Spacecraft attitude determination methods in an educational context

ERIC ENGER
Spacecraft attitude determination methods in an educational context

Eric Enger
MSc Student – Aerospace Engineering
KTH
Stockholm - Sweden
ericre@kth.se

Abstract: This work has as an objective to structure the content of a course on Attitude determination methods, part of an Aerospace Engineering Master program. A selection of books, papers, theses, web sites and films was reviewed to identify the most relevant topics within the areas of Static and Dynamic Attitude Determination and the ways to present them in an educational context. Theory is presented in a simplified way and examples were gathered to illustrate the theoretical part. Finally, a discussion is carried out on the main learning goals and challenges, required time for instruction and exercises and suggestion for a grading system.

Keywords: Attitude determination, attitude estimation, TRIAD, Wahba’s problem, Devenport’s method, QUEST method, OLAЕ method, Kalman filter, Extended Kalman filter, Multiplicative extended Kalman filter

I. INTRODUCTION

In the space industry, one critical aspect that must be considered for a successful mission is the attitude determination of a spacecraft in relation to an inertial reference frame. That information will be constantly fed into the control system, which will evaluate and proceed with the actions required to achieve desired attitude.

Considering an educational context, by that meaning an Aerospace engineering program in a master’s degree level, Attitude determination is a subject that should be included in a Spacecraft dynamics course, together with Rigid body kinematics, Eulerian mechanics, Analytical dynamics and Stability and control, to give the student a complete view of the problems being studied.

By writing this work, it was assumed that the reader has already a basic knowledge of Attitude determination methods, as the intention here is to help him/her to prepare the content of a course on this matter. Some equation derivations are presented in the appendixes, but it was a general assumption that the reader shall find them in the references.

The purpose of this work is to identify relevant topics to be included in a course as mentioned before, considering that the student has already gone through the Rigid body kinematics part and has a reasonable knowledge about it in general and specially about attitude parameters as Direction cosine matrix (DCM), Euler parameters (quaternions), Classic Rodrigues parameters (CRP our Gibbs vector). As it will be explained later, basic knowledge on statistics theory, as well as on quaternion algebra are also required.

The work starts with a literature review in section II, where the researched references are discussed in terms of their content relevance as educational aids in an Attitude determination part of a course. After that review, the content of the of the course is defined as being part of two main topics: Static attitude determination methods, consisting of TRIAD, Wahba’s problem, Devenport’s q-method, Quaternion estimation method and Optimal linear attitude estimator and Dynamic attitude determination methods consisting of Kalman filtering and its variations, e.g. Extended Kalman filter and Multiplicative extended Kalman filter.

List of Symbols

\( B \) Designates a vector in the body frame
\( N \) Designates a vector in the inertial frame
\( \hat{\cdot} \) Designates a unit vector (section III A)
\( s \) Sun sensor vector
\( m \) Magnetic field vector
\( C_{BN} \) Direction cosine matrix from inertial to estimated body frame
\( t_i \) TRIAD vector
\( \theta \) Rotation vector
\( \theta \) Rotation from B to N frames around principal axis
\( J \) Cost function
\( g \) Gain function
\( \bar{q} \) Quaternion
\( p \) Classic Rodrigues parameters

The symbols listed above are general symbols, those specific for Kalman filtering are listed in Table 1.
In section III, each one of those topics are explored by explaining their respective theories and by showing their characteristics, advantages and disadvantages. The topics are illustrated by examples for a better understanding of the theory.

The discussion in section IV, aims to point out the most important aspects of the content presented in section III, explore the challenges that a student may have in understanding the theory, establish a grading system according to the level of difficulty of the examples presented and the required work load for achieving the learning goals.

II. LITERATURE REVIEW

To address the general scope of the work presented in the introduction, a literature review was carried out to map the already available material in Attitude Determination.

The research starts revisiting Kinematics in chapter 3 of reference [1], where the methods of describing orientation are presented through the concepts of Direction Cosine Matrix (DCM), Euler Angles, Principal Rotation Vector (PRV), Euler Parameters (quaternions), Classical Rodrigues Parameters (CRP) and Modified Rodrigues Parameters (MRP). For each of these rigid body attitude coordinate sets, the most important topics treated are mathematical definitions of the parameters and their properties, the redundancies and/or singularities involved, how to deal with singularities when they occur, relationship between the set of parameters and DCM and pertaining kinematic differential equations. There are also examples that illustrate the use of each coordinate set.

Although Schaub & Junkins [1] writes in didactic and accessible way, in lectures 1, 2 and 3 [2] he presents the same material in the form of a series of lecture videos, which substantially helps understanding the subject. In the first part, a set of 13 videos are dedicated to an introduction to kinematics, where the concepts of particle kinematics, angular velocity vector and vector differentiation are revisited, in a total of 111 minutes. The second and third parts cover Rigid Body Kinematics, approximately the same content as in [1]. Part 2 deals with DCM and Euler Angles in 18 videos and a total of 190 minutes of recorded material, whereas in part 3 PRV, quaternions, CRP and MRP are presented in 27 videos and a total of 216 minutes.

Once attitude description methods were reviewed, it is possible to move on to methods of attitude determination. Reference [2] fourth video is the continuation of the lecture series mentioned before and deals with the problem of static attitude determination. It is meant by static that a set of instantaneous position observations is made, and the problem becomes the one of how to solve the geometry of those measurements. At this point no rotation rates are considered, which will be used later in the dynamic attitude determination. Schaub [2] begins with the simple problem of attitude determination in the 2D space, which gives a good start for understanding the concept. Then he moves on to 3D, where the necessity of using at least two vectors is explained.

The Triaxial Attitude Determination method (TRIAD) is a consequence of that requirement and will build up a reference frame using those two observation vectors. Those vectors are assumed to be known in the inertial frame (position of the sun, magnetic field or a certain star) and can thus be compared with the measurements in the body frame to determine the attitude of the body in relation to the inertial frame.

When uncertainties in the measurements are considered and more than two observation vectors are taken into account, a more sophisticated method is required. Schaub explains how to state the problem in the form of a minimum squares, which is known as Wahba’s problem. The solution of that problem is the attitude estimation that we are looking for and three ways of solving it are proposed: Devenport’s q-method, quaternion estimator (QUEST) and Optimal linear attitude estimator (OLAE). Reference [3] covers in chapter 5 the Static Attitude Determinations methods in a deeper and more extensive way, presenting more methods to solve Wahba’s problem, and to perform error analysis.

In reference [4] Yang presents the quaternion-based method for spacecraft attitude determination and control. It begins with a brief but complete discussion on quaternion algebra and it moves on to spacecraft dynamics and modelling. Wahba’s problem and Devenport’s formula are described and some methods for quaternion estimation are discussed: QUEST, Fast Optimal Attitude Matrix (FOAM) and Estimator of the Optimal Quaternion (ESOQ).

Adding complexity, we go further to the Dynamic Attitude Determination methods. The observation is not a unique instantaneous measurement, but a series of measurements over time. Angular rate in form of gyros signal may also be used to enhance the quality of the estimation. Markley & Crassidis [3] describe in chapter 6 the use of Kalman filtering for attitude estimation considering the use of quaternions as orientation parameters. It is assumed that the reader has a previous knowledge of concepts and properties of Kalman Filters and estimation theory. In chapter 12 of [3] a revision on Estimation Theory is presented, but for a first-time reader, smaller steps should be taken for a better understanding of the subject.

Reference [5] starts with a review of Random Signals in its first part and then moves on to a very comprehensive study of Kalman Filtering and its applications, with many illustrative examples. For the present study, chapter 4 about discrete Kalman filter basics and chapter 7 on linearization and nonlinear filtering were extremely useful to fill the gaps required to understanding chapter 6 of [3].

Crassidis, Markley and Cheng [6] present a survey of nonlinear estimation methods available for attitude determination. The authors start by reviewing quaternion parametrization and describing the gyro model. They go then on describing the Extended Kalman Filter and some of its variations as additive EKF, multiplicative EKF, deterministic EKF-like estimator. Other techniques are also described such as QUEST-based methods, two-step attitude estimator, unscented filtering, particle filters, orthogonal attitude filter, among others. The intent of the paper is to present in a rather summarizes way the different alternative techniques, their main characteristics, applications and differences.

Rhudy, Salguero and Holappa [7] present a tutorial on Kalman filtering designed for instruction to undergraduate students. They use a simple language to introduce the idea
behind Kalman filters and its algorithm. A simple example of an object in freefall is used to illustrate the linear Kalman filter use. The algorithm is very well detailed, the results are presented in the form of graphs and the MATLAB code is supplied in the appendix. EKF is also presented, but in a more superficial way. An interesting introduction to covariance matrix was also included with some figures representing different sorts of distributions.

Another presentation of Kalman filtering in a very simple and accessible way was found in Bzarg [8], a blog maintained by Tim Babb, who works for Pixar animation studios. The text is very focused in the conceptual aspect of Kalman filtering with very good pictures representing the covariances and uncertainties in each step of the algorithm. Babb uses different colours to identify the several different elements involved in the algorithm, which could be an interesting didactical tool.

The use of EKF depends on an accurate knowledge of the system dynamics, the measurements models and their respective noise characteristics. As the latter are normally unavailable or not reliable, Kiani, Barzegar & Pourtakdoust [9] present a more sophisticated version of EKF that stores the innovation sequence and the state prediction residuals in order to produce estimates of the measurement and process noises, respectively. Besides, new techniques are introduced to define the appropriate size of the stored information and to guarantee that the covariance matrices remains positive definite. The subject of that paper is more advanced than the scope of this work, but it may be used as an example for further studies.

In reference [10] Markley presents the differences between two mathematical models considering quaternions as attitude parameter. The first one is the additive model, where the estimation error is represented by difference between the true state and the estimate. In contrast with that, in the multiplicative model the error is represented by a quaternion rotation (quaternion multiplication). Markley presents EKF formulations for both models and concludes comparing their pros and cons.

Markley continues in [11] with a complete development of the Multiplicative EKF (MEKF) starting with a brief summary of the main parametrizations used in spacecraft attitude determination (quaternions, Gibbs vector and modified Rodrigues parameters) and then a description on how the attitude error representation, state and covariance propagation, measurement model and updated should look like. The author also presents the second order filter, which considers second-order terms in the nonlinear Kalman Filter, to be used when nonlinearities are significant relative to the measurement and noise terms.

Trawny & Roumeliotis [12] present a very well structured and concise explanation of the structure and use of MEKF. They start reviewing the elements of quaternion algebra needed to understand MEKF and in the second part they carry out a step-by-step construction of the algorithm to finalize with the propagation and update equations. Dugas [13] presents a MATLAB code based on the theory and equations developed in [12] that could be used in exercises, projects or demonstration of MEKF use.

Some videos found on Internet could help students to understand KF concepts and its utilization. MATLAB [14] has a very well-produced series of seven short films (up to nine minutes) covering the topics of what a KF does, the concepts involved and how the algorithm works in a very accessible language and full of graphics and examples.

In another, not as well-produced, series of three videos (up to fourteen minutes), a certain Student Dave [15] presents concepts and examples. The imaging is basically his handwriting on a white board, but despite the low-budget production, his explanations are very accessible and easy to understand.

A very good introduction to KF is given by Biezen [16] in a series of 55 short videos. Of special interest are the 6 first videos explaining how the algorithm works and giving a very nice and simple example. A one-dimensional temperature sensor is modelled and Biezen and he goes through the algorithm step by step in a very illustrative way.

Stachniss [17] from University of Bonn made available a whole course on “Simultaneous Localization and Mapping”, consisting of 20 lecture recording videos with durations varying between twenty and one hundred minutes. Of interest for the subject of this work are the videos three through six that cover Bayesian filters, Extended and Unscented Kalman filters.

Some master theses were reviewed, which results could be used as examples of use of Kalman Filters in attitude estimation. Marmion [18] presents in his work the results using a gyro and a quaternion based additive EKF. There is an interesting comparison between the results from the gyro alone, EKF and another estimator supplied by the gyro’s manufacturer.

Keil [19] uses Euler angle based EKF to determine orbit and attitude of a satellite in Molniya Orbit. Despite being a long and very comprehensive work, it is of special interest section 4 on attitude determination. The author explains how the matrices are built and applies EKF to for different sensors separately: a magnetometer, an Earth sensor, a sun sensor and a star sensor. In the results it was included a comparison of the error distribution obtained for each sensor.

Kleinbauer [20] presents a very well-structured description of linear KF and EKF, which concludes with a table having side by side the equations for both algorithms. The author also explains how to implement EKF in MATLAB and supplies an example simple calculating the orbit of a geostationary satellite.

