I})$. Substituting $\frac{1}{2}(\mathbf{W}_i + \mathbf{I})$ for $\mathbf{W}_i$, i.e.

$$\begin{pmatrix} \sqrt{a_1} \mathbf{I} \\ \sqrt{a_2} \mathbf{I} \\ \vdots \\ \sqrt{a_n} \mathbf{I} \end{pmatrix} \mathbf{q} = \begin{bmatrix} \frac{1}{2} \sqrt{a_1} (\mathbf{W}_1 + \mathbf{I}) \\ \frac{1}{2} \sqrt{a_2} (\mathbf{W}_2 + \mathbf{I}) \\ \vdots \\ \frac{1}{2} \sqrt{a_n} (\mathbf{W}_n + \mathbf{I}) \end{bmatrix} \mathbf{q} \quad (26)$$

, the quaternion evolves from the mixture of single optimal solution. Here, the rotation operator is revised as

$$\mathcal{R} = \sum_{i=1}^n \frac{1}{2} a_i (\mathbf{W}_i + \mathbf{I}) = \frac{1}{2} \left(\mathbf{I} + \sum_{i=1}^n a_i \mathbf{W}_i \right) \quad (27)$$

This equals to the least-square of the set of pre-computed single rotated quaternion, which is definitely faster and more robust than rotation from a randomly given initial quaternion.

The covariance of the calculated optimal quaternion is equal to that of QUEST under the condition of least-square optimality:

$$\begin{aligned}\Sigma_{\mathbf{Q}, \text{OLEQ}} &= \frac{1}{4} \sigma_{tot}^2 \left(\mathbf{I} - \sum_{i=1}^n \mathbf{D}^b \{ \mathbf{D}^b \}^\top \right) \\ \Sigma_{\mathbf{q}, \text{OLEQ}} &= [\mathbf{q}_{opt}] \begin{pmatrix} 0 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & \Sigma_{\mathbf{Q}, \text{OLEQ}} \end{pmatrix} [\mathbf{q}_{opt}]^\top\end{aligned}\quad (28)$$

where $\mathbf{Q} = (q_1, q_2, q_3)^\top$ is the vector part of the quaternion. $[\mathbf{q}]$ defines the following matrix

$$[\mathbf{q}] = \begin{pmatrix} q_3 & -q_2 & q_1 & q_0 \\ q_2 & q_3 & -q_0 & q_1 \\ -q_1 & q_0 & q_3 & q_2 \\ -q_0 & -q_1 & -q_2 & q_3 \end{pmatrix} \quad (29)$$

A. Variant One: Recursive-OLEQ

We have seen from the above formulations that for each epoch, the vector observations are batchedly processed thoroughly. When used in aerospace electronic systems, the measured vector observation pairs in neighboring time epochs are usually continuous since they are always been smoothed by the sum filters and low-pass filters (LPF). Therefore, with this consideration, the attitude quaternion can be propagated

from the last estimated one using the rotation operator described before. In this way the quaternions are recursively computed with much less computations and the accuracy is maintained. A more convenient clue is that for high reliable attitude determination systems, high-precision rate gyroscopes are employed usually. This provides us with a second-stage accelerating scheme, inspired by the conventional recursive algorithms like filter QUEST, REQUEST and etc. [27], that we may first rotate the estimated quaternion in last time epoch with zero-order angular transition matrix by

$$\mathbf{q}_k = \Phi_{k,k-1} \mathbf{q}_{k-1} \quad (30)$$

where

$$\Phi_{k,k-1} = \mathbf{I} + \frac{\Delta t}{2} [\Omega \times] \quad (31)$$

in which Δt is the sampling time and $[\Omega \times]$ composed by the angular rate from the gyroscope, which is detailed in many classical literatures. After this, even a single rotation by $\mathcal{R}$ would be very accurate then. The one-step covariance matrix of the obtained quaternion is calculated by

$$\Sigma_{\mathbf{q}_k, \text{ROLEQ}} = \mathcal{R} \Phi_{k,k-1} \Sigma_{\mathbf{q}_{k-1}, \text{ROLEQ}} \Phi_{k,k-1}^\top \mathcal{R}^\top \quad (32)$$

B. Variant Two: SOLEQ

1) *Two-Vector Case*: When there are two pairs of vector observations, regardless of the weights of respective sensors, it is natural that the attitude quaternion may be computed via

$$\mathbf{q} = \frac{1}{4} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I}) \mathbf{q}_{rand} \quad (33)$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are for the 1st and 2nd sensors. (19) is reasonable since we have

$$\left[\frac{1}{2} (\mathbf{W} + \mathbf{I}) \right]^2 = \frac{1}{4} (\mathbf{W}^2 + 2\mathbf{W} + \mathbf{I}) = \frac{1}{4} (2\mathbf{W} + 2\mathbf{I}) = \frac{1}{2} (\mathbf{W} + \mathbf{I}) \quad (34)$$

However, for the two-vector case, one can write

$$\begin{aligned}\left[\frac{1}{4} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I}) \right]^2 &= \left[\frac{1}{4} (\mathbf{W}_1 \mathbf{W}_2 + \mathbf{W}_1 + \mathbf{W}_2 + \mathbf{I}) \right]^2 \\ &\neq \frac{1}{4} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I})\end{aligned}\quad (35)$$

This reflects that the accurate quaternion may be recursively computed with

$$\begin{cases} \mathbf{q}_k = \frac{1}{4} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I}) \mathbf{q}_{k-1} \\ \mathbf{q}_k = \frac{\mathbf{q}_k}{\|\mathbf{q}_k\|} \end{cases} \quad (36)$$

which exits when the Euler distance $\|\mathbf{q}_k - \mathbf{q}_{k-1}\|$ is less than one predetermined threshold η . Note that the above initialization procedure is equivalent to the following process

$$\begin{cases} \mathbf{q}^j = \left[\frac{1}{4} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I}) \right]^j \mathbf{q}_{rand} \\ \mathbf{q}^j = \frac{\mathbf{q}^j}{\|\mathbf{q}^j\|} \end{cases} \quad (37)$$

where j is chosen to make sure $\|\mathbf{q}^j - \mathbf{q}^{j-1}\| < \eta$. Let us define some notations i.e. transformation operators

$$\mathcal{A} = \frac{1}{4} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I}) \quad (38a)$$

$$\mathcal{B} = \mathcal{A}^\top = \frac{1}{4} (\mathbf{W}_2 + \mathbf{I}) (\mathbf{W}_1 + \mathbf{I}) \quad (38b)$$

Theorem 3. For the two-vector attitude determination case, the steady-state evolution in (37) is not affected by transformation operators' order, such that

$$\begin{cases} \mathbf{q}^j = \mathcal{A}^j \mathbf{q}_{rand} \\ \mathbf{q}^j = \frac{\mathbf{q}^j}{\|\mathbf{q}^j\|} \end{cases} \Leftrightarrow \begin{cases} \mathbf{q}^j = \mathcal{B}^j \mathbf{q}_{rand} \\ \mathbf{q}^j = \frac{\mathbf{q}^j}{\|\mathbf{q}^j\|} \end{cases}, j \rightarrow +\infty \quad (39)$$

