NATIONAL TECHNICAL UNIVERSITY OF ATHENS

SCHOOL OF MECHANICAL ENGINEERING

Diploma Thesis

Nonlinear State Estimation for Small Unmanned Aircraft

by

Dimitrios Stylianos Parsinas Pylorof

Thesis supervisor: Professor Kostas J. Kyriakopoulos

Submitted in March 2012 in partial fulfillment of the requirements
for the degree of

Diploma in Mechanical Engineering

Dimitrios Stylianos Parsinas Pylorof

email: dpylorof@gmail.com

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

Μη-γραμμική εκτίμηση κατάστασης μικρών, μη επανδρωμένων αεροσκαφών

του

Δημήτριου Στυλιανού Παρσινά Πυλόρωφ

Επιβλέπων: Κώστας Ι. Κυριακόπουλος
Καθηγητής ΕΜΠ

Υποβλήθηκε το Μάρτιο του 2012 για τη μερική εκπλήρωση των υποχρεώσεων για την απόδοση του

Διπλώματος Μηχανολόγου Μηχανικού

Δημήτριος Στυλιανός Παρσινάς Πυλόρωφ

email: dpylorof@gmail.com

to my beloved mother

Page intentionally left blank

Nonlinear State Estimation for Small Unmanned Aircraft

by

Dimitrios Stylianos Parsinas Pylorof

Abstract

This thesis is concerned with the nonlinear state estimation problem for small unmanned aircraft. State estimation is an integral part of any unmanned aircraft's flight control system. The quantities required for flight control, like attitude, 3D position and velocity, are not usually measured directly or they are not available at the frequency required; they have to be derived or extrapolated, respectively, from the available sensor measurements. Furthermore, due to size and cost, small unmanned aircraft carry rather noisy and even biased microelectromechanical sensors, complicating the problem even further.

The usual sensor payload for the aircraft class of interest is an Inertial Measurement Unit (IMU), containing three accelerometers and gyroscopes that measure accelerations and angular velocities, respectively, in the inertial body frame. The aircraft's velocity, position and attitude can be obtained through integrations at the IMU output rate. Due to the IMU's output noise and bias though, a trivial integration is not enough as there will be an ever growing drifting error. A Global Navigation Satellite System (GNSS) receiver is used to provide a direct measurement of the aircraft's position and velocity at a much lower frequency but with bounded error characteristics. The measurements from the IMU and the GNSS receiver should be fused in a state estimation framework, to filter out the sensor noise, compensate for bias and provide the best possible state estimate.

A solution to the small unmanned aircraft state estimation problem is presented here, using a modified version of the nonlinear Unscented Kalman Filter (UKF). The original Kalman Filter has been the de facto solution for real time estimation problems since its

inception, however it cannot be used in aircraft state estimation as the underlying kinematic and measurement equations are nonlinear; its nonlinear extension also, namely the Extended Kalman Filter (EKF) is ruled out because it can exhibit numerical issues and even divergence. The UKF is one of the most promising nonlinear state estimation algorithms for use in real time applications; it is based on a deterministic sampling principle which avoids linearizations and calculation of Jacobian matrices, in direct contrast with the derivative based EKF. In the solution presented here, the singularity free attitude quaternion is used for the representation of the aircraft's attitude. The quaternion, however, is a constrained quantity and that imposes several modifications to the UKF algorithm. As far as the aircraft's 3D kinematics are concerned, minimal assumptions are made in the derivation of the respective estimator models; the Earth's ellipsoidal model is used, allowing almost global operation of the aircraft with no modifications.

A numerically efficient version of the algorithm that requires the least possible number of function evaluations per cycle is derived. The results are presented and verified through a simulation example. Although this thesis has been developed to provide a solution to the state estimation problem of the small unmanned helicopter built by the NTUA Control Systems Lab, the resulting algorithms can be readily used onboard any small unmanned aircraft with similar sensor payload, either fixed wing or rotorcraft.

Acknowledgments

I would like to sincerely thank **Professor Kostas J. Kyriakopoulos** for his continuous support and supervision throughout my studies in the Mechanical Engineering School of the National Technical University of Athens, and for trusting me to work as a research assistant in his lab since the beginning of my 3rd year in September 2008.

I would also like to thank **Dr. George Karras** and **Ph.D. Candidate Mr. Panos Marantos** for their advice, help and guidance during the development of this thesis.