Raiti [22] studies different configurations of the attitude determination and control system for the MIST satellite (Miniature STudent satellite) from KTH. Although it goes very deeply in the specific case of that satellite, it still is a very good example project application of both attitude determination and spacecraft control theory. Specific for this study, Raiti’s theoretical part of Multiplicative Extended Kalman Filter was very enlightening as well the clever way to limit the problems caused by linearization.
A. General Description

The attitude of a spacecraft is a set of at least three independent quantities or any parametrization of the attitude matrix, that basically indicates the rotational position of the body in relation to an inertial reference frame. In other words, how much and around which axis the body should be rotated so that its reference system matches the inertial reference frame.

Attitude determination methods can roughly be divided in two categories: static and dynamic. Static methods are time independent, where all measurements (e.g. sun vector, magnetic field vector, etc.) are taken at the same time, or close enough in time that spacecraft motion between observations can be ignored or easily compensated for. It is a deterministic approach and there is no need of information about past states. The problem is then to solve the geometry of those measurements in the body reference frame and compare them with their corresponding known description in the inertial reference frame.

At first sight a static attitude determination method may seem of not much use in a space environment, where most of the processes are time-dependent, but not only has its study an educational importance, but it also can be thought as the problem to be solved within a time step of a dynamic attitude determination method.

Dynamic attitude determination methods, beside taking into account movement and, consequently, being time-dependent, they also consider measurement not as a deterministic process but one where random noise is present, which requires a statistical approach. Filtering methods, as Kalman filters, are used to organize information from past and actual measurements, the knowledge of spacecraft motion together with possible errors in system dynamics model. The result is better referred as being an attitude “estimation” instead of “determination” because of the statistical nature of those methods.

B. Static Attitude Determination Methods

Schaub [2] presents following methods for static attitude determination:

a) Triaxial Attitude Determination (TRIAD) method:

Starting with the analogy in the two-dimension space, the attitude problem of finding the heading can be solved by determining one direction, which can be measured for instance from a compass, and then comparing it with a known representation of the environment, for instance a map. In the 2D problem, only one vector was required to define an attitude.

In the 3D problem, at least two observation vectors will be needed, one to define an axis and the other to define the rotation about that axis. Those two vectors define a plane, which together with an orthogonal vector may define a reference frame. Mapping that frame to the body reference frame and comparing it with the two vectors known representation in the inertial reference frame will give the relative attitude of the body to the inertial frame.

That is better understood with the simple example of having a sun sensor and a magnetic sensor in a spacecraft in Earth orbit. The sun sensor gives a body frame measurement unit-vector $\hat{s}$, and the magnetic sensor, $\hat{m}$. Assuming that those vectors are known in the inertial frame and are represented by $s$ and $m$, respectively, the challenge is to discover the Direction cosine matrix $C_{BN}$ that maps the vectors’ description in the inertial frame to that in the body frame, as expressed in (1) and (2):

\[
\hat{s} = C_{BN} s \quad (1)
\]

\[
\hat{m} = C_{BN} m \quad (2)
\]

It is preferred to use $C_{BN}$ instead of $C_{BN}$, to indicate that we are dealing with an estimated body frame. Some sources for uncertainty may be, among others, low measurement precision, noise, dynamic nature of the magnetic field and orbital fluctuations.

It should be noted that each observation vector (unit direction vector) contains two independent degrees of freedom. However, the 3D attitude problem is a three-degree of freedom problem and by using the measurements of the two observation directions, the attitude determination problem is always an over-determined problem.

One way to find the $C_{BN}$ matrix is to use the TRIAD Method. This method was the first algorithm used for determining spacecraft attitude and it begins constructing an intermediate reference frame with the two observation vectors. The first unit-vector of the frame will have the same direction and sense as the most accurate of the two observation vectors. In the case of sun and magnetic vectors, the sun vector usually is the most accurate (3). The second unit-vector is the orthogonal vector to the plane defined by the sun and magnetic vectors, as shown in (4) and the third one is orthogonal to the first and second vectors, as in (5).

\[
\hat{t}_1 = \hat{s} \quad (3)
\]
\[
\hat{t}_2 = \frac{\hat{s} \times \hat{m}}{|\hat{s} \times \hat{m}|} \quad (4)
\]
\[
\hat{t}_3 = \hat{t}_1 \times \hat{t}_2 \quad (5)
\]

Figure 1 illustrate the vectors.
Those three vectors can be written in the body frame coordinates and form the intermediate to body DCM shown in (6) and they can also be mapped in the inertial frame and form the intermediate to inertial DCM shown in (7):

\[ C_{BT} = \begin{bmatrix} \beta \hat{t}_1, \beta \hat{t}_2, \beta \hat{t}_3 \end{bmatrix} \]  
\[ C_{NT} = \begin{bmatrix} N \hat{t}_1, N \hat{t}_2, N \hat{t}_3 \end{bmatrix} \]

The final DCM that we are looking for, defining the body attitude with respect to the inertial frame is the multiplication of those two DCMs as per (8):

\[ C_{BN} = C_{BT}C_{NT} \]

The TRIAD method has the advantage of being very simple and intuitive but has the disadvantage of only allowing the use of two vectors. Besides, the information from the second vector (magnetic vector) is only partially used, as part of the cross product to determine \( \hat{t}_2 \) in (4), whereas the information of the first vector (sun vector) is totally used in (3).

**Example 1 – The TRIAD method:** Given two reference vectors in the inertial frame \( N \vec{v}_1 \) and \( N \vec{v}_2 \) (e.g. position of the sun and position of a certain star measured from the Earth) and the true attitude (unknown in reality) \( \theta_{\text{true}} \), from which the observation vector in the body frame shall be created, estimate the body attitude.

\[
N \vec{v}_1 = [1 \ 0 \ 0]^T \\
N \vec{v}_2 = [0 \ 0 \ 1]^T \\
\theta_{\text{true}} = [30^\circ \ 20^\circ \ -10^\circ ]
\]

The solution is obtained following a simple sequence of steps as indicated below:

1) In order to generate the observation vectors, a true value for the attitude \( \theta_{\text{true}} \) is assumed and the corresponding DCM, \( C_{BNtrue} \), is found.

2) The reference vectors shall be transported to the body frame to get \( \beta \vec{v}_1 \) and \( \beta \vec{v}_2 \). This is how the reference vectors should be expected to be measured in the body frame.

3) Those values shall now be corrupted and normalized to simulate sensor reading errors and normalize the vectors to get \( \beta \vec{v}_1 \) and \( \beta \vec{v}_2 \).

4) Using (3), (4) and (5) with \( \beta \vec{v}_1 \) and \( \beta \vec{v}_2 \), the TRIAD seen from the body frame, composed by \( \beta \hat{t}_1 \), \( \beta \hat{t}_2 \) and \( \beta \hat{t}_3 \), is created.

5) Using (3), (4) and (5) with \( N \vec{v}_1 \) and \( N \vec{v}_2 \), the TRIAD seen from the inertial frame, composed by \( N \hat{t}_1 \), \( N \hat{t}_2 \) and \( N \hat{t}_3 \), is created.

6) The DCM that transforms coordinates in the TRIAD to the body frame, \( C_{BT} \), and from the TRIAD to the inertial frame, \( C_{NT} \), are built using (6) and (7), respectively.

7) The final attitude DCM is calculated using (8).

8) To check the accuracy, the rotation between the body frame \( B \) and the estimated body frame \( \hat{B} \) shall be calculated by (9):

\[ C_{BB} = C_{BN}C_{BNtrue} \]  

Notice that \( C_{BB} \) shall be close to identity, which would be the case if the sensor readings were perfect.

9) For a better evaluation of the accuracy, \( C_{BB} \) should be transformed in a Principal Rotation Vector parameter and the deviation will be its rotation \( \phi \).

For the numerical values of the above-mentioned steps, please refer to Appendix 1. It is interesting to highlight the obtained value for the deviation \( \phi = 1.8525^\circ \), to be compared later with those obtained with the other methods.

b) Wahba’s problem

If we want to improve the TRIAD method using several vectors, each DCM relating the body attitude to the inertial frame \( C_{BN} \) obtained from each pair of vectors would look slightly different due to sensor imprecisions.

Mathematician Grace Wahba stated the problem on how to deal with multiple observations in a way to minimize the error created by sensors uncertainties. The assumptions made are that the number of measurements \( M\geq1 \) (sun, magnetic field, stars, etc.); the mapping of those vectors in the inertial reference frame are known and that the mapping from the inertial frame to the body frame is described by (10):

\[ \beta \vec{\nu}_k = C_{BN} N \vec{\nu}_k \quad k = 1, ..., M \]  

where \( C_{BN} \) is the DCM being searched.

The problem is stated as finding the best DCM that reduces the difference between the vector observed in the body frame and its corresponding inertial frame representation (reference) transformed to the body frame, as stated in (11):

\[ J(C_{BN}) = \frac{1}{2} \sum_{k=1}^{M} w_k |\beta \vec{\nu}_k - C_{BN} N \vec{\nu}_k|^2 \]  

where \( w_k \) is the weight (importance) of the \( k \)-th vector.

![Figure 2 - Least squares curve fitting applied in 2D](image)

In the perfect world, \( \beta \vec{\nu}_k \) and \( C_{BN} N \vec{\nu}_k \) would be equal for any pair of vectors and \( C_{BN} \) would be the same no matter what vectors were used to compute it. Due to the uncertainties in the real world, \( J(C_{BN}) \) has a positive value, which needs to be minimized. Note that the cost function \( J \) can be compared to the error in a least squares curve fitting problem, as shown in...
Figure 2, but with a 3×3 matrix because of the three-dimensionality of the problem.

There are several methods found in the literature to solve Wahba’s problem. Three of them will be presented here as they appear in [3] and [2]: Devenport’s q-method; quaternion estimator – QUEST and optimal linear attitude estimator – OLAE.

c) Devenport’s q-method:

Equation (12) shows how the cost function \( J \) is re-written in Devenport’s q-method:

\[
J(C_{BH}) = \frac{1}{2}\sum_{k=1}^{M} w_k \left( b\hat{q}_k - C_{BH} N\hat{q}_k \right)^T \left( b\hat{q}_k - C_{BH} N\hat{q}_k \right)
\]

\[
= \frac{1}{2}\sum_{k=1}^{M} w_k \left( b\hat{q}_k^T b\hat{q}_k + N\hat{q}_k^T N\hat{q}_k - 2 b\hat{q}_k^T C_{BH} N\hat{q}_k \right)
\]

\[
= \sum_{k=1}^{M} w_k \left( 1 - b\hat{q}_k^T C_{BH} N\hat{q}_k \right)
\]

(12)

where \( w_k \) is the weight (importance) of the \( k \)-th vector.

Note that \( b\hat{q}_k \) and \( N\hat{q}_k \) are unit vectors, thus \( b\hat{q}_k^T b\hat{q}_k = N\hat{q}_k^T N\hat{q}_k = 1 \) and that \( C_{BH} \) is an orthogonal matrix, thus \( C_{BH}^T C_{BH} = I \). The problem of minimizing the cost function \( J(C_{BH}) \) is transformed in that of maximizing the gain function given by (13):

\[
g(C_{BH}) = \sum_{k=1}^{M} w_k b\hat{q}_k^T N\hat{q}_k
\]

(13)

Introducing quaternion as \( \bar{q} = [q_1 q_2 q_3]^T \), with vector part \( q = [q_1 q_2 q_3]^T \) and scalar part \( q_4 \), the matrix \( C_{BH} \) may be written in terms of quaternions as

\[
C_{BH} = (q_4^2 - q^T q) I_{3×3} + 2 qq^T - 2q_4[q \times]
\]

(14)

where \([q \times] \) is the skew-symmetric matrix of the quaternion’s vector part.