Proof. The integrated transformation can be computed by

$$\begin{aligned} \mathcal{A}\mathcal{B} &= \frac{1}{16} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I})^2 (\mathbf{W}_1 + \mathbf{I}) \\ &= \frac{1}{8} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I}) (\mathbf{W}_1 + \mathbf{I}) \\ &= \frac{1}{2} \mathcal{A} (\mathbf{W}_1 + \mathbf{I}) = \frac{1}{2} (\mathbf{W}_1 + \mathbf{I}) \mathcal{B} \end{aligned} \quad (40)$$

and accordingly

$$\mathcal{B}\mathcal{A} = \frac{1}{2} \mathcal{B} (\mathbf{W}_2 + \mathbf{I}) = \frac{1}{2} (\mathbf{W}_2 + \mathbf{I}) \mathcal{A} \quad (41)$$

These generate

$$\begin{aligned} \mathcal{A}\mathcal{B}\mathcal{A} &= \frac{1}{2} (\mathbf{W}_1 + \mathbf{I}) \mathcal{B}\mathcal{A} \\ &= \frac{1}{4} (\mathbf{W}_1 + \mathbf{I}) (\mathbf{W}_2 + \mathbf{I}) \mathcal{A} = \mathcal{A}^2 \end{aligned} \quad (42)$$

and

$$\mathcal{A}\mathcal{B}\mathcal{A}\mathcal{B} = \mathcal{A}^2 \mathcal{B} = \mathcal{A} (\mathcal{A}\mathcal{B}) = \frac{1}{2} \mathcal{A}^2 (\mathbf{W}_1 + \mathbf{I}) \quad (43)$$

Finally, we have

$$(\mathcal{A}\mathcal{B})^j = \frac{1}{2} \mathcal{A}^j (\mathbf{W}_1 + \mathbf{I}) = \frac{1}{2} (\mathbf{W}_1 + \mathbf{I}) \mathcal{B}^j \quad (44)$$

This proves that the mixed steady-state transformation $(\mathcal{A}\mathcal{B})^j$ can be achieved by independent transformations from $\mathcal{A}$ or $\mathcal{B}$, which finishes the proof. ■

Following this theorem, the confronted problem is to compute the power $\mathcal{A}^j$. In fact, $\mathcal{A}$ is formed by $\frac{1}{2} (\mathbf{W}_1 + \mathbf{I})$ and $\frac{1}{2} (\mathbf{W}_2 + \mathbf{I})$. Their respective eigenvalue decomposition can be given by

$$\begin{cases} \frac{1}{2} (\mathbf{W}_1 + \mathbf{I}) = \mathbf{V}_1 \mathbf{S}_1 \mathbf{V}_1^{-1} = \mathbf{V}_1 \mathbf{S}_1 \mathbf{V}_1^\top \\ \frac{1}{2} (\mathbf{W}_2 + \mathbf{I}) = \mathbf{V}_2 \mathbf{S}_2 \mathbf{V}_2^{-1} = \mathbf{V}_2 \mathbf{S}_2 \mathbf{V}_2^\top \end{cases} \quad (45)$$

where $\mathbf{V}$ and $\mathbf{D}$ are constituted by eigenvectors and eigenvalues respectively as $(\mathbf{W}_i + \mathbf{I})$ is real symmetric [28]. Since $\mathbf{W}_1$ and $\mathbf{W}_2$ are in the same form, the eigenvalue matrices are equal to each other, i.e.

$$\mathbf{S}_1 = \mathbf{S}_2 = \mathbf{S} \quad (46)$$

Then $\mathcal{A}$ can be rewritten as

$$\mathcal{A} = \mathbf{V}_1 \mathbf{S} \mathbf{V}_1^\top \mathbf{V}_2 \mathbf{S} \mathbf{V}_2^\top \quad (47)$$

Identically, we have

$$\mathcal{B} = \mathbf{V}_2 \mathbf{S} \mathbf{V}_2^\top \mathbf{V}_1 \mathbf{S} \mathbf{V}_1^\top \quad (48)$$

Combining (47) and (48), it is obtained that

$$\mathcal{A}\mathcal{B} = \mathbf{V}_1 \mathbf{S} \mathbf{V}_1^\top \mathbf{V}_2 \mathbf{S} \mathbf{V}_2^\top \mathbf{V}_2 \mathbf{S} \mathbf{V}_2^\top \mathbf{V}_1 \mathbf{S} \mathbf{V}_1^\top \quad (49)$$

Note that

$$\mathbf{V}_1 \mathbf{V}_1^\top = \mathbf{V}_1^\top \mathbf{V}_1 = \mathbf{V}_2 \mathbf{V}_2^\top = \mathbf{V}_2^\top \mathbf{V}_2 = \mathbf{I} \quad (50)$$

(49) is simplified as

$$\mathcal{A}\mathcal{B} = \mathbf{V}_1 \mathbf{S} \mathbf{V}_1^\top \mathbf{V}_2 \mathbf{S} \mathbf{V}_2^\top \mathbf{V}_1 \mathbf{S} \mathbf{V}_1^\top \quad (51)$$

Here we define

$$\mathbf{U} = \mathbf{S} \mathbf{V}_1^\top \mathbf{V}_2 \mathbf{S} \mathbf{V}_2^\top \mathbf{V}_1 \mathbf{S} \quad (52)$$

Actually it is decomposed by

$$\begin{aligned} \mathbf{U} &= \mathbf{H} \mathbf{H}^\top \\ \mathbf{H} &= \mathbf{S} \mathbf{V}_1^\top \mathbf{V}_2 \mathbf{S} \end{aligned} \quad (53)$$

An interesting fact is that the eigenvalue matrix $\mathbf{S}$ can be analytically calculated and is given by

$$\mathbf{S} = \text{diag}(0, 0, 1, 1) \quad (54)$$

where $\text{diag}(\cdot)$ represents the diagonal matrix. This further yields $\mathbf{H}$ to be a matrix with the form of

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & h_1 & h_2 \\ 0 & 0 & h_3 & h_4 \end{pmatrix} \quad (55)$$

Then $\mathbf{U}$ is computed by

$$\begin{aligned} \mathbf{U} &= \mathbf{H} \mathbf{H}^\top \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & u_1 & u_2 \\ 0 & 0 & u_3 & u_4 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & h_1^2 + h_2^2 & h_1 h_3 + h_2 h_4 \\ 0 & 0 & h_1 h_3 + h_2 h_4 & h_3^2 + h_4^2 \end{pmatrix} \end{aligned} \quad (56)$$

where $u_2 = u_3$. Using this, we have

$$(\mathcal{A}\mathcal{B})^j = \mathbf{V}_1 \mathbf{U} \mathbf{V}_1^\top \mathbf{V}_1 \mathbf{U} \mathbf{V}_1^\top \cdots \mathbf{V}_1 \mathbf{U} \mathbf{V}_1^\top = \mathbf{V}_1 \mathbf{U}^j \mathbf{V}_1^\top \quad (57)$$