Μη-γραμμική εκτίμηση κατάστασης μικρών μη επανδρωμένων αεροσκαφών

του Δημήτριου Στυλιανού Παρσινά Πυλόρωφ

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται τη μη-γραμμική εκτίμηση κατάστασης μικρών μη επανδρωμένων αεροσκαφών. Η εκτίμηση κατάστασης είναι αναγκαίο συστατικό στοιχείο του συστήματος αυτομάτου ελέγχου της πτήσης οιουδήποτε μη επανδρωμένου αεροσκάφους. Τα μεγέθη που απαιτούνται για τον έλεγχο της πτήσης, όπως ο προσανατολισμός, η θέση και η ταχύτητα στις τρεις διαστάσεις, συνήθως δεν μετρώνται άμεσα ή δεν είναι διαθέσιμα στην επιθυμητή συχνότητα, επομένως πρέπει να υπολογισθούν από τις όποιες διαθέσιμες μετρήσεις. Επίσης, λόγω κόστους και βάρους, τα μικρά μη επανδρωμένα αεροσκάφη φέρουν αισθητήρες τύπου MEMS που παρουσιάζουν ιδιαίτερα αυξημένο θόρυβο στις μετρήσεις τους, ακόμα και bias, περιπλέκοντας περαιτέρω το πρόβλημα.

Ο βασικός αισθητήρας που φέρεται από τα αεροσκάφη που εξετάζονται εδώ είναι ένα Inertial Measurement Unit (IMU) με τρία επιταχυνσιόμετρα και τρία γυροσκόπια τα οποία μετρούν επιταχύνσεις και γωνιακές ταχύτητες αντίστοιχα στο αδρανειακό σύστημα αναφοράς. Η θέση, η ταχύτητα και ο προσανατολισμός του αεροσκάφους μπορούν να προκύψουν εξ' ολοκληρώσεως, όποτε υπάρχουν μετρήσεις από το IMU. Ωστόσο, λόγω της ύπαρξης θορύβου στις μετρήσεις αλλά και bias, μια απλή ολοκλήρωση δεν είναι αρκετή καθώς θα υπάρχει ένα διαρκώς αυξανόμενο σφάλμα στα αποτελέσματα. Για το λόγο αυτό, ένας δέκτης για κάποιο παγκόσμιο δορυφορικό σύστημα πλοήγησης (GNSS) χρησιμοποιείται επικουρικά ώστε να παρέχει μια άμεση μέτρηση της θέσης και της ταχύτητας του αεροσκάφους, σε πολύ μικρότερη συχνότητα από το IMU αλλά με φραγμένο και γνωστό σφάλμα χωρίς bias. Οι μετρήσεις από το IMU και το GNSS πρέπει να συνδυασθούν κατάλληλα, ώστε να απορριφθεί ο θόρυβος και να προκύψει η καλύτερη δυνατή εκτίμηση της κατάστασης του αεροσκάφους.

Μια λύση στο ανωτέρω πρόβλημα της εκτίμησης κατάστασης για μικρά μη επανδρωμένα αεροσκάφη παρουσιάζεται εδώ, χρησιμοποιώντας μια παραλλαγή του Unscented Kalman Filter (UKF). Ο πρωτότυπος αλγόριθμος του Kalman Filter είναι η τυπική λύση για πληθώρα προβλημάτων εκτίμησης κατάστασης από τη στιγμή που προτάθηκε παρέχοντας τη βέλτιστη εκτίμηση κατάστασης και λαμβάνοντας υπόψη το δυναμικό μοντέλο της προς εκτίμηση διεργασίας, ωστόσο δεν μπορεί να χρησιμοποιηθεί για την εκτίμηση κατάστασης ενός αεροσκάφους καθώς οι κινηματικές εξισώσεις που διέπουν την τρισδιάστατη κίνησή του και η συσχέτιση των μετρήσεων με το διάνυσμα κατάστασης είναι μη γραμμικά συστήματα. Η μη γραμμική επέκτασή του, το Extended Kalman Filter (EKF) έχει σοβαρά μειονεκτήματα ως λύση καθώς μπορεί να προκληθούν αριθμητικές αστάθειες αλλά ακόμα και απόκλιση της εκτίμησης. Το UKF είναι ίσως ο πιο υποσχόμενος αλγόριθμος εκτίμησης κατάστασης για χρήση σε εφαρμογές πραγματικού χρόνου. Στηρίζεται σε μια μεθοδολογία ντετερμινιστικής δειγματοληψίας η οποία αποφεύγει εντελώς τις γραμμικοποιήσεις και το σχηματισμό Ιακωβιανών μητρών, σε αντίθεση με το EKF. Στη λύση που παρουσιάζεται εδώ, χρησιμοποιείται το quaternion για την αναπαράσταση του προσανατολισμού του αεροσκάφους, επιτρέποντας κάθε κίνηση / περιστροφή αφού δεν υπάρχουν οι περιορισμοί των γωνιών Euler. Το quaternion ωστόσο, επειδή δεν είναι διάνυσμα αλλά μια ποσότητα περιορισμένη στο SO(3), επιβάλει διάφορες μεταβολές στις εξισώσεις του αλγορίθμου του UKF. Όσον αφορά στην τρισδιάστατη κινηματική του αεροσκάφους, γίνονται οι ελάχιστες παραδοχές στην ανάπτυξη των εξισώσεων του κινηματικού μοντέλου αφού λαμβάνεται υπόψη το πλήρες ελλειψοειδές μοντέλο της γης και άρα είναι δυνατή η σχεδόν παγκόσμια λειτουργία του αλγορίθμου χωρίς τροποποιήσεις.