Devenport proves [3] that the gain function can be written according to (15):

\[
g(\bar{q}) = \bar{q}^T K\bar{q}
\]

(15)

where

\[
K = \begin{bmatrix}
S - \sigma I_{3×3} & Z \\
Z^T & \sigma
\end{bmatrix}
\]

(15.1)

\[
B = \sum_{k=1}^{M} w_k b\hat{q}_k^T N\hat{q}_k
\]

(15.2)

\[
S = B + B^T
\]

(15.3)

\[
\sigma = \text{tr}(B)
\]

(15.4)

\[
Z = [B_{23} - B_{32} B_{31} - B_{13} B_{12} - B_{21}]^T
\]

(15.5)

As quaternions must abide the unit length constraint, \(|\bar{q}| = 1\), it is not possible to maximize the gain function directly. For that reason, the Lagrange multiplier is introduced, and the new gain function is defined as shown in (16)

\[
g'(\bar{q}) = \bar{q}^T K\bar{q} - \lambda(\bar{q}^T \bar{q} - 1)
\]

(16)

Note that \( g'(\bar{q}) = g(\bar{q}) \) when the constraint \(|\bar{q}| = 1\) is fulfilled. Differentiating \( g' \) with respect to \( \bar{q} \) and equalling it to zero to find the maximum point, results in (17):

\[
\frac{d}{d\bar{q}} (g'(\bar{q})) = 2K\bar{q} - 2\lambda \bar{q} = 0 \Rightarrow K\bar{q} = \lambda \bar{q}
\]

(17)

In this way, the maximization problem has been reduced to an eigenvalue problem and the maximum eigenvalue will maximize the gain function \( g \), as shown in (18):

\[
g(\bar{q}) = \bar{q}^T K\bar{q} = \bar{q}^T \lambda \bar{q} = \lambda \bar{q}^T \bar{q} = \lambda
\]

(18)

The answer to the problem is the eigenvector associated with the maximum eigenvalue.

Summarizing Devenport’s method: the 4×4 matrix \( K \) is computed according to (15.1) through (15.5); the eigenvalues and eigenvectors of matrix \( K \) are found; the largest eigenvalue and associated eigenvector are chosen. This eigenvector is the Euler parameter vector that maximizes the gain function \( g(\bar{q}) \).

The disadvantage of this method if the necessity of calculating the eigenvectors and eigenvalues of a four by four matrix, which may be demanding in terms of computational resources [2].

Example 2 - Devenport’s method: Same situation as in the previous example with two reference vectors, though more vectors could be used.

The general solution follows a set of steps similar to Example 1:

1) 2) and 3) are the same as in the previous example.

4) Set the q-method parameters: establish the weights \( w_1 \) and \( w_2 \), calculate the matrices \( B \) and \( S \) as per (15.2) and (15.3), respectively, the scalar \( \sigma \) (15.4) and the vector \( Z \) (15.5).

5) Assemble matrix \( K \) with q-method parameters according (15.1).

6) Compute matrix \( K \)’s eigenvalues \( \lambda \).

7) Attitude quaternion \( \bar{q}_{BH} \) is the eigen-vector associated with biggest eigen-value.

To check accuracy, follow steps 8) and 9) of previous example.

The results obtained following the above-mentioned steps are presented in Appendix 2. The calculated deviation was \( \varphi = 1.6960^\circ \), slightly better than the one obtained with the TRIAD method.

d) QUEST method

Similar to Devenport’s method, the quaternion estimator method solves Wahba’s problem by finding the eigenvalues of the \( K \) matrix. However, QUEST uses Newton-Raphson iteration method from an initial estimated point, which requires less computations.

Starting from Wahba’s problem as stated in Devenport’s method by equations (12) and (13) and including Devenport’s method conclusion expressed in (18), the cost function \( J \) may be written as in (19):

\[
J = \sum_{k=1}^{M} w_k - g = \sum_{k=1}^{M} w_k - \lambda_{opt}
\]

(19)

As we are assuming small errors and thus, \( J \approx 0 \), the approximate value of \( \lambda_{opt} \) to be used as a first guess becomes, as expressed in (20):
\[ \lambda_{\text{opt}} \approx \sum_{k=1}^{M} w_k \]  

(20)

Therefore, the starting point for the Newton-Raphson iteration process will be the sum of the weights.

By definition, the eigenvalues of \( K \) must satisfy the characteristic equation (21):

\[ f(s) = \det(K - sI_{4 \times 4}) = 0 \]  

(21)

As \( K \) is a \( 4 \times 4 \) matrix, \( f(s) \) will be a 4\textsuperscript{th} degree polynomial. MATLAB function \texttt{poly(K)} returns the coefficients of that polynomial.

The Newton-Raphson iteration method is summarized in equations (22.1), (22.2) and (22.3):

\[
\begin{align*}
\lambda_0 &= \sum_{k=1}^{M} w_k \\
\lambda_1 &= \lambda_0 - \frac{f(\lambda_0)}{f'(\lambda_0)} \\
\lambda_{\text{opt}} &= \lambda_{\text{max}} = \lambda_1 = \lambda_{i-1} - \frac{f(\lambda_{i-1})}{f'(\lambda_{i-1})}
\end{align*}
\]

(22)

(22.1)

(22.2)

(22.3)

Once the maximum eigenvalue is found, the associated eigenvector is found using Classic Rodrigues Parameters described in (23):

\[
\mathbf{q} = \begin{bmatrix} p \end{bmatrix}_{(1)}
\]

(23)

The eigenvalue problem can be expressed as (24) by substituting (23) into (17):

\[
K \begin{bmatrix} p \end{bmatrix}_{(1)} = \lambda_{\text{opt}} \begin{bmatrix} p \end{bmatrix}_{(1)}
\]

\[
\Rightarrow \begin{bmatrix} S - \sigma I_{3 \times 3} & Z \\ Z^T & \sigma \end{bmatrix} \begin{bmatrix} p \end{bmatrix}_{(1)} = \lambda_{\text{opt}} \begin{bmatrix} p \end{bmatrix}_{(1)}
\]

\[
(S - \sigma I_{3 \times 3})p + Z = \lambda_{\text{opt}} p
\]

\[
p = \left( (\lambda_{\text{opt}} + \sigma)I_{3 \times 3} - S \right)^{-1} Z
\]

(24)

and the corresponding quaternion is calculated by (25):

\[
\mathbf{q} = \frac{1}{\sqrt{1 + p^T p}} \begin{bmatrix} p \end{bmatrix}_{(1)}
\]

(25)

Thus, the optimal eigenvalue is found, the quaternion as described above gives the object’s attitude. There is an inversion of a \( 3 \times 3 \) matrix, but that is a faster process in comparison to solving the characteristic equation.

**Example 3 – QUEST method:** Same situation as in the two previous examples with two reference vectors.

The general solution follows a set of steps similarly to Examples 1 and 2:

1) through 5) are the same as in the previous example.

6) Find the characteristic polynomial \( f(s) \) as per (21) and its corresponding first derivative \( f'(s) \).

7) Proceed with Newton-Raphson iteration (22.1) through (22.3) to find \( \lambda_{\text{opt}} \).

8) Use (24) and (25) to find the attitude quaternion.

9) To check accuracy, follow steps 8) and 9) of Example 1.

The solution using the QUEST method must give the same result as when using Devenport’s q-method as both are solving the same eigenvalue problem. Considering just the first approach as in (20), the obtained deviation is \( \varphi = 1.7010 \), which is already very close to the value obtained with Devenport’s and better than the one obtained with the TRIAD method. When applying Newton-Raphson iteration process the deviation becomes the same as in Devenport’s.

**e) OLAE Method**

The Optimal Linear Attitude Estimator – OLAE method has a different approach than the previous ones. It does not solve Wahba’s problem as Devenport’s and QUEST methods, but it formulates the estimation problem in a different way.

The starting point is the Cayley transform that obtains the DCM from a classic Rodrigues parameter, as stated by (26):

\[
\mathbf{C}_{\hat{n}} = (I_{3 \times 3} + [\mathbf{p} \times])^{-1}(I_{3 \times 3} - [\mathbf{p} \times])
\]

(26)

Where \( \mathbf{p} \) is the CRP vector and \([\mathbf{p} \times]\) its skew-symmetric matrix.

Multiplying both sides by \( n \hat{\mathbf{q}} \) and using (10), we get the relation expressed in (27):

\[
(I_{3 \times 3} + [\mathbf{p} \times])n \hat{\mathbf{q}} = (I_{3 \times 3} - [\mathbf{p} \times])n \hat{\mathbf{q}} = \mathbf{b} \hat{\mathbf{q}} - n \hat{\mathbf{q}}
\]

(27)

Defining the sum vector \( \mathbf{s}_i \) (28) and the difference vector \( \mathbf{d}_i \) (29) as:

\[
\mathbf{s}_i = \mathbf{b} \hat{\mathbf{q}} + n \hat{\mathbf{q}}
\]

(28)

\[
\mathbf{d}_i = \mathbf{b} \hat{\mathbf{q}} - n \hat{\mathbf{q}}
\]

(29)

Then inserting (28) and (29) into (27), we get the following expression (30) for the difference vector:

\[
\mathbf{d}_i = -[\mathbf{p} \times] \mathbf{s}_i = [\mathbf{s}_i \times] \mathbf{p}
\]

(30)

Defining the complete difference vector \( \mathbf{d} \) as (31.1), and the sum and weight matrices \( \mathbf{S} \) and \( \mathbf{W} \), respectively, as (31.2) and (31.3):

\[
\mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_M \end{bmatrix}_{3M \times 1}
\]

(31.1)

\[
\mathbf{S} = \begin{bmatrix} [\mathbf{s}_1 \times] \\ [\mathbf{s}_2 \times] \\ \vdots \\ [\mathbf{s}_M \times] \end{bmatrix}_{3M \times 3}
\]

(31.2)

\[
\mathbf{W} = \begin{bmatrix} [\mathbf{w}_1 I_{3 \times 3}]_{3M \times 3} \\ \vdots \\ [\mathbf{w}_M I_{3 \times 3}]_{3M \times 3} \end{bmatrix}
\]

(31.3)

Then the classic Rodrigues parameter that defines the body attitude relative to the inertial frame can be written as (31.4):

\[
\mathbf{p} = (\mathbf{S}^T \mathbf{W}^{-1} \mathbf{S})^{-1} \mathbf{S}^T \mathbf{W} \mathbf{d}
\]

(31.4)

**Example 4 – OLAE method:** Same situation as in the three previous examples with two reference vectors.

The general solution follows a set of steps similarly to previous examples:

1) through 3) are the same as in the previous example.
4) Find the difference vector \( \mathbf{d} \) according to (29) and (31.1), being \( \mathbf{B}_i \) the measured vector in the body frame and \( \mathbf{N}_i \) its correspondent reference vector in the inertial frame.

5) Find the sum vector according to (28) and assemble the sum matrix as per (31.2).

6) Assemble the weight matrix according to (31.3).

7) Calculate the estimated attitude as a CRP from (31.4).

To check accuracy, follow steps 8) and 9) of Example 1.

The obtained results following the above-mentioned steps are presented in Appendix 3. The calculated deviation was \( \varphi = 1.6872^\circ \), slightly better than those obtained with the other three methods.

**Summary**

A summary of the main characteristics of the static attitude determination methods discussed in this work is presented below:

**TRIAD method:**
- Uses only two observation vectors.
- Simple to use and to understand.
- Uses all information from the first vector, but not all from the second vector.

**Wahba’s problem:**
- Allows multiple observations.
- States the attitude determination problem as a least squares optimization problem.
- Compares the observed (measured) vector in the body fixed frame with the known vector in the inertial frame.
- Seeks the optimal DCM that will reduce the overall error between measured and expected attitude.

**Devenport’s method:**
- Solves Wahba’s problem.
- Uses Euler parameters (quaternions).
- Reduces the optimization problem to an eigenvalue problem.
- Enhances precision in relation to the TRIAD method.
- More complicated and demands more computational capacity (solves eigenvalues of a 4×4 matrix).

**QUEST method:**
- Also solves Wahba’s problem.
- Does not need to solve the eigenvalue problem.
- Uses an iteration process around an estimated point.
- Enhanced speed in relation to Devenport’s method.
- Same precision as Devenport’s method.

**OLAE Method:**
- Does not solve Wahba’s problem.
- Formulates the estimation problem in a different way, by manipulating the Cayley transform.
- Uses Classic Rodrigues Parameters.

**C. Dynamic attitude estimation methods**

When we consider attitude determination of a dynamic system, mainly two factors are being added to the problem: measurements are being taken over time as attitude is now time dependent and measurements contain statistical noise and other inaccuracies and uncertainties. Because of the statistical nature of the problem, the term “attitude estimation” is preferred over “attitude determination”. Statistics, in general, and Estimation, in particular, are therefore mandatory pre-requisites for studying dynamic attitude estimation.

**a) Kalman Filter (KF):**

Kalman filtering (KF) is a very popular tool for treating measurement data and making good estimations out of them and thus it will be the focus in this section. However, as it is part of Space dynamics course, rather than Control systems or Mathematics course, the goal should be to present the subject from a user point of view instead of from a developer’s one.

A quick review of basic statistics concepts in the beginning of a course on this subject would be of great use, specially including mean value, variance, covariance and expected value. Those concepts are frequently used while explaining KF concepts and modelling. Reference [5] has a good review of all concepts used in KF and in [7] a review of covariance is presented in a simple and accessible way.