$\mathbf{U}$ can be decomposed with eigenvalue decomposition as well, such that

$$\mathbf{U} = \mathbf{V}_U \mathbf{S}_U \mathbf{V}_U^\top \quad (58)$$

which can be analytically given by

$$\begin{aligned} \mathbf{S}_U &= \text{diag} \left(0, 0, -\frac{1}{2} \sqrt{\Delta} + \frac{u_1 + u_4}{2}, +\frac{1}{2} \sqrt{\Delta} + \frac{u_1 + u_4}{2} \right) \\ \mathbf{V}_U &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-u_1 + u_4 + \sqrt{\Delta}}{2u_2} & \frac{u_1 - u_4 + \sqrt{\Delta}}{2u_2} \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{aligned} \quad (59)$$

with

$$\Delta = u_1^2 - 2u_1 u_4 + 4u_2^2 + u_4^2 \quad (60)$$

Letting

$$\begin{aligned} \lambda_{U,1} &= -\frac{1}{2} \sqrt{\Delta} + \frac{u_1 + u_4}{2} \\ \lambda_{U,2} &= +\frac{1}{2} \sqrt{\Delta} + \frac{u_1 + u_4}{2} \end{aligned} \quad (61)$$

, $\mathbf{U}^j$ is finally computed by

$$\mathbf{U}^j = \mathbf{V}_\mathbf{U} \mathbf{S}^j \mathbf{V}_\mathbf{U}^\top = \mathbf{V}_\mathbf{U} \text{diag}(0, 0, \lambda_{\mathbf{U},1}^j, \lambda_{\mathbf{U},2}^j) \mathbf{V}_\mathbf{U}^\top \quad (62)$$

Therefore

$$(\mathcal{AB})^j = \mathbf{V}_1 \mathbf{U}^j \mathbf{V}_1^\top = \mathbf{V}_1 \mathbf{V}_\mathbf{U} \text{diag}(0, 0, \lambda_{\mathbf{U},1}^j, \lambda_{\mathbf{U},2}^j) \mathbf{V}_\mathbf{U}^\top \mathbf{V}_1^\top \quad (63)$$

Required computation of $\mathbf{V}_i$ is given by

$$\begin{aligned} \frac{1}{2}(\mathbf{W}_i + \mathbf{I}) &= \tilde{\mathbf{V}}_i \mathbf{S} \tilde{\mathbf{V}}_i^{-1}, \\ \tilde{\mathbf{V}}_i(1, 1) &= +\frac{1}{V}(D_{x,i}^b - D_{x,i}^r)(D_{z,i}^b - D_{z,i}^r) \\ \tilde{\mathbf{V}}_i(1, 2) &= -\frac{1}{V}(D_{x,i}^b - D_{x,i}^r)(D_{y,i}^b - D_{y,i}^r) \\ \tilde{\mathbf{V}}_i(1, 3) &= +\frac{1}{V}(D_{x,i}^b + D_{x,i}^r)(D_{z,i}^b + D_{z,i}^r) \\ \tilde{\mathbf{V}}_i(1, 4) &= -\frac{1}{V}(D_{x,i}^b + D_{x,i}^r)(D_{y,i}^b + D_{y,i}^r) \\ \tilde{\mathbf{V}}_i(2, 1) &= +\frac{1}{V}(D_{x,i}^b - D_{x,i}^r)(D_{y,i}^b + D_{y,i}^r) \\ \tilde{\mathbf{V}}_i(2, 2) &= +\frac{1}{V}(D_{x,i}^b - D_{x,i}^r)(D_{z,i}^b + D_{z,i}^r) \\ \tilde{\mathbf{V}}_i(2, 3) &= +\frac{1}{V}(D_{x,i}^b + D_{x,i}^r)(D_{y,i}^b - D_{y,i}^r) \\ \tilde{\mathbf{V}}_i(2, 4) &= +\frac{1}{V}(D_{x,i}^b + D_{x,i}^r)(D_{z,i}^b - D_{z,i}^r) \\ \tilde{\mathbf{V}}_i(3, 1) &= 1, \tilde{\mathbf{V}}_i(3, 2) = 0, \tilde{\mathbf{V}}_i(3, 3) = 1, \tilde{\mathbf{V}}_i(3, 4) = 0 \\ \tilde{\mathbf{V}}_i(4, 1) &= 0, \tilde{\mathbf{V}}_i(4, 2) = 1, \tilde{\mathbf{V}}_i(4, 3) = 0, \tilde{\mathbf{V}}_i(4, 4) = 1 \\ V &= (D_{y,i}^b)^2 + (D_{z,i}^b)^2 - (D_{y,i}^r)^2 - (D_{z,i}^r)^2 \end{aligned} \quad (64)$$

where $\tilde{\mathbf{V}}_i(x, y)$ stands for the element of $\tilde{\mathbf{V}}_i$ in x -th row and y -th column. Related information can also be acquired from [14]. It should be noted that

$$\tilde{\mathbf{V}}_i \tilde{\mathbf{V}}_i^\top \neq \tilde{\mathbf{V}}_i^\top \tilde{\mathbf{V}}_i \neq \mathbf{I} \quad (65)$$

Thus the Gram-Schmidt orthogonalization should be applied to $\tilde{\mathbf{V}}_i$, enabling $\mathbf{V}_i \mathbf{V}_i^\top = \mathbf{V}_i^\top \mathbf{V}_i = \mathbf{I}$ [28]. A typical commitment to achieve this is to compute the QR decomposition [29], such that

$$\mathbf{V}_i \mathbf{R} = \tilde{\mathbf{V}}_i \quad (66)$$

where $\mathbf{R}$ denotes an invertible upper triangular matrix. If $\mathbf{q}_{rand} = (1, 0, 0, 0)$, the suboptimal quaternion is equal to the normalized first column of $(\mathcal{AB})^j$.