Στο κείμενο που ακολουθεί αναπτύσσεται μια αριθμητικά βελτιωμένη εκδοχή του σχετικού αλγορίθμου, η οποία απαιτεί τις λιγότερες δυνατές εκτελέσεις των μοντέλων διεργασίας και μετρήσεων ανά επανάληψη. Τα αποτελέσματα παρουσιάζονται και επιβεβαιώνονται μέσα από προσομοίωση. Παρότι η παρούσα εργασία αναπτύχθηκε ώστε να παρέχει μια λύση στο πρακτικό πρόβλημα της εκτίμησης κατάστασης του μικρού μη επανδρωμένου ελικοπτέρου του Εργαστηρίου Αυτομάτου Ελέγχου του ΕΜΠ, οι προκύπτοντες αλγόριθμοι και τα σχετικά συμπεράσματα μπορούν να αξιοποιηθούν πάνω σε οποιοδήποτε μικρό μη επανδρωμένο

αεροσκάφος με παρόμοιους αισθητήρες, είτε αεροπλάνο, είτε ελικόπτερο.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω πολύ τον **Καθηγητή Κώστα Ι. Κυριακόπουλο** για τη συνεχή στήριξη και επίβλεψή του καθ' όλη τη διάρκεια των σπουδών μου στη Σχολή Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, καθώς και για την εμπιστοσύνη που έδειξε στο πρόσωπό μου προσφέροντάς μου τη δυνατότητα να εργαστώ ως ερευνητής στο εργαστήριό του από την αρχή του 3ου μου έτους στη σχολή το Σεπτέμβριο του 2008.

Επίσης θα ήθελα να ευχαριστήσω το **Δρ. Γιώργο Καρρά** και τον **Υποψ. Διδάκτορα Πάνο Μαράντο** για τις συμβουλές, τη βοήθεια και για την καθοδήγησή τους κατά την ανάπτυξη της παρούσας εργασίας.

Σημείωση

Λόγω της πληθώρας όρων της θεωρίας δυναμικών συστημάτων, αυτομάτου ελέγχου και αεροναυπηγικών όρων, που δυστυχώς δεν τυγχάνουν επιτυχημένης απόδοσης στα ελληνικά αλλά εκ των πραγμάτων εμπεριέχονται στην παρούσα εργασία, το κείμενο που ακολουθεί είναι στην αγγλική γλώσσα.

Table of Contents

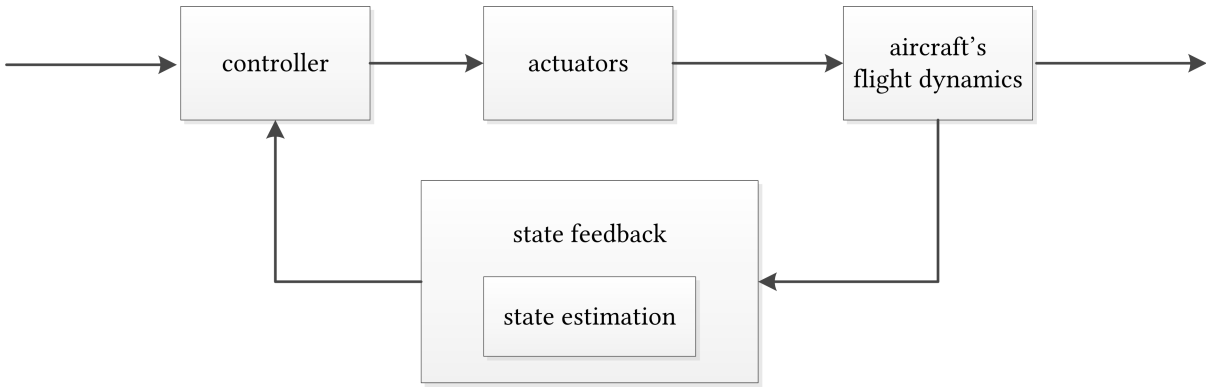
Chapter 1

Preliminaries

In this chapter the small unmanned aircraft state estimation problem is described and a discussion of the various estimated quantities follows. The chapter concludes with a review of the typical sensors used in small unmanned aircraft, namely Inertial Measurement Units (IMUs) and Global Navigation Satellite System (GNSS) receivers.

1.1 Estimator requirements

Any flight control system, regardless of the underlying theory that makes it work (PID, Model Predictive Control, Adaptive Control, etc.), requires an accurate estimate of the aircraft's kinematic state at the beginning of every control iteration [1].