The first questions to be answered are: what is KF, what it does and why we need it? Before having any experience with sensors, students tend to believe that sensor data are equivalent to reality rather than being an imperfect representation of it that should be treated in such a way to get information that is reliable and as close to reality as possible. A sensor reading shall not be considered as a deterministic entity, a point in a diagram, but as a statistic quantity better...
represent by a cloud of probabilistic distribution, as represented in Figure 3.

Thus, KF could be defined as an iterative algorithm, which assumes that the measurement consists of a true value corrupted by a noise or error and in a mathematical way it makes an educated guess of which part of the observation that is noise and which that is the closest we can get to the true value (that is why the term “filter” is used). For doing so, the algorithm needs more information that comes partly from the knowledge on the process, in the form of the equations of motions for example, and partly from other observations/measurements.

So, answering the proposed questions: KF is an iterative algorithm that estimates what is noise and what is usable information. KF could be used for getting better estimates out of noisy measurements.

References [8] and [14] present the concepts and proceeding of KF in simple and accessible way and could be used as support material.

Other important characteristics of KF is that it is a discrete process, data is sampled at intervals of time $\Delta t$, and it is recursive process, it only needs information from the actual and previous states. The process is divided into a prediction phase, where a preliminary estimation is made, and an update phase, where observation results are incorporated.

When trying to understand KF, one challenge a student may face is the notation. There are several elements in the algorithm, which may be confusing in the beginning, and they should be presented gradually as they appear in the equations. A helping tool could be having always available a table with the used notation and respective description, as in Table 1.

<table>
<thead>
<tr>
<th>$x$</th>
<th>State vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>Estimate error covariance matrix</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Transition matrix</td>
</tr>
<tr>
<td>$Q$</td>
<td>Process covariance matrix</td>
</tr>
<tr>
<td>$R$</td>
<td>Measurement covariance matrix</td>
</tr>
<tr>
<td>$K$</td>
<td>Kalman gain matrix</td>
</tr>
<tr>
<td>$z$</td>
<td>Measurement or observation vector</td>
</tr>
<tr>
<td>$H$</td>
<td>Measurement transition matrix</td>
</tr>
<tr>
<td>$I$</td>
<td>Identity matrix</td>
</tr>
<tr>
<td>$k$</td>
<td>Subscript $k$: k-th time step</td>
</tr>
<tr>
<td>$\hat{x}$</td>
<td>Hat indicates that the variable is an estimator, e.g. $\hat{x}$</td>
</tr>
<tr>
<td>$\hat{x}^-$</td>
<td>Superscript $-$ indicates best estimate before considering measurements, e.g. $\hat{x}^-$</td>
</tr>
</tbody>
</table>

One important assumption used in KF is that the process and measurement uncertainties are white noises, which means they are Gaussian distributions with known covariances and mean values equal to zero. Another important assumption is that the process, transition or system dynamic model is linear and has the form as (32):

$$x_{k+1} = \phi_k x_k + w_k$$

Where $x_k$ is the state vector that contains the variables that are being studied (position, velocity, temperature, etc.); $\phi_k$ is the linear transition matrix that modifies the state vector from one time step to next and $w_k$ is the transition model noise vector, that represents the uncertainty in the transition model. It has a zero mean value and $Q_k$ as covariance matrix, as described in (33).

$$E[w_k] = 0; \quad E[w_k w_i^T] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases}$$

The measurement or observation model is also assumed to be linear and can be described as in (34):

$$z_k = H_k x_k + v_k$$

where $z_k$ is the observation (measurements) vector, where sensor data is presented; $H_k$ is the linear observation matrix that establishes the relationship between the state vector and the observation made and $v_k$ is the observation model noise vector that represents the uncertainty in the measurement model and it has a zero mean value and $R_k$ as covariance matrix, as shown in (35):

$$E[v_k] = 0; \quad E[v_k v_i^T] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases}$$

Covariances $Q_k$ and $R_k$ are assumed to be known and noises $w_k$ and $v_k$ are independent in relation to each other, which is represented in (36):

$$E[w_k v_i^T] = 0 \quad \text{for all } k \text{ and } i$$

The outcome of the KF algorithm is an estimate of the state vector $\hat{x}_k$ for each time step $k$ and its pertaining probability distribution expressed in terms of the estimate covariance matrix $P_k$. Those values could be represented as the cloud in Figure 3, if the elements in the state vector are uncorrelated.

References 

Table 1 - Kalman filter notation

Figure 4 - Statistic representation of a state consisting of correlated position and velocity [8].
(when \( P_k \) being a diagonal matrix), or as the cloud in Figure 4 if they are correlated.

The estimate covariance matrix or estimate error covariance matrix \( P_k \) represents how trustworthy the estimate is. If we define the estimate error as being the difference between the true value and the estimate value of the state vector, as in (37), the estimate error covariance can be described by (38):

\[
e_k = x_k - \hat{x}_k
\]
\[
P_k = E[e_k e_k^T]
\]  

As mentioned before, the Kalman loop consists of two phases: the propagation or prediction phase, where the process equations are used to predict the state in the next time step, and the update phase where measurement data are used to trim the estimate. As it is an iterative process, the order of those two phases is not important and as it is a recursive process, only the information from the previous time step is needed. For this reason, it is necessary to provide the best estimation for the initial state \( \hat{x}_0 \) and its associated error covariance matrix \( P_0 \). If they are totally unknown, any value may be assumed, as the process will normally converge after some iterations, as it will be shown in the examples.

The steps in the Kalman loop with respective equation are listed below.

0. Best initial estimate:
   \( \hat{x}_0; P_0 \).

1. Compute the Kalman gain:
   \( K_k = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \) (39.1)

2. Update estimate with the measurement \( z_k \):
   \( \hat{x}_k = \hat{x}_k + K_k(z_k - H_k \hat{x}_k) \) (39.2)

3. Update the error covariance estimate:
   \( P_k = (I - K_k H_k) P_k \) (39.3)

4. Project ahead the estimate:
   \( \hat{x}_{k+1} = \Phi_k \hat{x}_k \) (39.4)

5. Project ahead the error covariance:
   \( P_{k+1} = \Phi_k P_k \Phi_k^T + Q_k \) (39.5)

6. Return to step 1.

We will know take a closer look at each step, its components and what they represent.

Steps one to three are part of the update phase. In step one, the Kalman gain \( K_k \) is calculated by (39.1). Basically, it is a comparison of the estimate uncertainty with the measurement uncertainty. It is a set of figures between zero and one. The closer the Kalman gain gets to one, the smaller is the influence of the measurement covariance \( R_k \), meaning that the measurement is more reliable than the prediction by the process knowledge. On the other side, if the Kalman gain is close to zero, the estimate error covariance \( P_k \) is close to zero, meaning that we have more confidence in the process model.

In the second step, the estimate is updated with the information from the measurement, as expressed in (39.2). The difference between the measurement \( z_k \) and what was expected to be the measurement considering the estimate \( H_k \hat{x}_k \) is calculated. This difference is weighted by the Kalman gain to be added to the pre-update estimate. Again,

\[
H_k \hat{x}_k
\]  

we see that if the Kalman gain \( K_k \) value is high, meaning a low confidence in the process model relatively to the observation, more weight will be put to the measurement, whereas if \( K_k \) has a lower value, indicating higher confidence in the process model relatively to the observation, less weight will be put to the measurement.

The estimate error covariance is also updated in step three and we can see from (39.3) that if the Kalman gain is high (more reliable measurement), the updated error covariance decreases more significantly, resulting in a more accurate estimate than if \( K_k \) is low (more confidence in the process model than in the measurement). In the latter case, the measurement accuracy has a smaller influence in decreasing \( P_k \) and getting more accurate estimates.

Steps four and five are the propagation phase, where the updated estimate and error covariance will be sent to the next time as pre-updated values. The propagation of the estimate in step four is ruled by the process model, as shown in (39.4).

\[ \Phi_k \]

Figure 5 - Intersection between expected observation \( H_k \hat{x}_k \) (estimate 1) and observation \( z_k \) (estimate 2) [8].

Figure 6 - Intersection of two Gaussian distributions [8].
The propagation of the error covariance (39.5) uses the same transition matrix \( \Phi_k \) as for the estimate plus a measure of the confidence in the process in the form of process covariance \( Q_k \).

Figure 5 gives an illustration of what is being done. The green dot and cloud represent a measurement point \( z_k \) and its covariance \( R_k \), respectively. This cloud should be seen as a Gaussian distribution of the probable location of the true value, given the fact that we observed \( z_k \). The same is valid for the pink cloud, that represents the prediction of measurement considering the estimate. The covariance of the pink cloud is \( H_k \Sigma_k H_k^T \). Thus, there are two estimate clouds for the same point and the true value must then be in the intersection of them both, which represents a smaller uncertainty.

Another illustration that helps understanding the basic idea behind KF is shown in Figure 6. The intersection of two Gaussian distributions with different dispersions, gives another Gaussian distribution with intermediate dispersion. Given two normal distributions \( \mathcal{N}(x, \mu_0, \sigma_0) \) and \( \mathcal{N}(x, \mu_1, \sigma_1) \) the intersection of them will have following mean value (40) and variance (41):

\[
\begin{align*}
\mu' &= \mu_0 + k(\mu_1 - \mu_0) \\
\sigma'^2 &= \sigma_0^2 - k \sigma_0^2 + k \sigma_1^2.
\end{align*}
\]

Comparing equations (42), (43) and (44) with (39.1), (39.2) and (39.3) it is possible to understand that KF is a multiplication of Gaussian distributions.

**Example 5 - One-dimensional linear Kalman Filter:**

This example is the simplest problem possible to be solved with Kalman filter and is thought to be used as an introduction to the subject. It is simple enough to be solved using a white board during a lecture.

Consider a sensor measuring a fluid with constant temperature [16]. From a given series of temperature measurement data \( z_k \) on the second column of Table 2, estimate the temperature (suppose the true temperature is \( 72\degree C \)).

The solution is given in a stepwise description as in the previous examples.

1) Assume an initial estimate \( \hat{x}_0 = 68\degree C \), initial estimate error \( P_0 = 2^2 \degree C^2 \) and the measurement covariance \( R_k = 4^2 \degree C^2 \) (considered constant over time).

2) Write the transition and the measurement models, as (45) and (46), respectively:

\[
x_{k+1} = x_k + v_k
\]

\[
z_k = x_k + v_k
\]

Note that the transition and measurement models’ linear coefficients are: \( \Phi_k = H_k = 1 \) and it is considered here that the process error and covariance are \( w_k = Q_k = 0 \) (perfect model). Consequently, the pre-update value of the estimate and estimate error at step \( k \) are equal to respective post-update values of the previous step: \( \hat{x}_k = \hat{x}_{k-1} \) and \( P_k = P_{k-1} \).

3) Calculate the Kalman gain of the first step using (47), which was derived from (39.1):

\[
K_k = P_{k-1}/(P_{k-1} + R_k)
\]

4) Update the estimate with the Kalman gain and the measurement using (48), which was derived from (39.2):

\[
\hat{x}_k = \hat{x}_{k-1} + K_k(z_k - \hat{x}_{k-1})
\]

5) Update the estimate covariance using (49), which was derived from (39.3):

\[
P_k = (1 - K_k)P_{k-1}
\]

Table 2 shows the results for the first four steps, that could even be calculated by hand on a white-board tutorial. It is important to notice that although the first estimate \( \hat{x}_0 \) is rather far from the true value, after some few iterations, it will be quite close. Figure 7 shows the evolution of the estimate considering 100 iterations. It is also interesting to notice that the Kalman gain decreases for each iteration showing that the contribution of the measurements in updating the estimate becomes less and less important as the estimate tends to the true value.

**Example 6 - Multi-dimensional Kalman Filter:**

In this example the level of complexity will be raised to include a second dimension in the state vector. The problem considers an object in freefall, no air resistance, uncertain
information about initial position and uncertain position measurement from a laser rangefinder. The task is to estimate the position of the object during a time period.

1) The process equations, which in this case are the equations of motion are described in (50):

\[ x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} \] (50.1)

where

- state vector: \( x_k = [\hat{h}_k \ \dot{\hat{h}}_k] \) (50.2)
- transition matrix: \( F_{k-1} = F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \) (50.3)
- input matrix: \( G_{k-1} = G = \begin{bmatrix} -\frac{1}{2}g(\Delta t)^2 \\ -\Delta t \end{bmatrix} \) (50.4)
- input vector: \( u_{k-1} = u = g \) (50.5)

It is assumed no uncertainties in the equation of motion, \( w_k = 0 \), and thus, the process noise covariance matrix becomes:

\[ Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \]

2) Measurement equation is described in (51):

\[ z_k = H_kx_k + v_k \] (51)

where:

- \( z_k \): Observation vector;
- \( H_k = [1 \ 0] \): linear observation matrix as only \( \hat{h}_k \) is being measured;
- \( v_k \): observation model noise vector.