2) *n-Vector Case*: Corresponding to the above notations and derivations, the n -vector case's transformation operators are defined by

$$\begin{aligned} \mathcal{A} &= \mathbf{V}_1 \mathbf{S} \mathbf{V}_1^\top \cdots \mathbf{V}_i \mathbf{S} \mathbf{V}_i^\top \cdots \mathbf{V}_n \mathbf{S} \mathbf{V}_n^\top \\ \mathcal{B} &= \mathbf{V}_n \mathbf{S} \mathbf{V}_n^\top \cdots \mathbf{V}_i \mathbf{S} \mathbf{V}_i^\top \cdots \mathbf{V}_1 \mathbf{S} \mathbf{V}_1^\top \end{aligned} \quad (67)$$

Defining

$$\begin{aligned} \mathbf{H} &= \mathbf{S} \mathbf{V}_1^\top \left(\prod_{i=2}^{n-1} \mathbf{V}_i \mathbf{S} \mathbf{V}_i^\top \right) \mathbf{V}_n \mathbf{S} \\ &= \frac{1}{2^{n-2}} \mathbf{S} \mathbf{V}_1^\top \left[\prod_{i=2}^{n-1} (\mathbf{W}_i + \mathbf{I}) \right] \mathbf{V}_n \mathbf{S} \end{aligned} \quad (68)$$

, we have

$$\mathbf{U} = \mathbf{S} \mathbf{V}_1^\top \cdots \mathbf{V}_i \mathbf{S} \mathbf{V}_i^\top \cdots \mathbf{V}_n \mathbf{S} \mathbf{V}_n^\top \cdots \mathbf{V}_i \mathbf{S} \mathbf{V}_i^\top \cdots \mathbf{V}_1 \mathbf{S} = \mathbf{H} \mathbf{H}^\top \quad (69)$$

Then

$$\begin{aligned} \mathcal{AB} &= \mathbf{V}_1 \mathbf{U} \mathbf{V}_1^\top \\ \Rightarrow (\mathcal{AB})^j &= \mathbf{V}_1 \mathbf{U}^j \mathbf{V}_1^\top \\ &= \mathbf{V}_1 \mathbf{V}_\mathbf{U} \text{diag}(0, 0, \lambda_{\mathbf{U},1}^j, \lambda_{\mathbf{U},2}^j) \mathbf{V}_\mathbf{U}^\top \mathbf{V}_1^\top \end{aligned} \quad (70)$$

Accordingly, the normalized first column of $(\mathcal{AB})^j$ constitutes the attitude quaternion.

3) *The Effect of Power Order*: In this sub-section we show that the selection of j is in fact not influential to the final result at all. Letting

$$\begin{aligned} g_1 &= \frac{-u_1 + u_4 + \sqrt{\Delta}}{2u_2} \\ g_2 &= \frac{u_1 - u_4 + \sqrt{\Delta}}{2u_2} \end{aligned} \quad (71)$$

we have

$$\begin{aligned} (\mathcal{AB})^j(x, y) &= \mathbf{V}_1 \mathbf{V}_\mathbf{U} \text{diag}(0, 0, \lambda_{\mathbf{U},1}^j, \lambda_{\mathbf{U},2}^j) \mathbf{V}_\mathbf{U}^\top \mathbf{V}_1^\top \\ &= [V_1(y, 4) + V_1(y, 3)g_1] [V_1(x, 4) + V_1(x, 3)g_1] \lambda_{\mathbf{U},1}^j \\ &\quad + [V_1(y, 4) + V_1(y, 3)g_2] [V_1(x, 4) + V_1(x, 3)g_2] \lambda_{\mathbf{U},2}^j \end{aligned} \quad (72)$$

Here we also have

$$\lambda_{\mathbf{U},2} > \lambda_{\mathbf{U},1} > 0 \quad (73)$$

since

$$u_1 + u_4 = h_1^2 + h_2^2 + h_3^2 + h_4^2 > 0 \quad (74)$$

and

$$\begin{aligned} \Delta - (u_1 + u_4)^2 &= -4u_1u_4 + 4u_2^2 \\ &= 4 \left[(h_1h_3 + h_2h_4)^2 - (h_1^2 + h_2^2)(h_3^2 + h_4^2) \right] \\ &= 4 \left[h_1^2h_3^2 + h_2^2h_4^2 + 2h_1h_2h_3h_4 - h_1^2h_3^2 - h_2^2h_3^2 - h_1^2h_4^2 - h_2^2h_4^2 \right] \\ &= -4(h_2h_3 - h_1h_4)^2 < 0 \end{aligned} \quad (75)$$

Therefore with increasing iteration numbers, the item multiplied by $\lambda_{\mathbf{U},1}^j$ gradually vanishes in the results. The limiting result of $(\mathcal{AB})^j$ turns out to be

$$\begin{aligned} \lim_{j \rightarrow +\infty} (\mathcal{AB})^j(x, y) &= [V_1(y, 4) + V_1(y, 3)g_2] [V_1(x, 4) + V_1(x, 3)g_2] \lambda_{\mathbf{U},2}^j \end{aligned} \quad (76)$$

And the quaternion solution is none about which column of $\mathcal{AB}^j$, which is the normalization of the following vector

$$\hat{\mathbf{q}} = \begin{pmatrix} V_1(1, 4) + V_1(1, 3)g_2 \\ V_1(2, 4) + V_1(2, 3)g_2 \\ V_1(3, 4) + V_1(3, 3)g_2 \\ V_1(4, 4) + V_1(4, 3)g_2 \end{pmatrix} \quad (77)$$

C. Discussion of OLEQs

The three derived OLEQs can be used in different occasions. The OLEQ incorporates the weights so that the determination results are optimal in the sense of least square. When there is aid of gyroscope, the ROLEQ can achieve faster and more smooth estimates. The meaning of the proposed SOLEQ is that it owns very simple linear expression that may generate short and tidy analytic results for certain sensor combinations. Also, when required in application where the weights can hardly be accurately determined e.g. vision attitude determination, the SOLEQ could be an alternative choice, as well.

TABLE I: Test Cases

Case	Reference Vectors	Noise Standard Deviations
1	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [0, 1, 0]^\top, \mathbf{D}_3^r = [0, 0, 1]^\top$	$\sigma_1 = 10^{-6}, \sigma_2 = 10^{-6}, \sigma_3 = 10^{-6}$
2	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [0, 1, 0]^\top$	$\sigma_1 = 10^{-6}, \sigma_2 = 10^{-6}$
3	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [0, 1, 0]^\top, \mathbf{D}_3^r = [0, 0, 1]^\top$	$\sigma_1 = 0.01, \sigma_2 = 0.01, \sigma_3 = 0.01$
4	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [0, 1, 0]^\top$	$\sigma_1 = 0.01, \sigma_2 = 0.01$
5	$\mathbf{D}_1^r = [0.6, 0.8, 0]^\top, \mathbf{D}_2^r = [0.8, -0.6, 0]^\top$	$\sigma_1 = 10^{-6}, \sigma_2 = 0.01$
6	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [1, 0.01, 0]^\top, \mathbf{D}_3^r = [1, 0, 0.01]^\top$	$\sigma_1 = 10^{-6}, \sigma_2 = 10^{-6}, \sigma_3 = 10^{-6}$
7	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [1, 0.01, 0]^\top$	$\sigma_1 = 10^{-6}, \sigma_2 = 10^{-6}$
8	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [1, 0.01, 0]^\top, \mathbf{D}_3^r = [1, 0, 0.01]^\top$	$\sigma_1 = 0.01, \sigma_2 = 0.01, \sigma_3 = 0.01$
9	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [1, 0.01, 0]^\top$	$\sigma_1 = 0.01, \sigma_2 = 0.01$
10	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [0.96, 0.28, 0]^\top, \mathbf{D}_3^r = [0.96, 0, 0.28]^\top$	$\sigma_1 = 10^{-6}, \sigma_2 = 0.01, \sigma_3 = 0.01$
11	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [0.96, 0.28, 0]^\top$	$\sigma_1 = 10^{-6}, \sigma_2 = 0.01$
12	$\mathbf{D}_1^r = [1, 0, 0]^\top, \mathbf{D}_2^r = [0.96, 0.28, 0]^\top$	$\sigma_1 = 0.01, \sigma_2 = 10^{-6}$