As there is no set of sensors that can provide the quantities of interest at the required frequency and with acceptable measurement noise characteristics, they have to be estimated from the available measurements, making, thus, state estimation an integral part of the system's feedback loop. Given that an aircraft can move in 6 degrees of freedom, the state vector to be estimated is comprised of:

- the aircraft's **position on Earth**
- the aircraft's three dimensional **velocity**
- the aircraft's **attitude** with respect to the Earth's North, East and Down axes

There are many ways to represent and propagate the above quantities. To allow for the greatest possible maneuverability, the attitude quaternion [13] has been chosen for the representation of attitude in the estimation algorithm developed here, as it provides a singularity free attitude representation solution and also features a rather convenient linear time propagation model. The attitude quaternion is further discussed in section 1.3. The aircraft's position is represented and propagated using curvilinear coordinates in the ECEF frame [2], [3], [4]. As far as the velocity is concerned, it can be either represented in the body frame or in the local frame axes with similar formulation, results and accuracy; here the local frame approach is chosen.

The **accuracy** of the resulting state estimate is of great importance for the success of the aircraft's mission. An accurate estimate of the state vector means that the aircraft can perform well even in demanding scenarios, like formation flight, flight near obstacles, automatic takeoff and landing, etc. On the other hand, a decrease in estimation accuracy can severely hinder the aircraft's capabilities. The estimate's accuracy is determined by the sensors present and operating at any moment, their noise characteristics and the estimation algorithm.

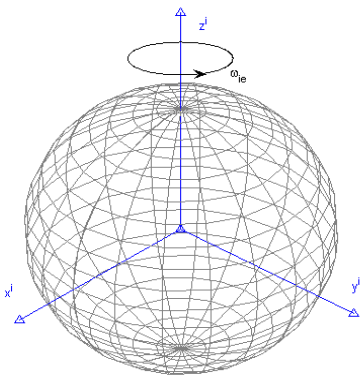
As a new state estimate must be computed before the beginning of each control iteration and since these control iterations can happen at 50 ms or at even more frequent intervals

in the flight control systems of small unmanned aircraft, the estimation process itself should last as less time as possible. Of course this imposes some performance requirements on the sensors as well, but the estimation software's **computational cost** is the main concern here

1.2 Coordinate frames

Coordinate frames consist of a center and axes that are used to describe the motion of a body, its position and its orientation. The discussion of coordinate frames is of great importance in the aircraft state estimation problem, since the quantities that have to be estimated and the ones that are measured are described in different frames (position / velocity / acceleration) or they involve the orientation of one frame with respect to another (attitude).

In this section the various frames of interest, namely the Earth Centered Inertial (ECI), the Earth Center Earth Fixed (ECEF), the local frame and the body frame, are described. Subsequently, the aircraft kinematic model that will be used later in the state estimator's formulation is derived. This section follows the development of the topic found in [2] and [3].



1.2.1 Earth Centered Inertial

The Earth Centered Inertial frame (ECI) is centered at the Earth's center of mass. The z^i axis is parallel to the Earth's polar axis of rotation, while the x^i axis points in the direction of vernal equinox. Finally, the y^i axis points 90°

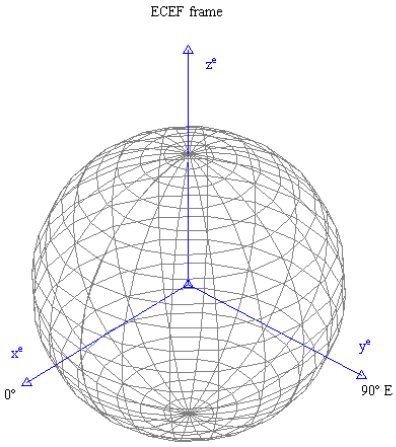
degrees ahead (taken in the direction of Earth's rotation) of the x^i axis, positioned in such a way to make the $(x^i y^i z^i)$ system a right-handed orthogonal set.

The ECI frame can be considered inertial for all small unmanned aircraft applications, ignoring the Earth's acceleration in its elliptic orbit around the Sun. This is the frame with respect to which the inertial sensors (accelerometers and gyroscopes) make measurements.

It should be noted that the Earth rotates with respect to the ECI system with angular velocity ω_{ie}.

1.2.2 Earth Centered Earth Fixed

The Earth Centered Earth Fixed frame (ECEF) is centered at the Earth's center of mass. The z^e axis is parallel to the Earth's polar axis of rotation, like in the ECI frame. The x^e axis points to the intersection of the equator and the prime meridian. The y^e axis precedes the x^e axis by 90° degrees in the direction of Earth's rotation, making the ECEF frame a right-handed orthogonal set. As a result, the axes of the ECEF frame are fixed with respect to the Earth and they rotate with respect to the ECI frame.