Assumptions:

- Measurement system variance: \( \sigma^2 = 4 \text{ m}^2 \);
- Only one term in the output vector, measurement noise covariance matrix becomes a scalar: \( R = 4 \text{ m}^2 \);
- Initial true position: \( h_0 = 100 \text{ m} \);
- Initial estimated position: \( \hat{h}_0 = 105 \text{ m} \);
- Initial estimated position variance: \( \sigma^2_{\hat{h}_0} = 10 \text{ m}^2 \) (rough estimation);
- Initial true velocity: \( \dot{h}_0 = 0 \text{ m/s} \);
- Initial estimated velocity: \( \dot{\hat{h}}_0 = 0 \text{ m/s} \);
- Initial estimated velocity variance: \( \sigma^2_{\dot{\hat{h}}_0} = 0.01 \text{ m}^2/\text{s}^2 \) (smaller uncertainty);
- True initial state vector: \( x_0 = \begin{bmatrix} 100 \\ 0 \end{bmatrix} \);
- Estimate initial state vector: \( \hat{x}_0 = \begin{bmatrix} 105 \\ 0 \end{bmatrix} \);
- Assumed initial state error covariance matrix:

\[ P_0 = \begin{bmatrix} 10 & 0 \\ 0 & 0.01 \end{bmatrix} \]

Following (39), which was done by a MATLAB code, the estimated position is presented in two situations. Figure 8 shows the results considering a perfect process model, where the process noise covariance matrix is zero. The estimate rapidly approaches the true position value after a few iterations. If a small process uncertainty is inserted in matrix \( Q \)'s diagonal (here 0.1 m\(^2\) and 0.1 m\(^2\)/s\(^2\)), the estimate varies around the true position, but with a smaller variance than that of the measured values, as shown in Figure 9.

2b) Linearized Kalman Filter (LKF) and Extended Kalman Filter (EKF):

The Kalman filter studied in previous section uses as a basic assumption that both transition and observation models are linear. That assumption is needed as KF works with Gaussian distributions and a linear model guarantees that the results will also have a Gaussian distribution as illustrated in Figure 10. However, this is not the case for non-linear transition functions and the KF procedures as seen so far cannot be used.

The solution is to locally linearize the function using a first order Taylor series expansion. A first approach is to use the so called Linearized Kalman Filter (LKF), where the linearization is carried out about some nominal trajectory in state space that does not depend on the measurement data, as shown in Figure 11.

LKF has only limited application, as the nominal trajectory must be previously known. Extended Kalman Filter (EKF) is a more useful tool because it does not have that requirement, as will be shown later. However, LKF may be an interesting educational step in preparation for understanding EKF.
The state variables to be used now will be the incremental quantities, instead of total quantities and the process will be described by a set of differential equations. Process and measurement models can be described by equations (52) and (53), respectively:

\[
\dot{x} = f(x, u_d, t) + u(t) \quad (52)
\]

\[
z = h(x, t) + v(t) \quad (53)
\]

where, \( f \) and \( h \) are known nonlinear, differentiable functions; \( u_d \) is a deterministic forcing function and \( u \) and \( v \) are white noise processes with zero cross-correlation.

Assuming that an approximate trajectory \( x^*(t) \) may be determined by some means, it will be referred to as the nominal or reference trajectory [5]. The actual trajectory may then be described as in (54):

\[
x(t) = x^*(t) + \Delta x(t) \quad (54)
\]

Substituting (54) in the process and measurement models (52) and (53), respectively, and making a Taylor series expansion, we get the following linearized dynamic model (55) and linearized measurement equation (56) (see appendix 4 for details):

\[
\Delta \dot{x} = F \Delta x + u(t) \quad (55)
\]

\[
z - h(x^*, t) = H \Delta x + v(t) \quad (56)
\]

where \( F \) and \( H \) are the Jacobian of \( f \) and \( h \), respectively, applied along the reference trajectory:

\[
F = \left[ \frac{\partial f}{\partial x} \right]_{x=x^*} \quad (57.1)
\]

\[
H = \left[ \frac{\partial h}{\partial x} \right]_{x=x^*} \quad (57.2)
\]

It is important to remember that the state vector is now \( \Delta x \), instead of \( x \) as it was in KF and that the “measurement” in the measurement model is the actual measurement less that predicted by the nominal trajectory in the absence of noise \( z - h(x^*, t) \).

The incremental estimate update equation in the Kalman loop becomes (58):

\[
\Delta \hat{x}_k = \Delta \hat{x}_{k|k} + K_k(z_k - \hat{z}_k) \quad (58.1)
\]

where

\[
\hat{z}_k = h(x^*_k) + H_k \Delta \hat{x}_k \quad (58.2)
\]

Taking into account those considerations, the Kalman loop is applied in the same way as previously for KF.

The EKF is similar to LKF, but now, as there is no pre-determined reference trajectory, linearization takes place about an estimated trajectory that is being continually updated with the state estimates resulting from the measurements, as illustrated in Figure 12.

One disadvantage of EKF is that there is a chance that the updated trajectory will be poorer than the nominal one, leading to eventual divergence of the filter. That makes EKF less robust compared to LKF, especially when the initial uncertainty and measurement errors are large.

In contrast to LKF, EKF uses total instead of incremental quantities in the state variable. If we add \( x^*_k \) at both sides of (58.1), we get the total estimate update equation (59):

\[
x_k + \Delta \hat{x}_k = x^*_k + \Delta \hat{x}_{k|k} + K_k(z_k - \hat{z}_k) \Rightarrow
\]
\begin{align*}
\Rightarrow \hat{x}_k &= \hat{x}_k^- + K_k(z_k - \hat{z}_k^-) \\
\text{(59)}
\end{align*}

where \((z_k - \hat{z}_k^-)\) is the measurement residual and \(\hat{z}_k^-\) is given by (60), being calculated using \(h(\hat{x}_k^-)\) instead of \(h(x_k^-)\) in (58.2):

\begin{equation}
\hat{z}_k^- = h(\hat{x}_k^-) + H_k \hat{x}_k^- \\
\text{(60)}
\end{equation}

Once the update is made in the EFK, the incremental \(\Delta \hat{x}_k\) is reduced to zero. To project \(\hat{x}_k\) to the next time step \(\hat{x}_{k+1}\) we need to solve the set of nonlinear differential equations shown in (61) with initial condition \(x = \hat{x}_k\) at \(t_k\).

\begin{align*}
\hat{x}_{k+1} = \\
\text{Solution of the nonlinear differential equation} \\
\dot{x} = f(x, u_d, t) \text{ at } t = t_{k+1}, \text{ subject to the initial condition } x = \hat{x}_k \text{ at } t_k \\
\text{(61)}
\end{align*}

Estimate error covariance propagation occurs in the same fashion as for KF, according to equation (39.5). However, in order to guarantee symmetry and positive definiteness of the \(P\) matrix and avoid divergence, the form presented in (62) is preferred for the update phase:

\begin{equation}
P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \\
\text{(62)}
\end{equation}

The Kalman loop steps for EKF are summarized below.

1. **Best initial estimate:**
   \(\hat{x}_0^0; \quad P_0^0\).

2. **Compute the Kalman gain:**
   \[K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}\]
   \(\text{(63.1)}\)

3. **Update estimate with the measurement \(z_k\):**
   \[\hat{x}_k = \hat{x}_k^- + K_k (z_k - \hat{z}_k^-)\]
   \(\text{(63.2)}\)

4. **Update the error covariance estimate:**
   \[P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T\]
   \(\text{(63.3)}\)

5. **Project ahead the estimate:**
   \begin{align*}
   \hat{x}_{k+1}^- = \\
   \text{Solution of the nonlinear diff. eq.} \\
   \dot{x} = f(x, u_d, t) \text{ at } t = t_{k+1}, \text{ subject to the initial cond. } x = \hat{x}_k \text{ at } t_k \\
   \text{(63.4)}
   \end{align*}

6. **Project ahead the error covariance:**
   \[P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + Q_k\]
   \(\text{(63.5)}\)

7. **Return to step 1.**

**Example 7: Underdamped harmonic oscillator**

Consider a spring-mass-damper system with known damping coefficient \(\zeta\) and known natural frequency \(\omega\). The set of continuous-time differential equations that models this problem is given by (64):

\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\omega^2 x_1 - 2\zeta \omega x_2 + g + w \\
\text{(64.1)}
\end{align*}

and the measurement model will be described by (65):

\begin{equation}
z = x_1 + v \\
\text{(65)}
\end{equation}

being \(w\) and \(v\) white noise with covariances \(Q_c\) and \(R_c\), respectively. Equations (64) can be written in matrix form as (66):

\begin{equation}
x = Ax + B(g + w) \\
\text{(66.1)}
\end{equation}

where

\begin{equation}
A = \begin{bmatrix}
0 & 1 \\
-\omega^2 & -2\zeta \omega
\end{bmatrix} \\
\text{(66.2)}
\end{equation}

\begin{equation}
B = \begin{bmatrix}
0 \\
1
\end{bmatrix} \\
\text{(66.3)}
\end{equation}

Those equations need to be discretized to be used in a numerical environment. Reference [21] gives the discrete-time form for the state equation (67), assuming small time interval \(\Delta t\):

\begin{equation}
x_k = F x_{k-1} + G g + Aw \\
\text{(67.1)}
\end{equation}

where:

\begin{align*}
F &= I + A \Delta t \\
G &\approx B \Delta t \\
A &\approx I \Delta t \\
H &= C \\
\omega_k &\sim N(0, Q) \\
Q &= Q_c \Delta t \\
R &= R_c / \Delta t
\end{align*}

Figure 13 presents the estimates for position and velocity obtained by the EKF procedure, from a MATLAB simulation.

From the result it is possible to observe that, despite a rather broad dispersion in the measured points (large \(R_c\)), it is possible to obtain a quite reliable estimate, very close to the true value, if we are confident about our process model (low \(Q_c\)). It is interesting to change the different parameters and see the effect in the results. Even with larger values of \(Q_c\), the estimates are significantly closer to the true value than the measurement are. Smaller time steps give better adherence of the estimate curve in relation to the true value, because of the small time-step assumption in the discretization method.

One extension of this same problem is when the damping factor is assumed constant but unknown and integrates the three-dimensional state vector. A third equation shall be added to the system of differential equations in (64), as described in (68):

\begin{equation}
\dot{x}_3 = 0 \\
\text{(68)}
\end{equation}

and the new continuous-time matrices in (66) will assume the form of (69):

\begin{equation}
A = \begin{bmatrix}
0 & 1 & 0 \\
-\omega^2 & -2\zeta \omega & -\omega \xi_3 \\
0 & 0 & -\omega \xi_2
\end{bmatrix} \\
\text{(69.1)}
\end{equation}

\begin{equation}
B = \begin{bmatrix}
0 \\
1 \\
0
\end{bmatrix} \\
\text{(69.2)}
\end{equation}
The obtained result of the estimate damping factor is a constant value after some iterations, as shown in Figure 14. It is interesting to note the behaviour of damping factor estimate as it varies significantly in the beginning to rapidly converge to the true value. It is also interesting to point out that several values for \( Q_c \) and \( \Delta t \) were tested resulting in different conversion values for the damping factor, showing that this are the handles to adjust the EKF model.

Grewal & Andrews [24] provides several examples including MATLAB codes. However, these codes should be used with caution as sometimes they do not reflect exactly what they were intended to. This example was inspired by examples 5.7 and 8.4, but the codes needed to be rewritten to give consistent results. Reference [24] is however an interesting source for examples and theory reference.

Grewal & Andrews [24] provides several examples including MATLAB codes. However, these codes should be used with caution as sometimes they do not reflect exactly what they were intended to. This example was inspired by examples 5.7 and 8.4, but the codes needed to be rewritten to give consistent results. Reference [24] is however an interesting source for examples and theory reference.

\[
\hat{q} \equiv E(q) \quad \text{and} \quad \Delta q \equiv q - \hat{q}
\]

\[
\therefore E(|q|^2) \equiv E \left( |\hat{q} + \Delta q|^2 \right) = E \left( |\hat{q}|^2 + 2|\hat{q}|\Delta q + |\Delta q|^2 \right)
\]

\[
= |\hat{q}|^2 + 2|\hat{q}|E(|\Delta q|) + E(|\Delta q|^2) > |\hat{q}|^2 \tag{70}
\]

As \( \hat{q} \) has a unit norm, the result of its estimation is that \(|\hat{q}| < 1\), which violates the requirement of unit quaternions. The solution to that is to consider the error as being a small rotation and use a multiplicative model as in (71), because the quaternion multiplication of two unit-quaternions gives as result a unit-Quaternion.

\[
\hat{q} = \delta \hat{q} \otimes \hat{q} \tag{71}
\]

Quaternions have the advantage of not presenting any singularities, but the disadvantage of being redundant and the extra information adds complexity to the numerical calculation (for instance it is more time demanding to invert a 4×4 matrix than a 3×3). Other parametrizations based on three-component vectors Classic Rodrigues parameters or Gibbs vectors have the advantage of not being redundant, but they do present singularities.