IV. SIMULATIONS AND EXPERIMENTS

A. Simulation: Common Cases

In this sub-section, the sensor observations are simulated with random reference vectors and true DCM in

$$\mathbf{D}^b = \mathbf{C}_{true} \mathbf{D}^r + \varepsilon \quad (78)$$

where ε is the noise item which is supposed to be independent and subject to Gaussian distribution. The reference vectors and the standard deviations of noise items are selected according to the classical test samples by Markley (see Table I), where the reference DCM $\mathbf{C}_{true}$ is

$$\mathbf{C}_{true} = \begin{pmatrix} 0.352 & 0.864 & 0.360 \\ -0.864 & 0.152 & 0.480 \\ 0.360 & -0.480 & 0.800 \end{pmatrix} \quad (79)$$

Using the simulated samples, the mean attitude root mean-squared errors (RMSEs) in Euler angles are evaluated with our

proposed algorithms OLEQ, SOLEQ and representative algorithms including QUEST and FLAE, which are shown in Table II, III, IV. Table V contains the computed average Wahba's loss function values by different cases and algorithms. These algorithms are executed on the MATLAB r2016 software on a PC for 10000 times with each data sample.

We first observe the attitude RMSEs. From the results of OLEQ, QUEST and FLAE, it is noticeable to determine that they have the similar attitude determination accuracy. Combining the same statistics in Table V, the proposed OLEQ is well verified for its optimality. From the presented results, we see that SOLEQ has larger attitude errors and loss function values other optimal methods. The proposed SOLEQ is sub-optimal as it actually approximates the attitude estimator where the weights are ignored. Therefore, this simulation scenario has validated the correctness and optimality of the proposed OLEQ and SOLEQ.

TABLE II: Roll RMSE (deg)

Case	OLEQ	SOLEQ	QUEST	FLAE
1	4.3516×10^{-05}	6.1268×10^{-05}	4.3516×10^{-05}	4.3516×10^{-05}
2	5.9303×10^{-05}	6.0734×10^{-05}	5.9303×10^{-05}	5.9303×10^{-05}
3	4.3482×10^{-01}	6.0730×10^{-01}	4.3482×10^{-01}	4.3482×10^{-01}
4	6.0292×10^{-01}	6.1798×10^{-01}	6.0292×10^{-01}	6.0292×10^{-01}
5	4.3313×10^{-01}	4.3313×10^{-01}	4.9065×10^{-01}	$2.4281 \times 10^{+01}$
6	4.9590×10^{-03}	3.0793×10^{-01}	4.9590×10^{-03}	4.9590×10^{-03}
7	8.1132×10^{-03}	$1.3400 \times 10^{+00}$	8.1132×10^{-03}	8.1132×10^{-03}
8	$5.9553 \times 10^{+01}$	$6.2840 \times 10^{+01}$	$5.9553 \times 10^{+01}$	$5.9553 \times 10^{+01}$
9	$7.6662 \times 10^{+01}$	$7.6696 \times 10^{+01}$	$7.6662 \times 10^{+01}$	$7.6662 \times 10^{+01}$
10	$1.4313 \times 10^{+00}$	$1.7781 \times 10^{+00}$	$4.7663 \times 10^{+01}$	$1.5265 \times 10^{+00}$
11	$2.0254 \times 10^{+00}$	$2.0254 \times 10^{+00}$	$4.6392 \times 10^{+01}$	$2.8441 \times 10^{+00}$
12	$2.0818 \times 10^{+00}$	$2.0888 \times 10^{+00}$	$3.7218 \times 10^{+01}$	$1.7415 \times 10^{+01}$

TABLE IV: Yaw RMSE (deg)

Case	OLEQ	SOLEQ	QUEST	FLAE
1	4.3587×10^{-05}	6.1084×10^{-05}	4.3587×10^{-05}	4.3587×10^{-05}
2	4.8694×10^{-05}	6.1015×10^{-05}	4.8694×10^{-05}	4.8694×10^{-05}
3	4.4127×10^{-01}	6.0868×10^{-01}	4.4127×10^{-01}	4.4127×10^{-01}
4	4.8593×10^{-01}	6.1289×10^{-01}	4.8593×10^{-01}	4.8593×10^{-01}
5	2.5186×10^{-01}	2.5186×10^{-01}	2.8536×10^{-01}	$1.7459 \times 10^{+01}$
6	3.6421×10^{-05}	6.1651×10^{-05}	3.6421×10^{-05}	3.6421×10^{-05}
7	4.8748×10^{-05}	6.0826×10^{-05}	4.8748×10^{-05}	4.8748×10^{-05}
8	3.9812×10^{-01}	6.1557×10^{-01}	3.9812×10^{-01}	3.9812×10^{-01}
9	4.9366×10^{-01}	6.1163×10^{-01}	4.9366×10^{-01}	4.9366×10^{-01}
10	6.1834×10^{-05}	6.1834×10^{-05}	6.1836×10^{-05}	6.1844×10^{-05}
11	6.2069×10^{-05}	6.2069×10^{-05}	6.2069×10^{-05}	6.2077×10^{-05}
12	3.1726×10^{-01}	6.1275×10^{-01}	$5.1968 \times 10^{+00}$	$2.8814 \times 10^{+00}$

TABLE III: Pitch RMSE (deg)

Case	OLEQ	SOLEQ	QUEST	FLAE
1	4.0108×10^{-05}	5.7335×10^{-05}	4.0108×10^{-05}	4.0108×10^{-05}
2	5.2860×10^{-05}	5.6736×10^{-05}	5.2860×10^{-05}	5.2860×10^{-05}
3	4.0104×10^{-01}	5.6744×10^{-01}	4.0104×10^{-01}	4.0104×10^{-01}
4	5.3887×10^{-01}	5.7656×10^{-01}	5.3887×10^{-01}	5.3887×10^{-01}
5	3.9149×10^{-01}	3.9149×10^{-01}	4.4335×10^{-01}	$1.2561 \times 10^{+01}$
6	4.0121×10^{-05}	5.7809×10^{-05}	4.0121×10^{-05}	4.0121×10^{-05}
7	5.3398×10^{-05}	5.7657×10^{-05}	5.3398×10^{-05}	5.3398×10^{-05}
8	3.6755×10^{-01}	5.7326×10^{-01}	3.6755×10^{-01}	3.6755×10^{-01}
9	4.5938×10^{-01}	5.7880×10^{-01}	4.5938×10^{-01}	4.5938×10^{-01}
10	5.7186×10^{-05}	5.7186×10^{-05}	5.7184×10^{-05}	5.7186×10^{-05}
11	5.7845×10^{-05}	5.7845×10^{-05}	5.7846×10^{-05}	5.7844×10^{-05}
12	4.9161×10^{-01}	5.7554×10^{-01}	$7.9376 \times 10^{+00}$	$3.9359 \times 10^{+00}$