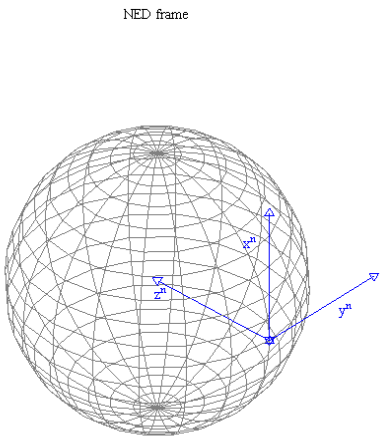
The ECEF is the frame in which the position of the aircraft is represented. Position in the ECEF frame can be represented using either rectangular coordinates (x^e, y^e, z^e) or curvilinear coordinates (ϕ, λ, h) which will be discussed in the earth modeling section 1.4.

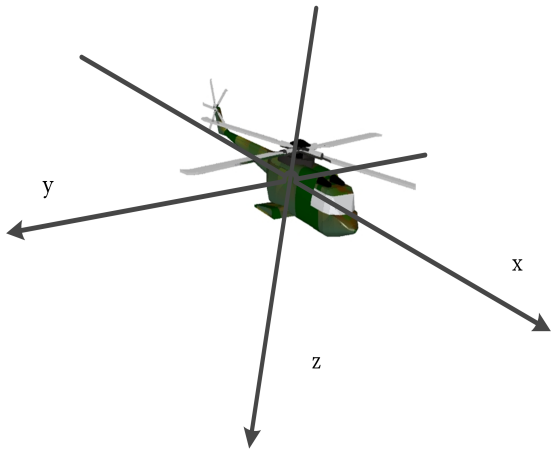
1.2.3 Local navigation frame (NED)

The local frame (abbreviated n or NED for North-East-Down) is centered at the vehicle's center of mass. Its z^n axis is normal to the reference ellipsoid (see section 1.4) and that direction is known as "Down". The x^n axis ("North") points to the Earth's North pole and the y^n axis ("East") completes the right-handed orthogonal set. The NED frame makes a flat earth approximation for the point of interest and it is of great importance as it is used to define the aircraft's attitude.

1.2.4 Body frame

The body frame is centered at the vehicle's center of mass. Axis x^b points forward, axis z^b down and axis y^b points to the right, completing the orthogonal set. Although the definition of forward, down, etc. takes into account the usual direction of movement, this is rather obvious for the type of vehicles considered in this text.





1.2.5 Frame notation for vectors

The notation used in [3] is followed here to allow for the maximum possible clarity in all the vector operations / transformations presented.

Vectors are denoted by x^c_{ab}, where the right subscript b refers to the frame whose motion is described, the left subscript a is the frame with respect to which this motion is described and the superscript c is the frame in whose axes system this motion is described (resolving frame).

The left subscript (reference frame) may be omitted in the upcoming chapters when the reference frame is obvious.

1.2.6 Transformation matrices

The coordinate transformation matrices are used to transform the resolving axes of a vector to another set. They are denoted by C^{to}_{from}, where the subscript refers the original frame and the superscript to the resulting frame.

For example,

$$x^b = C^b_a x^a \tag{1.1}$$

Coordinate transformation matrices have the following properties for all frames a and b:

$$\text{a) } (C^b_a)^T = (C^b_a)^{-1} = C^a_b \text{ (orthogonality)} \tag{1.2}$$

$$\text{b) } C^b_a C^a_b = I \tag{1.3}$$

$$\text{c) } \det C_a^b = I \tag{1.4}$$

1.3 Attitude & attitude kinematics

The aircraft's attitude can now be defined as the orientation of the body frame with respect to the NED frame. There are various attitude representations, namely Euler angles, Rodriguez parameters, quaternions, etc. [13]. Each of these representations can be used to assemble the rotation matrix which is defined as the transformation matrix R_{NED}^b. The rotation matrix is unique and global, meaning that there is only one rotation matrix for any two given frames and it is defined for any possible attitude.

In the two next sections, the Euler angles and the attitude quaternion will be presented respectively. The Euler angles provide a very simple and intuitive attitude representation, which exhibits, though, non uniqueness and non globality. The quaternion is an efficient global attitude representation and will be used in the estimator equations in the upcoming chapters.

1.3.1 Euler angles

The Euler angles attitude representation is based on three successive rotations that, if taken with the right order, will eventually align the NED frame with the body frame [2], [13].