Markley [11] proposes to get the best of the two worlds, using quaternions at the state vector and Gibbs parameters for the error vector. As errors are assumed to be small, they shall never reach the singularity angle (\( \delta \theta << 180^\circ \)) and every time the quaternion is updated, incorporating the error, the Gibbs vector is reset. Thus, at each step of the Kalman loop the reduced vector (Gibbs vector) is used, then it is transformed to quaternion (\( \delta \hat{q} \)) to update the estimate and the error vector is reset back to zero. That procedure guarantees the absence of singularities while using smaller vector, and consequently matrices, in the bulk of the calculations.

The multiplicative model can then be written as in (72), where the rotation error \( \delta \hat{q} \) is now parameterized by a three-component vector \( a \).

\[
\hat{q} = \delta \hat{q}(a) \otimes \hat{q} \tag{72}
\]

The use of this parametrization in a KF context is possible because if \( \hat{a} \equiv E(a) \) is the estimate of \( a \), then \( \delta \hat{q}(\hat{a}) \otimes \hat{q} \) is the estimate of the true quaternion \( \hat{q} \) [6]. The error vector \( a \) has zero mean and its covariance is the attitude error covariance in the body frame. In practical terms, that means that the covariance matrices in the Kalman loop pertaining to attitude error will be 3×3 matrices, instead of 4×4.

Assuming the error consists of small angle rotations, the attitude error can be approximated as in (73), being its vector part equivalent to the three-component parametrization vector.

\[
\delta \hat{q} \approx \left[ \begin{array}{c} \frac{1}{2} \delta \theta \\ 1 \end{array} \right] \tag{73}
\]

There are several different ways to parametrize \( \delta \hat{q} \). Markley [11] presents four alternatives, giving preference to defining the error vector as twice the Gibbs vector, while Trawney & Roumeliotis [12] uses \( \delta \theta \) as error vector parameter. For simplicity, we will use here the latter parametrization.
As the aim of this work is educational, it will be considered here the simplest case, where the turn rate is constant and known. This is in fact of no practical use as the turn rate is either unknown or noisy data from a gyro is to be used. However, the focus here is how to use MEKF with quaternions and multiplicative error.

The governing differential equation has the form of (74):

\[
\ddot{\mathbf{x}} = F_c \dot{\mathbf{x}}
\]

where

\[
\dot{\mathbf{x}} = \delta \dot{\theta}
\]

and \(F_c\) is the continuous time error transition matrix.

To determine \(F_c\), it is necessary to differentiate (72), in order to arrive at (75) (see derivation in Appendix 6).

\[
F_c = -[\omega \times]
\]

In reference [12], the continuous time error transition matrix \(F_c\) was discretized by (76) and then applying a Taylor expansion to arrive at (77):

\[
\Phi = \exp(F_c \Delta t)
\]

\[
\Phi = I_{3 \times 3} - \frac{1}{|\omega|} \sin(|\omega| \Delta t) (\omega \times) + \frac{1}{|\omega|^2} (1 - \cos(|\omega| \Delta t))(\omega \times)^2
\]

The system noise covariance matrix for the continuous time is given by (78) where \(\sigma_0\) is the system noise standard deviation.

\[
Q_c = \sigma_0^2 I_{3 \times 3}
\]

This value represents the degree of trust we have in the model describing the process and it is where, knowing the true state, we can calibrate the KF. The discretization is done simply by multiplying \(\Delta t\) as in (79):

\[
Q_d = Q_c \Delta t
\]

With these elements, we are ready to perform the propagation phase. The state is propagated using (80), which is the discretization of the continuous-time equation for quaternion time derivative (see in appendix 5 equation (A.5.14)).

\[
\tilde{q}_k = \exp \left( \frac{1}{2} \Omega (\omega) \Delta t \right) \tilde{q}_{k-1}
\]

The state covariance matrix is propagated in the usual way according to (63.5).

For the update phase, it is considered the simplest measurement model possible, that means the data received from the position sensor (sun sensor, star tracker, magnetometer, etc.) has already been transformed to quaternions. References [11] and [12] present the way that the transformation shall be carried out. The measurement model is then expressed by a three-component vector expressed by (81):

\[
\mathbf{z} = \tilde{q} \mathbf{r} + \mathbf{n}_m
\]

being \(\mathbf{r}\) the position vector, \(\tilde{q}\) the true attitude quaternion and \(\mathbf{n}_m\) the measurement noise vector with zero mean \(E[\mathbf{n}_m] = 0\) and \(\mathbf{R}\) as measurement covariance matrix \(E[\mathbf{n}_m \mathbf{n}_m^T] = \mathbf{R}\).

The Kalman loop requires that we use a three-component vector measurement error \(\Delta \mathbf{z}\) in the update phase, as shown in (82). The measurement error is the difference between the measurement received from the sensor (81) and the measurement expected to be received if the attitude estimation was a true value, \(\tilde{q}\), which in our simplified measurement model is the attitude estimate before update \(\tilde{q}_k\) expressed in a three-component representation.

\[
\Delta \mathbf{z} = \left( \frac{1}{2} \tilde{q} \mathbf{C}(\tilde{q}) - \frac{1}{2} \tilde{q} \mathbf{C}(\tilde{q})^T \right) \mathbf{r} + \mathbf{n}_m
\]

From this measurement model, the measurement matrix is derived to become as shown in (83) (see derivation in appendix 7).

\[
\mathbf{H} = [\tilde{q} \times]
\]

where \(\tilde{q}\) is the vector part of the estimate quaternion.

The Kalman gain is calculated as usual using (63.1) and the estimate correction will be calculated according to (84):

\[
\delta \tilde{q}^+ = \frac{1}{2} \delta \tilde{\theta} = \frac{1}{2} \mathbf{K} \Delta \mathbf{z}
\]

However, \(\delta \tilde{q}^+\) is a three-component vector. Equations (85) show the way to it in a quaternion form:

\[
\delta \tilde{q}^+ = \begin{bmatrix} \delta \tilde{q}^+ \cr \overline{\tilde{q}^+} \end{bmatrix}
\]

\[
\delta \tilde{q}^+ = \frac{1}{\sqrt{1 + \delta \tilde{q}^+ \delta \tilde{q}^+}} \begin{bmatrix} \delta \tilde{q}^+ \cr 1 \end{bmatrix}
\]

\[
\delta \tilde{q}^+ = \frac{\delta \tilde{q}^+ \delta \tilde{q}^+}{1 + \delta \tilde{q}^+ \delta \tilde{q}^+}
\]

Attitude estimate will be updated by (86) and covariance matrix by (63.3).

\[
\tilde{q}_k = \delta \tilde{q} \otimes \tilde{q}_k
\]

Example 8: Attitude estimation of a spinning object:

This example is about attitude estimation of a spinning object with known turn rate and input from a position sensor. This example is the closest to a space application, as the spinning object could be a spacecraft that needs inputs about its current attitude in order to feed its control system. It is assumed that the turn rate is given, but the initial attitude is unknown. This is a simplification for an educational context, as in a real-life situation the turn rate normally must be estimated either from a gyro output or using the dynamic equations of motions.

The model used here was based on the one presented in [12] and [13] with some adaptations. While these references consider information from a gyro with bias and noise, this example was simplified considering a constant and known turn rate, with the aim of concentrating the attention on the attitude parametrization. The state vector is defined as the four elements of the attitude quaternion, as given in (87):

\[
\mathbf{x}(t) = \tilde{q}(t) = [q_1 \ q_2 \ q_3 \ q_4]^T
\]

The differential equation governing the state is the kinematic equation, which already in the EKF-framework and after discretization is expressed by (88):
\[ \hat{\boldsymbol{q}}_k = \exp \left( \frac{1}{2} \Omega (\omega) \Delta t \right) \hat{\boldsymbol{q}}_{k-1} \]  

(88)

The turn rate was assumed to be known and given by (89):

\[ \omega_{\text{true}} = |\omega_{\text{true}}|e = \frac{2\pi}{5}[1 \ 1 \ 1]^T \]  

(89)

Using (88) and (89), \( \hat{\boldsymbol{q}}_{\text{true}} \) is calculated, considering an initial value of \( \hat{\boldsymbol{q}}_{0} = [0 \ 0 \ 0 \ 1]^T \). To create the values of the position sensor \( z \), already translated to attitude quaternions, each element of \( \hat{\boldsymbol{q}}_{\text{true}} \) was corrupted considering a white noise with standard deviation of 0.8.

The error is a three-dimensional vector \( \delta \theta \) is derived from the small angle approximation of the quaternion error according to (73).

The initial value chosen for the covariance matrix \( P \) was the identity matrix and the initial value for the estimate was chosen differently from the initial true value: \( \hat{\boldsymbol{q}}_0 = [1 \ 0 \ 0 \ 0]^T \) just to test the MEFK code. The results are presented in Figure 15 and it is possible to notice that, despite the added white noise, the estimate approaches the true value after around 5 seconds.

Complexity could be increased by having an unknown turn rate. In that case it would require the data from a gyro and the use of a model that relates the true turn rate to the measured one, considering bias and white noise, as proposed by [11] and [12]. Another alternative would be the inclusion of the spacecraft dynamic equations, as in [22].

![Figure 15 - Attitude estimation of a spinning object with constant turn rate - example of MEKF (\( \omega = 1.26 \) rad/s; \( R_s = 0.1 \); \( Q_s = 0.8 \); \( \Delta t = 0.01 \) s).](image)

IV. DISCUSSION.

The topics presented in section III cover the most important aspects of spacecraft attitude determination and estimation. There are more methods available in literature, but mainly they are variations of those presented here, with different numerical calculation methods for solving the maximization problem in Static attitude determination or slightly different approaches in more sophisticated versions of EKF. Once having grasped the main idea behind the methods discussed here, the student will have enough tools to continue studying by him/herself whenever required.

There might be some challenges during the presentation of those methods for a first-time student. A general one is the visualization of a 3D problem, chiefly because figures are represented in 2D. Most of the students, by the time they take a course on Master’s level, have already been exposed to this problem and have been able to develop their own method of understanding. However, some may still have some difficulties, especially, if they have not yet sedimented the concepts learned in Rigid body kinematics, which is a pre-requisite for an Attitude determination course. One solution for that could be the use of some computer animation or 3D model representing the inertial and body frames and measurement vectors and show that assembly from different angles.

Another general challenge is to establish the level of details in the derivation of the equations. Engineering students are trained to only believe in what can be proved. Some of them will require a very detailed mathematical demonstration, whereas for others, a more general approach is sufficient. The constraint here is normally the time available for lectures and exercises. However, the understanding of concepts, rationale of the models, their limitations and how they are used should be prioritized over the mathematics.

As in other topics included in a Spacecraft dynamics course, it is imperative in the study of Attitude determination methods to detain a solid background in Linear algebra to understand the concepts of eigenvalues, eigenvectors, Lagrange multiplier, among others, and also in Numerical analysis to be able to use for example the Newton-Raphson iteration method.

When it comes to Dynamic attitude estimation and Kalman filtering, the extra required background is statistics. A review on statistics and main concepts as mean value, standard deviation, variance, covariance and normal distribution may be necessary, as well as expected value and estimation theory. References [3] and [5] present good material for such reviews.

When looking at the five equations that makes up the Kalman loop for the first time gives a rather confusing sensation due to the large quantity of variables and complicated notation system. Babb [8] uses in his text different colours for each variable. That helps getting used to the equations, but only repetition will make them become less ugly.

Another aspect that could represent a challenge for the student is the fact that Kalman filtering is a recursive process, thus working with discrete time, while the main process measurement models are created considering continuous time. The discretization process is still necessary, even though it may be complicated, time-demanding and not very useful in the understanding of the attitude estimation concept. One solution could be to supply the students with already discretized models, whenever they need to develop a code for an exercise. Reference [21] may be a useful support for questions of discretization.