TABLE V: Loss Function Values

Case	OLEQ	SOLEQ	QUEST	FLAE
1	5.0651×10^{-13}	1.0130×10^{-12}	5.0651×10^{-13}	5.0651×10^{-13}
2	2.4901×10^{-13}	4.9802×10^{-13}	2.4901×10^{-13}	2.4901×10^{-13}
3	4.9338×10^{-05}	9.8666×10^{-05}	4.9338×10^{-05}	4.9338×10^{-05}
4	2.5369×10^{-05}	5.0736×10^{-05}	2.5369×10^{-05}	2.5369×10^{-05}
5	5.0582×10^{-13}	5.0582×10^{-13}	6.5095×10^{-13}	8.5878×10^{-10}
6	5.0422×10^{-13}	9.4333×10^{-10}	5.0422×10^{-13}	5.0422×10^{-13}
7	2.4728×10^{-13}	8.8239×10^{-09}	2.4728×10^{-13}	2.4728×10^{-13}
8	4.8216×10^{-05}	1.1593×10^{-04}	4.8216×10^{-05}	4.8216×10^{-05}
9	2.5327×10^{-05}	5.0651×10^{-05}	2.5327×10^{-05}	2.5327×10^{-05}
10	1.4827×10^{-12}	1.7575×10^{-12}	4.8431×10^{-10}	1.7195×10^{-12}
11	4.8573×10^{-13}	4.8573×10^{-13}	2.4106×10^{-10}	9.2333×10^{-13}
12	5.0105×10^{-13}	5.0105×10^{-05}	1.3143×10^{-10}	3.4336×10^{-11}

B. Simulation: Extreme Cases

Conventional Wahba's solutions face dilemma when exposed to some critical pairs of vector observations. For instance, Markley and Mortari give an example where the sensors are configured by [30]

$$\mathbf{D}_1^r = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{D}_2^r = \begin{pmatrix} -0.99712 \\ 0.07584 \\ 0 \end{pmatrix}, \mathbf{D}_3^r = \begin{pmatrix} -0.99712 \\ -0.07584 \\ 0 \end{pmatrix}$$

$$\sigma_1 = 1 \text{ arcsec}, \sigma_2 = \sigma_3 = 1 \text{ deg} \quad (80)$$

In such scenario, the root of the characteristic polynomial of the Davenport matrix can not be easily obtained by Newton iterations. The internal reason is given by Cheng [31] showing that it is resulted in by numerical loss according to the specific CPU word storage length. A flexible transformation of the characteristic polynomial is proposed then to significantly boost the convergence. As such configurations indeed happen in engineering practice, there is necessity to evaluate the proposed schemes by comparisons with representative solvers. With similar simulation techniques aforementioned, the vectors are simulated with given reference vectors and standard deviations by rotation of $\mathbf{C}_{true}$. Here the QUEST algorithm is revised to the Cheng's form. First, we mainly compare the two iterative methods QUEST and OLEQ because in our existing paper [14] the QUEST and FLAE have been proved to have very similar behaviour facing this extreme case. Here the iteration stops when the Euclidean norm of neighboring attitude quaternion difference is less than 1×10^{-8} . For QUEST, the maximum iteration number is set to 50. The obtained results are depicted in Fig. 1.

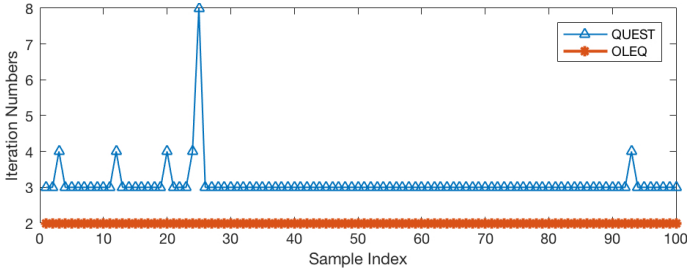


Fig. 1: Iteration numbers of QUEST and OLEQ in face of an extreme case.

We can see that the supervised QUEST can obtain accurate quaternion solutions within several iterations. Actually, before the Cheng's improvement, the QUEST may exceed the maximum iteration number from time to time. The proposed OLEQ, however, shows better performance dealing with this extreme case. Also, the final mean loss function values of the two algorithms are computed as 4.9890×10^{-11} and 2.8391×10^{-10} , which revealing that the proposed OLEQ can not only obtain faster solutions, but leads to smaller loss function values, compared with supervised QUEST. As is known to everyone, QUEST is the most representative Wahba's solution using Davenport's q-method. Many other algorithms like ESOQ, FOAM actually have the same performance with QUEST. Therefore, in this way, the OLEQ is proved to be faster and

more robust than the whole class of the algorithms based on Davenport's q-method. This also shows that the presented novel attitude evolution method shows brand new abilities.

C. Experiment: Accelerometer-Magnetometer Case

In this sub-section, we conduct an experiment where the accelerometer-magnetometer combination is adopted. Such sensor combination is extensively applied in nowadays low-cost attitude estimation schemes. The accelerometer is pre-calibrated using the 6-face bias cancelling while the magnetometer is calibrated using the method proposed by Y. Wu et al. [32].

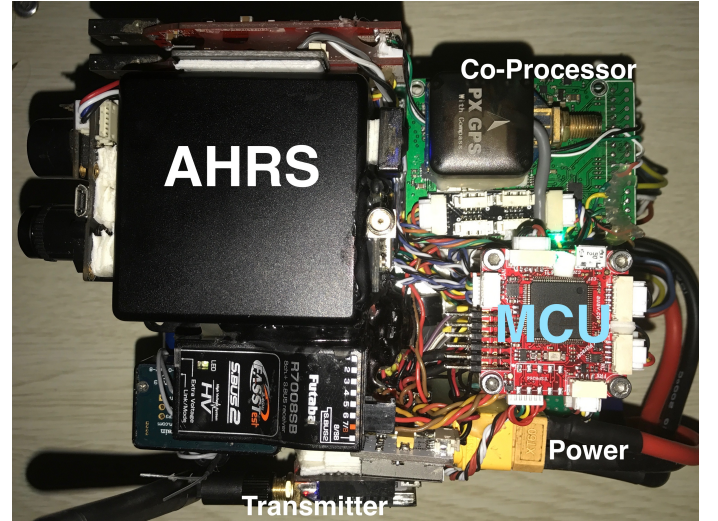


Fig. 2: Designed hardware for algorithm implementation.

The hardware is constituted by a battery, a high-end attitude and heading reference system (AHRS) with high precision internal accelerometer, magnetometer and gyroscope, a transmitter for remote data transmission and a micro controller for implementation of the algorithm using C++ language on the FreeRTOS. With the designed hardware platform shown in Fig. 2, we collect a data set with 10000 samplings.