Yaw rotation

First of all, the NED system is rotated by ψ radians around its Down axis until the original North axis coincides with the projection of the x^b axis on the local frame. The resulting frame will be called the intermediate frame I_1. The corresponding transformation matrix is

$$R_{NED}^{I_1} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.5}$$

Pitch rotation

Secondly, the I_1 frame is rotated by θ radians around its "East" axis, until its original North axis coincides with x^b. For convenience, this frame will be called I_2. The rotation matrix is

$$R_{I_1}^{I_2} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{1.6}$$

Roll rotation

Finally, the I_2 frame is rotated around the common "North" – x^b axes by ϕ radians until it fully aligns with the body frame. The roll rotation matrix is

$$R_{I_2}^{b} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{1.7}$$

Rotation matrix assembly

Multiplying the transformation matrices in the above given order yields the rotation matrix from the local to the body frame,

$$R^b_{NED} = R^b_{I_2} R^{I_2}_{I_1} R^{I_1}_{NED} \tag{1.8}$$

After substituting equations (5) (6) and (7) into (8) we get

$$R^b_{NED} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\sin\theta & -\sin\theta \\ -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \tag{1.9}$$

The inverse transformation matrix is given by the transpose,

$$R^{NED}_b = \left(R^b_{NED}\right)^T \tag{1.10}$$

The set (ϕ, θ, ψ) are called the roll, pitch and yaw angles respectively and they constitute the Euler angles attitude representation.

Time propagation

Given the angular rates about the body axes $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$, the time propagation of the Euler angles is described by the nonlinear system

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{1.11}$$

Issues

Apart from the obvious angle periodicity issue which can be solved in a practical manner (some limits must be imposed on the angles to avoid overlapping), the Euler angles attitude representation exhibits two major drawbacks.

- There is a singularity at $\pm 90^\circ$ pitch, where the roll and yaw angles cannot be defined; thus, the Euler angles is not a global attitude representation. Furthermore, at the same attitude at $\pm 90^\circ$ pitch, the two terms of the time propagation system (1.11) that involve $\sec\theta$ become infinite. If the vehicle approaches $\pm 90^\circ$ pitch, this can lead to numerical issues or even to a software crash unless it is taken care of beforehand.

- The continuous system (1.11) is highly nonlinear and that causes large errors in the discretization process, which is necessary in order to use the time propagation equation in a Kalman Filter (see Chapter 2).

The above drawbacks limit the use of Euler angles to applications where the vehicle in question doesn't operate near the $\pm 90^\circ$ pitch range (to avoid the singularity points) and to scenarios where the attitude changes slowly over time (to alleviate the discretization errors).

1.3.2 Quaternion attitude representation

The quaternion is a singularity-free attitude representation that allows for much more accurate and robust attitude mechanization in a state estimation framework in comparison with Euler angles.

Definition and operations

A quaternion q consists of a scalar part, q_0, and a three dimensional vector part $q_v = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$,

$$q = \begin{bmatrix} q_0 \\ q_v \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{1.12}$$

An abstract rotation of $\|\phi\|$ radians around axis $\frac{\phi}{\|\phi\|}$ can be represented by the quaternion,

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\|0.5\phi\| \\ \frac{\sin\|0.5\phi\|}{\|\phi\|}\phi \end{bmatrix} \tag{1.13}$$

If the quaternion is used to represent attitude, the rotation vector ϕ aligns the local frame with the body frame. However, the use of the quaternion is not restricted to the attitude case as it can represent any rotation.

It is easy to notice that, due to its definition, the quaternion has a unit length constraint,

$$\|q\| = 1 \tag{1.14}$$

The inverse quaternion is obtained by taking the negative of the vector part, as

$$q^{-1} = \begin{bmatrix} q_0 \\ -q_v \end{bmatrix} = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} \tag{1.15}$$

The quaternion multiplication between two quaternions a and b is given by [2]

$$a \otimes b = \begin{bmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & -a_3 & a_2 \\ a_2 & a_3 & a_0 & -a_1 \\ a_3 & -a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{1.16}$$

Successive rotations can be conveniently defined through quaternion multiplication, as

$$q_1^3 = q_2^3 \otimes q_1^2 \tag{1.17}$$

where the subscript and the superscript in each case refer to the original and the resulting frame of the rotation that each quaternion represents.

As the quaternion is not a vector quantity (due to the unity length constraint), quaternions cannot be added or subtracted, since the constraint will be violated. The quaternion multiplication operation should be used instead.

Rotation matrix assembly

The rotation matrix from the local frame to the body frame is

$$R^b_{NED} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_0 q_1 + q_2 q_3) \\ 2(q_0 q_2 + q_1 q_3) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{1.18}$$

and the rotation matrix from the body to the local frame is the transpose of (1.18),

$$R_b^{NED} = \left(R^b_{NED}\right)^T \tag{1.19}$$

Time propagation

The kinematic propagation of the attitude quaternion, when the angular rates about the body frame axes $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ are known, is given by

$$\dot{q} = \frac{1}{2}\Omega_b q = \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} q \tag{1.20}$$

Equation (1.20) is linear in terms of q, providing thus a significant advantage over the equivalent propagation equation of the Euler angles (1.11). However, despite the more robust mathematical formulation of the attitude quaternion, it is not as intuitive to humans as the Euler angles. For that reason, the quaternion is usually transformed into the equivalent Euler angles for the purpose of visualization.