Special attention should be given in the definition of the problem. The state vector definition should be frequently reminded, as it is easy to get lost in the mathematics and forget what exactly we are looking for. The dynamics and measurement models are the basis of the whole problem definition and obtaining the transition and observation matrices is not always straightforward. The role of discretization may be explored testing an example with
different time steps and even with different discretization techniques.

Another challenge in understanding Kalman filtering is the definition and role of the process and measurement covariance matrices, $Q$ and $R$, respectively. They may also be explored in an example assuming different values and observing the impact in the results.

When it comes to EKF, the effects of linearization are the main aspect to be stressed. A comparison between KF and EKF, with the aids of, for instance, the sets of equations (39) and (63), is a possible way of showing the differences. A special attention should be given with the “innovation”, the term multiplied by the Kalman gain in equations (39.2) and (63.2).

For the MEKF, the greatest challenge is to work with quaternion algebra. It is assumed that the student has always been trained in that matter before dealing with the attitude determination part of the course. However, some support or refreshing of the main concepts and relations may be useful. Also, the need of parametrization by Gibbs vectors, $\delta \theta$ or any other three-dimension vector, and how to use it, may be subject of questions by the student. Those methods were developed in the beginning of spaceflight when computer capacity was much smaller than what we have today. Probably, there would not be much of a problem dealing with four by four matrices originated by the quaternions, but using those parametrizations has also the educational advantage of providing the student with an opportunity of being proficient with those techniques and concepts.

Finally, another identified challenge were the effects of linearization. By treating non-linear processes with a linearized approximation, errors are being created that may have an accumulative effect inducing to numerical instabilities. Although it deviates from the scope of the course, the risk for instabilities shall be highlighted and some solutions suggested to overcome that problem, so that the student is at least aware of it. Reference [22] uses for instance an adaptive filter method to deal with uncertainties related to the Earth’s magnetic field model. It basically changed the measurement noise covariance to match the variance of the residual. A similar approach could be used to limit the linearization error or simply restart the estimation problem, if we know the time around where the instability starts.

As main textbook to be used in an Attitude determination course, one should consider Markley & Crassidis [3], in specific chapters 5 “Static Determination Methods” and 6 “Filtering for Attitude Estimation and Calibration”. Chapter 12 can also be useful with its review on estimation theory. A deeper explanation on Kalman Filtering and Extended Kalman Filtering can be found in Brown & Hwang [5] chapters 4 and 7, respectively. For Multiplicative Extended Kalman Filtering the reference to be used is Markley’s paper [11]. Those three references configure the theory basis of this course and the remaining references in the reference list may be used as support material.

When it comes to the required workload for the course, Schaub [2] covered the theory on Static attitude determination in two hours of lecture. Stachnis [17] used three hours to lecture on KF and EKF. An additional hour would be needed for MEKF. Ideally the course should also include an extra hour for reviewing estimation theory and quaternion algebra. All together it would be needed seven hours of lecture to cover the theory and at least the same for self-study and exercises.

One suggestion for a grading system could be as following:

- D – the student shall be able to answer conceptual questions related to both static and dynamic attitude determination methods, plus be able to solve exercises involving static attitude determination methods and simple KF problems.
- C – same as for D plus the student shall be able to solve more sophisticated KF problems and exercises involving EKF.
- A – same as for C, plus the student shall be able to solve MEKF problems.

V. CONCLUSION

There is a huge quantity of material on the subject treated in this work available in the internet. They come in different form as books, papers, thesis and reports, but also in more informal ways as videos, blogs and webpages. Each one concentrate in different aspects and convey the information in different levels of difficulty.

This work did not have the intention to exhaust all sources but to fish up from the sea of references those that covered the concepts and the way to use the algorithms in an easy and understandable way in a master’s degree student level.

I tried to use what I experienced in my own learning process to identify the most interesting and important aspect of the subject as well as the challenges I faced during this journey.

ACKNOWLEDGMENT

It really was a journey from the moment I first heard the expressions attitude determination (“isn’t it just reading a sensor output?”) and Kalman filter (“what’s the filter?”) till the conclusion of this work.

I’d like to thank everyone that supported me during this journey, especially Dr. Gunnar Tibert for suggesting the subject of this work, giving directions when the path seemed to disappear in the middle of the chaos and for the inspiration; Federico Raiti for his patience and important explanations on MEKF and my family, Janaíra, Felipe, Beatrice and Anna Carolina for the unconditional love, patience and support.

REFERENCES


Appendix 1 – Example 1 numerical solution – TRIAD method:

1) \( \theta_{true} = [30^\circ \ 20^\circ \ -10^\circ] \)

\[
C_{BNtrue} = \begin{bmatrix}
0.8138 & 0.4698 & -0.3420 \\
-0.5438 & 0.8232 & -0.1632 \\
0.2049 & 0.3188 & 0.9254
\end{bmatrix}
\]

2) \( b^v_1 = C_{BNtrue}^N n_1 = [0.8138 \ -0.5438 \ 0.2049]^T \)

\( b^v_2 = C_{BNtrue}^N n_2 = [-0.3420 \ -0.1632 \ 0.9254]^T \)

3) \( b^v_1 = [0.8190 \ -0.5282 \ 0.2242]^T \)

\( b^v_2 = [-0.3138 \ -0.1584 \ 0.9362]^T \)

4) \( b^t_1 = n^v_1 = [0.8190 \ -0.5282 \ 0.2242]^T \)

\( b^t_2 = \frac{b^v_1 \times b^v_2}{|n^v_1 \times n^v_2|} = [-0.4593 \ -0.8376 \ -0.2957]^T \)

\( b^t_3 = n^t_1 \times b^t_2 = [0.3440 \ 0.1392 \ -0.9286]^T \)

5) \( n^t_1 = n^v_1 = [1 \ 0 \ 0]^T \)

\( n^t_2 = n^v_1 \times n^v_2 = [0 \ -1 \ 0]^T \)

\( n^t_3 = n^t_1 \times n^t_2 = [0 \ 0 \ -1]^T \)

6) \( C_{BT} = \begin{bmatrix}
b^t_1 & b^t_2 & b^t_3
\end{bmatrix} = \begin{bmatrix}
0.8190 & -0.4593 & 0.3440 \\
-0.5282 & -0.8376 & 0.1392 \\
0.2242 & -0.2957 & -0.9286
\end{bmatrix} \)

\( C_{NT} = \begin{bmatrix}
n^t_1 & n^t_2 & n^t_3
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & -1
\end{bmatrix} \)

7) \( C_{BN} = C_{BT} C_{NT}^T = \begin{bmatrix}
0.8190 & 0.4593 & -0.3440 \\
-0.5282 & 0.8376 & -0.1392 \\
0.2242 & 0.2957 & 0.9286
\end{bmatrix} \)

8) \( C_{BB} = C_{BN} C_{BNtrue}^T = \begin{bmatrix}
0.9999 & -0.0112 & -0.0041 \\
0.0113 & 0.9995 & 0.0300 \\
0.0038 & -0.0301 & 0.9996
\end{bmatrix} \)

9) \( e_{BB} = [0.9294 \ 0.1218 \ -0.3484]^T \)

\( \varphi = 0.0323 \text{ rad} = 1.8525^\circ \)

Appendix 2 – Example 2 numerical solution – Devenport’s q-method:

1), 2) and 3) same as in previous example (Appendix 1)

4) \( w_1 = w_2 = 1 \)

\( B = w_1 b^v_1 n^v_1 + w_2 b^v_2 n^v_2 = \begin{bmatrix}
0.8190 & 0 & -0.3138 \\
-0.5282 & 0 & -0.1584 \\
0.2242 & 0 & 0.9362
\end{bmatrix} \)

\( S = B + B^T = \begin{bmatrix}
1.6380 & -0.5282 & -0.0896 \\
-0.5282 & 0 & -0.1584 \\
-0.0896 & -0.1584 & 1.8724
\end{bmatrix} \)

\( \sigma = \text{Tr}(B) = 1.7552 \)

\( Z = [B_{23} - B_{32} \ B_{31} - B_{13} \ B_{12} - B_{21}]^T = \begin{bmatrix}
-0.1584 & 0.5380 & 0.5282
\end{bmatrix}^T \)

5) \( K = \begin{bmatrix}
S - \sigma I_{3 \times 3} & Z \\
Z^T & \sigma
\end{bmatrix} = \begin{bmatrix}
-0.1172 & -0.5282 & -0.0896 & -0.1584 \\
-0.5282 & -1.7552 & -0.1584 & 0.5380 \\
-0.0896 & -0.1584 & 0.1172 & 0.5282 \\
-0.1584 & 0.5380 & 0.5282 & 1.7552
\end{bmatrix} \)

6) \( \lambda = [-1.9997 \ -0.0366 \ 0.0366 \ 1.9997] \)

7) \( q_{BN} = 4\text{th eigenvector} = \begin{bmatrix}
-0.1172 & 0.1414 & 0.2597 & 0.9481
\end{bmatrix}^T \)

The scalar part of \( q_{BN} \) was negative, indicating the longer rotation. To obtain the shorter rotation, multiply \( q_{BN} \) by \(-1\).

8) \( q_{BN} \rightarrow C_{BN} = \begin{bmatrix}
0.8251 & 0.4593 & -0.3289 \\
-0.5256 & 0.8376 & -0.1488 \\
0.2072 & 0.2957 & 0.9326
\end{bmatrix} \)

\( C_{BB} = C_{BN} C_{BNtrue}^T \begin{bmatrix}
0.9998 & -0.0170 & 0.0111 \\
0.0168 & 0.9996 & 0.0216 \\
-0.0114 & -0.0215 & 0.9997
\end{bmatrix} \)

9) \( e_{BB} = [0.7282 \ -0.3800 \ -0.5703]^T \)

\( \varphi = 0.0296 \text{ rad} = 1.6960^\circ \)

Appendix 3 - Example 4 numerical solution – OLAE method:

1), 2) and 3) same as in previous example (Appendix 1)
Appendix 4 – Derivation of LKF process and measurement models:

Initial process and measurement models:

\[ \dot{x} = f(x, u_d, t) + u(t) \]  
(A4.1)

\[ z = h(x, t) + v(t) \]  
(A4.2)

Reference trajectory:

\[ x(t) = x^*(t) + \Delta x(t) \]  
(A4.3)

Substituting (A4.3) in (A4.1) and (A4.2):

\[ \dot{x}^* + \Delta \dot{x} = f(x^*, \Delta x, u_d, t) + u(t) \]  
(A4.4)

\[ z = h(x^* + \Delta x, t) + v(t) \]  
(A4.5)

Taylor series expansion of (A4.4) and (A4.5):

\[ \dot{x}^* + \Delta \dot{x} \approx f(x^*, u_d, t) + F \Delta x + u(t) \]  
(A4.6)

\[ z \approx h(x^*, t) + H \Delta x + v(t) \]  
(A4.7)

where

\[ F = \left[ \frac{\partial f}{\partial x} \right]_{x=x^*} \]  
(A4.8)

\[ H = \left[ \frac{\partial h}{\partial x} \right]_{x=x^*} \]  
(A4.9)

Appendix 5 – Quaternion algebra review [12]:

Quaternion definition:

\[ \bar{q} = q_4 + q_1 i + q_2 j + q_3 k \]  
(A5.1)

where \( i^2 = j^2 = k^2 = -1 \).

The coordinate system used by the great majority of the author is the left-handed coordinate system. Only one author among references surveyed uses a right-handed coordinate system, but eventually at some point he uses some transformations, which are equivalent to using a left-handed coordinate system from the start. No further explanation for this procedure was found other than to avoid numerical instability [3]. The difference between using those two coordinate systems appears only in quaternion multiplication and when transforming quaternion to DCM. The formulation presented in most references are consistent and should not create any compatibility problem.