The main purpose of this sub-section is to validate the performances of the proposed OLEQ, SOLEQ and ROLEQ since the AHRS has angular rate readouts. The compared results with the reference angles from representative methods are obtained (see Fig. 3, 4, 5). Note that here the weights between the accelerometer and magnetometer for Wahba's solution are chosen as 0.63 and 0.37 according to their respective noise characteristics. Yet, the local magnetometer's reference vector is calculated as $\mathbf{M}^r = (0.60311, 0, -0.79766)^T$ in Wuxi, Jiangsu Province, China.

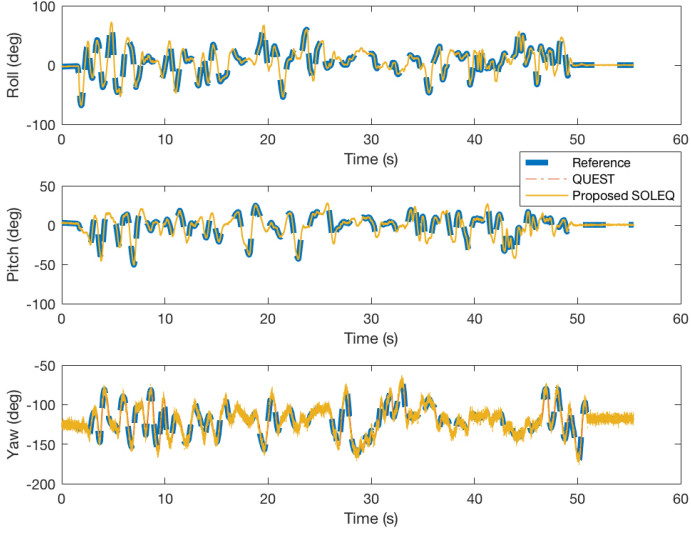


Fig. 3: Experiment results using QUEST and SOLEQ.

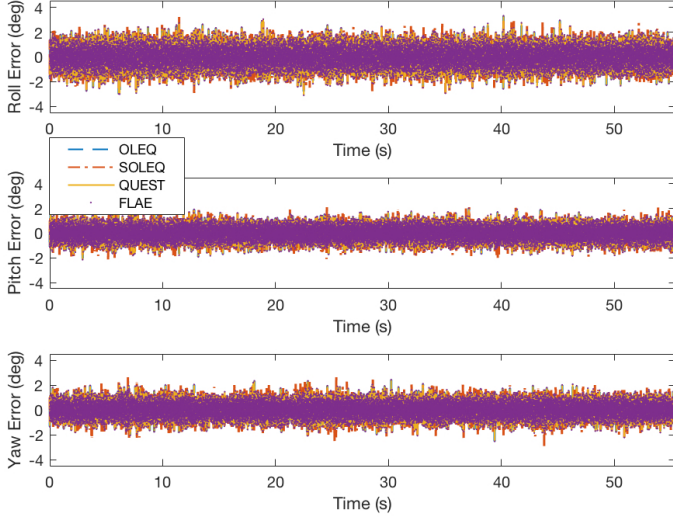


Fig. 4: Experiment results using sampled data and different algorithms.

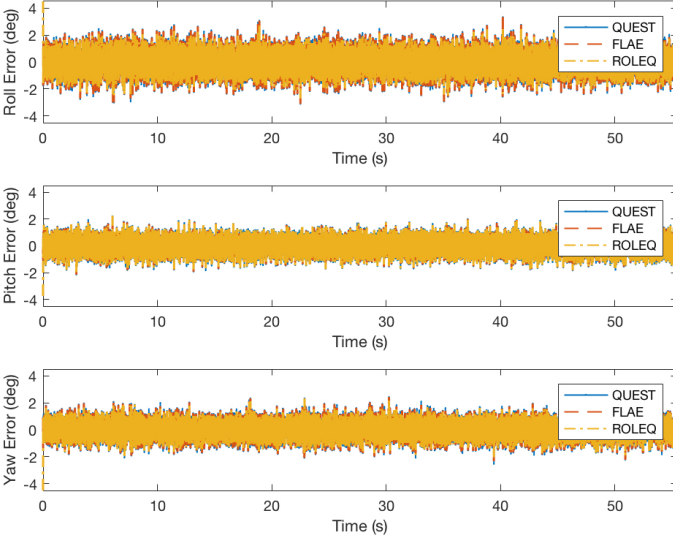


Fig. 5: Experiment results using sampled data and different algorithms.

In Fig. 3, the reference angles, QUEST solutions and SOLEQ solutions have been presented. The results indicate that the proposed sub-optimal estimator can estimate the attitude angles with similar macroscopic accuracy. Detailed attitude errors are plotted in Fig. 4 and 5. We may notice that in general, these algorithms have the same errors with respect to reference. The second picture shows that the overall attitude accuracy of the ROLEQ is slightly smaller than the others. This is because it is first processed by the angular rate data, which can be equivalent to a smoothing procedure.

D. Computation Time

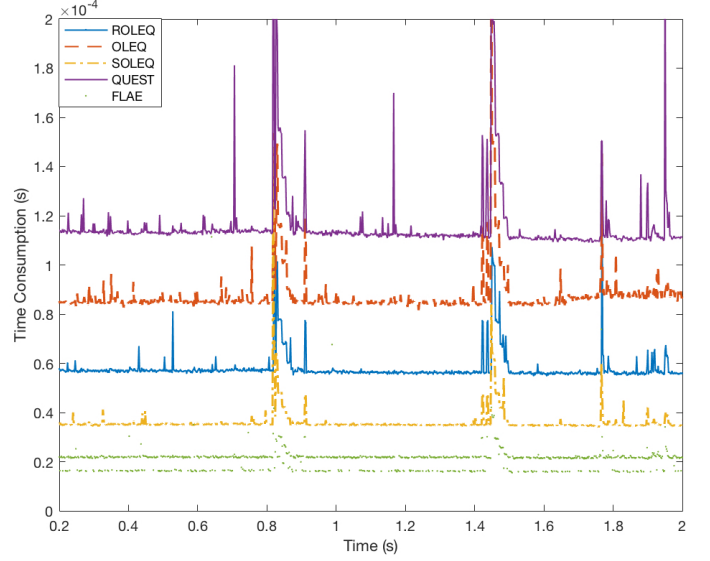


Fig. 6: Time consumption of various algorithms.