1.4 Earth modeling, position representation & propagation

In the WGS 84 model [23] followed in this text, the Earth's surface[1] is approximated by an oblate spheroid. The oblate spheroid is an ellipsoid shape with rotational symmetry whose minor axis is the axis of rotation of the Earth, z^e. The ellipsoid's semi major axis is the Earth's equatorial radius R_0 and the semi major axis is the Earth's polar radius R_p.

The eccentricity is given by

$$e = \sqrt{1 - \frac{R_p^2}{R_o^2}} \tag{1.21}$$

and the flattening by

$$f = \frac{R_0 - R_p}{R_0} \tag{1.22}$$

The values of the above parameters are given in [23].

Any point on or near the surface of the Earth can be described by its geodetic latitude p_ϕ, its longitude p_λ and its height h.

- The geodetic latitude p_ϕ is the angle formed by the normal to the ellipsoid originating from the center of the body frame and the equatorial plane.
- The longitude p_λ is the angle between the prime meridian plane and the plane formed by the Earth's z^e axis and the point of interest.
- The ellipsoidal height h is the distance of the point to the ellipsoid's surface.

1 The term surface here corresponds to the approximate sea level.

More details on the curvilinear coordinates can be found in more detailed texts, [3], [2].

On any point on the ellipsoid's surface, two radii of curvature are defined,

- The meridian radius of curvature $R_N(\phi)$, which is related to the rate of change of the latitude along a meridian [3],

$$R_N(\phi) = \frac{R_p}{\left(1 - e^2 \sin^2 \phi\right)^{\frac{3}{2}}} \tag{1.23}$$

- The transverse radius of curvature $R_E(\phi)$, which is related to the rate of change of latitude on the surface normal to a meridian [3],

$$R_E(\phi) = \frac{R_0}{\sqrt{1 - e^2 \sin^2 \phi}} \tag{1.24}$$

The rectangular and the curvilinear coordinates of a point on the ECEF frame are related by

$$x^e = \begin{bmatrix} (R_E(\phi) + h) \cos\phi \cos\lambda \\ (R_E(\phi) + h) \cos\phi \sin\lambda \\ \left[(1 - e^2) R_E(\phi) + h\right] \sin\phi \end{bmatrix} \tag{25}$$

Using the chain rule, the velocity of a point in the rectangular coordinates of the ECEF frame is

$$v_e^e = \frac{\partial x^e}{\partial \phi}\frac{\partial \phi}{\partial t} + \frac{\partial x^e}{\partial \lambda}\frac{\partial \lambda}{\partial t} + \frac{\partial x^e}{\partial h}\frac{\partial h}{\partial t} = \begin{bmatrix} \frac{\partial x^e}{\partial \phi} & \frac{\partial x^e}{\partial \lambda} & \frac{\partial x^e}{\partial h} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} \tag{1.26}$$

Calculating the partial derivatives of x^e yields

$$\left[\begin{array}{ccc} \frac{\partial x^e}{\partial \phi} & \frac{\partial x^e}{\partial \lambda} & \frac{\partial x^e}{\partial h} \end{array} \right] = R_n^e \left[\begin{array}{ccc} R_N(\phi)+h & 0 & 0 \\ 0 & (R_E(\phi)+h)\cos\phi & 0 \\ 0 & 0 & -1 \end{array} \right] \tag{1.27}$$

where R_n^e is the coordinate transformation matrix from the local to the ECEF frame (see [2]).

The North – East – Down velocity is now obtained by ([2],[3])

$$v_{NED} = v_e^n = R_e^n v_e^e = \left[\begin{array}{c} v_N \\ v_E \\ v_D \end{array} \right] = \left[\begin{array}{c} (R_N(\phi)+h)\dot{\phi} \\ \cos\phi(R_E(\phi)+h)\dot{\lambda} \\ -\dot{h} \end{array} \right] \tag{1.28}$$

Inverting equation (1.28) gives the propagation equation of the curvilinear ECEF coordinates as a function of the NED velocity,

$$\left[\begin{array}{c} \dot{\phi} \\ \dot{\lambda} \\ \dot{h} \end{array} \right] = \left[\begin{array}{c} \frac{v_N}{R_N(\phi)+h} \\ \frac{v_E}{(R_E(\phi)+h)\cos\phi} \\ -v_D \end{array} \right] \tag{1.29}$$

1.5 Typical small UAV sensors

Depending on the aircraft's mission, an unmanned aircraft may carry many different sensors. However, any UAV must be able to sustain itself in the air and fly over specific waypoints, as required for each mission objective; there is a requirement, thus, for the aircraft to know its position, attitude and velocity at any time, regardless of the mission. Separating the various sensors usually found on small UAVs to sensors directly related to the UAV's flight and to sensors related to specific missions, all UAVs carry the following

minimum sensor payload, to enable automatic flight and increase their autonomy level:

- Inertial Measurement Unit (IMU)
- Global Navigation Satellite System (GNSS) receiver for UAVs that operate outdoors
- Radio altimeter or range finder
- Cameras used for navigation
- Compass

As this thesis focuses on small UAVs that operate outdoors, we will not discuss further the use of cameras or any other positioning service that is tailored for indoors use or, in general, for use in GNSS denied environments.