In a left-handed coordinate system, following relations are valid:

\[ -ij = ji = k; \quad -jk = k = i; \quad -ki = ik = j \]  
(A5.2)

Quaternion:

\[ \bar{q} = \begin{bmatrix} q_4 & q_1 & q_2 & q_3 \end{bmatrix}^T \]  
(A5.3.1)

\[ \varphi = 0.0323 \text{ rad} = 1.8525^\circ \]

where, vector part is:

\[ q = \begin{bmatrix} e_x \sin(\theta/2) \\
 e_y \sin(\theta/2) \\
 e_z \sin(\theta/2) \end{bmatrix} = \bar{e} \sin(\theta/2) \]  
(A5.3.2)

Scalar part:

\[ q_4 = \cos(\theta/2) \]  
(A5.3.3)

Norm:

\[ ||\bar{q}|| = \sqrt{\bar{q}^T \bar{q}} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \]  
(A5.3.4)
Quaternion multiplication:
$$\vec{q} \otimes \vec{p} = \begin{bmatrix} q_4 & [q \times] & \vec{q} \end{bmatrix} \vec{p}$$ (A5.4.1)
where the skew-symmetric matrix operator is:
$$[q \times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$ (A5.4.2)
which is also used for vector cross-product:
$$q \times p = [q \times] p = \begin{bmatrix} q_2 p_3 - q_3 p_2 \\ q_3 p_1 - q_1 p_3 \\ q_1 p_2 - q_2 p_1 \end{bmatrix}$$ (A5.4.3)

Neutral quaternion (identity): $$\vec{q}_0 = [0 \ 0 \ 0 \ 1]^T$$ meaning,
$$\vec{q} \otimes \vec{q}_0 = \vec{q}_0 \otimes \vec{q} = \vec{q}$$ (A5.5)

Inverse rotation:
$$\vec{q}^{-1} = \frac{-[\vec{q} \times]}{q_4} = \begin{bmatrix} e\sin(\theta /2) \\ \cos(\theta /2) \end{bmatrix} = \begin{bmatrix} e\sin(- \theta /2) \\ \cos(- \theta /2) \end{bmatrix}$$ (A5.6.1)
$$\vec{q} \otimes \vec{q}^{-1} = \vec{q}^{-1} \otimes \vec{q} = \vec{q}_0$$ (A5.6.2)
$$(\vec{q} \otimes \vec{p})^{-1} = \vec{p}^{-1} \otimes \vec{q}^{-1}$$ (A5.6.3)

Useful identities:
$$[w \times] = -[w \times]^T$$ (A5.7.1)
$$[a \times]b = -[b \times]a \leftrightarrow a^T [b \times] = -b^T [a \times]$$ (A5.7.2)
$$[a \times] + [b \times] = [[a + b] \times]$$ (A5.7.3)
$$c[w \times] = [(cw) \times]$$ (A5.7.4)
$$w \times (cw) = c[w \times]w = 0_{3 \times 1}$$ (A5.7.5)

Lagrange formula:
$$[a \times][b \times] = b a^T - (a^T b) I_{3 \times 3}$$ (A5.8)

Powers of $[w \times]$
$$[w \times]^2 = w w^T - [w]^2 I_{3 \times 3}$$ (A5.9.1)
$$[w \times]^3 = -[w]^2 [w \times]$$ (A5.9.2)
$$[w \times]^4 = -[w]^3 [w \times]^2$$ (A5.9.3)
$$[w \times]^5 = [w]^4 [w \times]$$ (A5.9.4)
$$[w \times]^6 = [w]^5 [w \times]^2$$ (A5.9.5)
$$[w \times]^7 = [w]^6 [w \times]$$ (A5.9.6)

Matrix $\Omega$ (to be used in quaternion derivative):
$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y & \omega_x \\ -\omega_z & 0 & -\omega_x & \omega_y \\ -\omega_y & \omega_x & 0 & -\omega_z \\ -\omega_x & -\omega_y & \omega_z & 0 \end{bmatrix}$$

Powers of $\Omega$:
$$\Omega(\omega)^2 = -|\omega|^2 I_{4 \times 4}$$ (A5.10.1)
$$\Omega(\omega)^3 = -|\omega|^2 \Omega(\omega)$$ (A5.10.2)
$$\Omega(\omega)^4 = |\omega|^4 I_{4 \times 4}$$ (A5.10.3)
$$\Omega(\omega)^5 = |\omega|^4 \Omega(\omega)$$ (A5.10.4)
$$\Omega(\omega)^6 = -|\omega|^6 I_{4 \times 4}$$ (A5.10.5)

Relationship between quaternion and rotational matrix:
$$\begin{bmatrix} 0 \\ \tilde{q} \end{bmatrix} = \frac{1}{t} C(\tilde{q})^G p$$ (A5.11.1)
$$\frac{1}{t} C(\tilde{q}) = (2q_4^2 - 1)I_{3 \times 3} - 2q_4[q \times] + 2q q^T$$ (A5.11.2)
Where $L$ denotes the local frame and $G$, the inertial (global) one.

Triple product of quaternion:
$$\vec{q} \otimes \vec{p} \otimes \vec{q}^{-1} = \begin{bmatrix} C(\tilde{q}) & 0 \end{bmatrix} \begin{bmatrix} p_4 \end{bmatrix} = \begin{bmatrix} C(\tilde{q}) p \end{bmatrix}$$ (A5.12)

Very small rotation simplification:
$$\delta \tilde{q} = \begin{bmatrix} \delta q_4 \\ \frac{1}{2} \delta \theta \end{bmatrix} \approx \begin{bmatrix} \frac{1}{2} \delta \theta \\ 1 \end{bmatrix}$$ (A5.13.1)
$$\Rightarrow \frac{1}{t} C(\delta \tilde{q}) = I_{3 \times 3} - [\delta \theta \times]$$ (A5.13.2)

Quaternion time derivative:
$$\frac{d(\tilde{q})}{dt}(t) = \frac{1}{\Delta t} \lim_{\Delta t \to 0} \left( \frac{L(t + \Delta t) - L(t)}{\Delta t} \tilde{q} \otimes \frac{L(t)}{\delta q} \right)$$
$$\approx \frac{1}{\Delta t} \lim_{\Delta t \to 0} \left( \frac{1}{2} \frac{\delta}{\delta \theta} \right) \frac{\delta(\tilde{q})}{\delta \theta} = \frac{1}{2} \frac{\delta(\tilde{q})}{\delta \theta}$$ (A5.14)
$$\Rightarrow \frac{d(\tilde{q})}{dt}(t) = \begin{bmatrix} \Omega(\omega) \end{bmatrix} \frac{1}{t} \frac{\delta(\tilde{q})}{\delta \theta}$$ (A5.15.1)

Quatintion integration:
The solution of the differential equation has the form of:
$$\frac{d(\tilde{q})}{dt}(t) = \begin{bmatrix} \dot{\omega} \end{bmatrix} \frac{1}{t} \tilde{q}(t_k)$$ (A5.15.1)
Reordering and differentiating:
\[ \dot{\theta}(t, t_k) = \frac{\sin(t)}{\sin(\theta/2)} \quad (A5.21) \]

Gibbs Vector or Rodrigues Parameters [11]:
\[ \mathbf{g} \equiv \frac{a}{q_4} = \frac{\dot{\theta}}{\sin(\theta/2)} \quad (A5.21) \]

Gibbs to DCM:
\[ \Lambda(\mathbf{g}) = I_{3 \times 3} - 2(1 + g^2)^{-1} \left[ [g \times] - [g \times]^2 \right] \quad (A5.22) \]

Appendix 6 – Derivation of continuous time error state equations for MEKF:
\[ \frac{d}{dt}(\mathbf{q}) = \frac{d}{dt}(\delta \dot{\mathbf{q}} \otimes \mathbf{q}) \Rightarrow \dot{\mathbf{q}} = \delta \dot{\mathbf{q}} \otimes \mathbf{q} + \delta \mathbf{q} \otimes \dot{\mathbf{q}} \quad (A6.2) \]

(A6.1) into (A6.2)
\[ \left[ \frac{[\mathbf{q}]}{[\mathbf{q_4}]} \right] = \delta \dot{\mathbf{q}} \otimes \mathbf{q} + \delta \mathbf{q} \otimes \left( \frac{[\mathbf{q}]}{[\mathbf{q_4}]} \right) \quad (A6.3) \]

Rearranging (A6.3)
\[ \delta \dot{\mathbf{q}} \otimes \mathbf{q} = \frac{1}{2} \left( [\mathbf{q}] \otimes \mathbf{q} - \delta \mathbf{q} \otimes \left( [\mathbf{q}] \otimes \mathbf{q} \right) \right) \quad (A6.4) \]

Multiplying (A6.4) by \( \mathbf{q}^{-1} \)
\[ \delta \hat{\mathbf{q}} = \frac{1}{2} \left( [\mathbf{q}] \otimes \mathbf{q} - \delta \mathbf{q} \otimes [\mathbf{q}] \right) = [\mathbf{q}] \otimes \delta \mathbf{q} \quad (A6.5) \]

Using quaternion multiplication defined in (5.4.1):
\[ \delta \hat{\mathbf{q}} = \frac{1}{2} \left( \left[ \begin{array}{c} -[\mathbf{q} \times] \\
\mathbf{0} \end{array} \right] \otimes \delta \mathbf{q} - \left[ \begin{array}{c} -[\mathbf{q} \times] \\
\mathbf{0} \end{array} \right] \otimes \delta \mathbf{q} \right) \]
\[ = \frac{1}{2} \left( \left[ \begin{array}{c} -[\mathbf{q} \times] \\
\mathbf{0} \end{array} \right] \otimes \delta \mathbf{q} - \left[ \begin{array}{c} -[\mathbf{q} \times] \\
\mathbf{0} \end{array} \right] \otimes \delta \mathbf{q} \right) \]
\[ = \frac{1}{2} \left[ -2[\mathbf{q} \times] \right] \delta \mathbf{q} = -[\mathbf{q} \times] \delta \mathbf{q} \quad (A6.6) \]

Considering the small angle approximation:
\[ \delta \hat{\mathbf{q}} = \left( \begin{array}{c} \delta \mathbf{q} \\
\delta \mathbf{q_4} \end{array} \right) = \left( \begin{array}{c} \mathbf{k} \sin(\theta/2) \\
\cos(\theta/2) \end{array} \right) \approx \left( \begin{array}{c} \delta \theta/2 \\
0 \end{array} \right) \quad (A6.7) \]

The derivative of (A6.7) becomes:
\[ \delta \dot{\mathbf{q}} = \left( \begin{array}{c} \delta \theta/2 \\
0 \end{array} \right) \quad (A6.8) \]

Substituting (A6.7) and (A6.8) into (A6.6), we conclude that
\[ \delta \hat{\theta} = -[\mathbf{q} \times] \delta \theta \quad (A6.9) \]

And thus,
\[ \Rightarrow \mathbf{F}_c = -[\mathbf{q} \times] \quad (A6.10) \]

Appendix 7 – Derivation of measurement matrix for MEKF:
\[ \Delta \mathbf{z} = \left( \frac{\mathbf{k}}{5} \mathbf{C}(\mathbf{q}) \right) \mathbf{r} + \mathbf{n}_m \quad (A7.1) \]
\[ \mathbf{q} = \delta \mathbf{q} \otimes \mathbf{q} \quad (A7.2) \]
\[ \frac{\mathbf{k}}{5} \mathbf{C}(\mathbf{q}) = \frac{\mathbf{k}}{5} \mathbf{C}(\delta \mathbf{q} \otimes \mathbf{q}) \quad (A7.3) \]

Consider:
\[ \mathbf{q} = \left[ \begin{array}{c} \mathbf{q} \\
\mathbf{q_4} \end{array} \right] \quad (A7.4) \]
\[ \delta \hat{\mathbf{q}} \approx \left( \begin{array}{c} \delta \theta/2 \\
0 \end{array} \right) \quad (A7.5) \]
\[ \frac{\partial}{\partial t} C(q) = (2q^2 - 1)I_{3\times3} - 2q^2 [q \times] + 2qq^r \quad (A7.6) \]

(A7.5) into (A7.6):
\[ \Rightarrow \frac{\partial}{\partial t} (\delta q) \approx I_{3\times3} - [\delta \theta \times] \quad (A7.7) \]

(A7.7) into (A7.3):
\[ \Rightarrow \frac{\partial}{\partial t} C(q) = (I_{3\times3} - [\delta \theta \times]) \frac{\partial}{\partial t} C(q) \]
\[ \Rightarrow \frac{\partial}{\partial t} C(q) - \frac{\partial}{\partial t} C(q) = (I_{3\times3} - [\delta \theta \times]) \frac{\partial}{\partial t} C(q) - \frac{\partial}{\partial t} C(q) \]
\[ = -[\delta \theta \times] \frac{\partial}{\partial t} C(q) \quad (A7.8) \]

(A7.8) into (A7.1)
\[ \Delta z = -[\delta \theta \times] \frac{\partial}{\partial t} C(q) \cdot \gamma _{\oplus} + n_m \]
\[ \Rightarrow \Delta z = \left[ \frac{\partial}{\partial t} C(q) \cdot \gamma _{\oplus} \times \right] \delta \theta + n_m \quad (A7.9) \]

As \( \Delta z = H \cdot \delta \theta + n_m \)
\[ \Rightarrow H = \left[ \frac{\partial}{\partial t} C(q) \cdot \gamma _{\oplus} \times \right] = [q \times] \quad (A7.10) \]