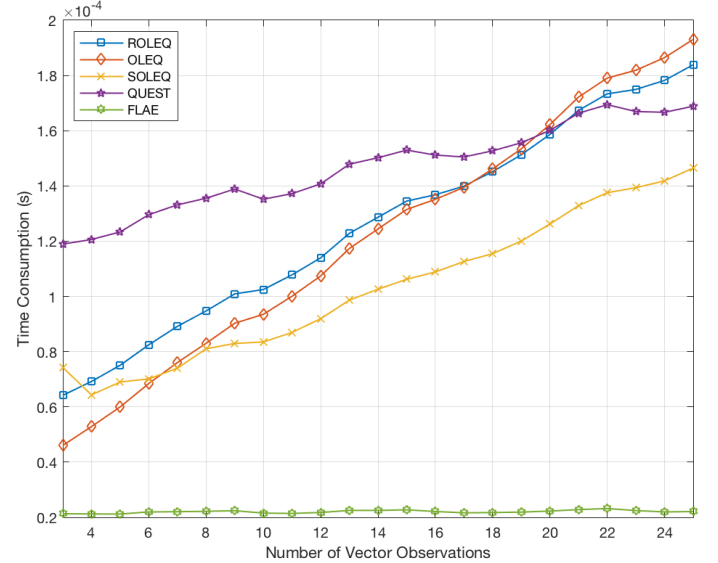


Fig. 7: Time consumption of algorithms with respect to numbers of vector observations.

From another point of view, the time consumption of various algorithms should be investigated. The time consumption is calculated on the embedded platform by C++ programming language that ensures the fairness. A rough evaluation is

done with two pairs of vector observations in few samples which shows direct time consumption results (see Fig. 6). As shown in the figure, for two pairs of vector observations, the three proposed algorithms' computation times are between QUEST and FLAE. However, from the expressions of the algorithms presented before, the number of vector observations is influential to the final time consumption. Hence with the simulation samples, each algorithm is again tested for 20000 times with different vector observation numbers. The time consumption is averaged, which is plotted in Fig. 7. The results show that the algorithms are all linear owing the time complexities of $O(n)$. QUEST, OLEQ and ROLEQ join at the vector observation number of 20. For common tasks, such number covers most sensor amounts. Although FLAE owns the least time consumption, it can not overcome drawbacks of extreme cases so well as OLEQ. That is to say, the proposed algorithms can replace the original algorithms for faster and more robust attitude determination.

V. CONCLUSION

This paper revisits the attitude determination from vector observations. Novel linear algorithms are designed to obtain accurate attitude estimates in the sense of least-square or regardless of the weights' effect. Handling in this manner, the computed quaternion is identical or suboptimal with respect to conventional Wahba's solutions including QUEST and FLAE. Numerical simulations exhibit that the proposed OLEQs own the similar accuracy with representative solvers. It is also evaluated that facing extreme cases, the OLEQs show much more robustness less computation iterations. The computation speeds of OLEQs are tested revealing that they belong to computationally efficient algorithms. Moreover, an experiment of accelerometer-magnetometer attitude determination is conducted showing the effectiveness of the proposed OLEQs in real-world embedded applications. The presented approach provides the audience with a brand new viewpoint of attitude evolution and hopefully would benefit related applications.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China under the grant number of 41604025. Dr. F. L. Markley provided constructive suggestions on the internal theory of this paper. Yunheng Inc., Chengdu gave us the instrumentation services during experimental validation. We genuinely thank them for their support.

REFERENCES

- [1] 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.
- [2] Z. Zhou, Y. Li, J. Zhang, and C. Rizos, "Integrated Navigation System for a Low-Cost Quadrotor Aerial Vehicle in the Presence of Rotor Influences," *Journal of Surveying Engineering*, vol. 4, no. May, pp. 1–13, 2016.
- [3] G. Chang, T. Xu, and Q. Wang, "M-estimator for the 3D symmetric Helmert coordinate transformation," *Journal of Geodesy*, 2017.
- [4] Y. Wu, J. Wang, and D. Hu, "A New Technique for INS / GNSS Attitude and Parameter Estimation Using Online Optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2642–2655, 2014.
- [5] L. Chang, F. Qin, and F. Zha, "Pseudo Open-Loop Unscented Quaternion Estimator for Attitude Estimation," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4460–4469, 2016.
- [6] S. Han and J. Wang, "Quantization and colored noises error modeling for inertial sensors for GPS/INS integration," *IEEE Sensors Journal*, vol. 11, no. 6, pp. 1493–1503, 2011.
- [7] L. Chang, J. Li, and S. Chen, "Initial Alignment by Attitude Estimation for Strapdown Inertial Navigation Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 3, pp. 784–794, 2015.
- [8] G. Wahba, "A Least Squares Estimate of Satellite Attitude," p. 409, 1965.
- [9] 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.
- [10] F. L. Markley, "Attitude Determination using Vector Observations and the Singular Value Decomposition," pp. 245–258, 1988.
- [11] 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.
- [12] P. B. Davenport, "A vector approach to the algebra of rotations with applications," Tech. Rep. August, 1968. [Online]. Available: <http://hdl.handle.net/2060/19680021122>
- [13] G. Chang, T. Xu, and Q. Wang, "Error analysis of Davenport's q method," *Automatica*, vol. 75, pp. 217–220, 2017. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0005109816303624>
- [14] J. Wu, Z. Zhou, R. Li, and L. Yang, "Attitude Determination Using a Single Sensor Observation: Analytic Quaternion Solutions and Property Discussion," *IET Science, Measurement & Technology (Accepted)*, 2017.
- [15] Y. Yang and Z. Zhou, "An analytic solution to Wahba problem," *Aerospace Science and Technology*, vol. 30, no. 1, pp. 46–49, 2013.
- [16] 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.
- [17] J. R. Forbes and A. H. J. de Ruiter, "Linear-Matrix-Inequality-Based Solution to Wahba's Problem," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 1, pp. 147–151, 2015.
- [18] R. Mahony, T. Hamel, and J. M. Pfimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [19] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *IEEE International Conference on Rehabilitation Robotics*, 2011.
- [20] Y. Tian, H. Wei, and J. Tan, "An adaptive-gain complementary filter for real-time human motion tracking with MARG sensors in free-living environments," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 2, pp. 254–264, 2013.
- [21] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.
- [22] C. Ye, S. Hong, and A. Tamjidi, "6-DOF Pose Estimation of a Robotic Navigation Aid by Tracking Visual and Geometric Features," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1169–1180, 2015.
- [23] 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.
- [24] —, "How to estimate attitude from vector observations," *Advances in the Astronautical Sciences*, vol. 103, no. PART III, pp. 1979–1996, 2000.
- [25] F. L. Markley, "Humble problems," *Advances in the Astronautical Sciences*, vol. 124 II, pp. 2205–2222, 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] M. D. Shuster, "Filter QUEST or REQUEST," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 643–645, 2009.
- [28] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [29] Y. Yang, "Spacecraft attitude determination and control: Quaternion based method," *Annual Reviews in Control*, vol. 36, no. 2, pp. 198–219, 2012.
- [30] F. L. Markley and D. Mortari, "Quaternion Attitude Estimation Using Vector Observations," *Journal of the Astronautical Sciences*, vol. 48, no. 2/3, pp. 359–380, 2000.
- [31] 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.

- [32] Y. Wu and W. Shi, "On Calibration of Three-Axis Magnetometer," *IEEE Sensors Journal*, vol. 15, no. 11, pp. 6424–6431, 2015.