1.5.1 Inertial Measurement Unit (IMU)

Inertial Measurement Units (IMUs) contain accelerometers and gyroscopes, measuring accelerations and angular velocities respectively in the inertial body frame.

IMU operation

Ideally, integration of the IMU output could provide the quantities required for control (attitude, velocity and position). However this is not the case, as the IMU output is rather noisy and therefore there would be an ever growing navigation error. A position / velocity sensor with bounded error must be used to contain the IMU growing drifting errors. This is the purpose of state estimation for small UAVs – to augment the IMU measurements with a lower bandwidth / bounded error sensor, like the GNSS receiver. Standalone IMU navigation (also called dead – reckoning) is still used in some applications, as in submarine navigation, where there is no easy way to compensate for the IMU errors, in indoors UAVs where there is no GNSS coverage and in commercial aircraft where very accurate IMUs are

used and still there in a drift of a couple of kilometers per hour in the position output. In UAV applications, due to the performance and error characteristics of the IMUs used, standalone IMU operation is not an option.

IMUs used in small UAVs are usually of MEMS (microelectromechanical systems) type and they feature a strapdown configuration. Strapdown IMUs have three accelerometers and three gyroscopes fixed in an orthogonal set of axes. The combination of the strapdown configuration and the MEMS type sensors results in very low size (few centimeters), weight (few grams) and cost IMUs and that is their reason for success in the small UAV sector.

IMU errors

This particular combination, however, can also cause many accuracy issues. First of all, the three strapdown axes never actually form a truly orthogonal set; there is always some misalignment that may need to be taken care of [2]. Secondly, the MEMS nature of the sensors themselves can cause very poor results in the sensor output, particularly in terms of growing biases, scale factors, excessive noise, etc. Axes misalignment, biases and scale factors can be dealt with during factory calibration of the IMU and also – in some units – during operation using a closed loop system with temperature control. These error sources can be dealt with in the state estimation algorithm as well, while noise can be filtered out only in a state estimation framework, utilizing kinematic models of the vehicle of interest and some related measurements with bounded error characteristics.

Measurable quantities

As noted above, an IMU's gyroscope measures the angular velocities of the body frame with respect to the inertial frame. Although the measurements are made around the sensor's sensitive axes and not the actual vehicle's body axes, it is reasonable to assume that with careful placement of the IMU, the measurements made correspond to the angular

velocities around the body frame axes,

$$\begin{bmatrix} \omega_{g1} \\ \omega_{g2} \\ \omega_{g3} \end{bmatrix} \cong \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{1.30}$$

We ignore the rotation of the Earth and the curvilinear motion of the origin of the body frame [3], as these quantities are too small when compared with the noise characteristics of MEMS gyroscopes.

As far as acceleration is concerned, it should be noted that the IMU's accelerometers do not measure the acceleration itself a_{ib}^b, but the specific force f_{ib}^b along the instrument's sensitive axes instead,

$$f_{ib}^b = a_{ib}^b - g_{ib}^b \tag{1.31}$$

where g_{ib}^b is the gravitational force along the body axes that can be calculated as a function of curvilinear position using the WGS84 model [23].

Apart from sensor noise, the IMU output can exhibit significant biases [3]. The accelerometer bias b_a and the gyroscope bias b_g can be modeled using slowly time varying white noise processes, as ([17], [18])

$$\dot{b}_a = n_{abp} \tag{1.32}$$

$$\dot{b}_g = n_{gbp} \tag{1.33}$$

where n_{abp} and n_{gbp} are Gaussian random variables with zero mean and covariance $Q_{n_{abp}}$ and $Q_{n_{gbp}}$ respectively.

1.5.2 Global Navigation Satellite System (GNSS)

Global Navigation Satellite Systems provide positioning and timing service for targets on or near Earth. They consist of three segments, namely,

- the **space segment**, which refers to a constellation of satellites orbiting the earth in geosynchronous orbits. Each satellite transmits a message containing, among others, its position and the time of transmission. The number and the orbits of satellites is chosen such as there is continuous global coverage and positioning service for all points on Earth.

- the **user segment**, which refers to receiver devices that receive and decode the signals transmitted by GNSS satellites. A GNSS receivers is able to process the information contained in the GNSS messages and eventually calculate its position on earth and the current time, by solving a nonlinear system of equations (described bellow).

- the **control segment** that is operated by the owner of each GNSS constellation and is concerned with maintaining the system's fidelity and uninterrupted operation of the system.

GNSS operation

The GNSS receiver acquires and decodes the signals transmitted by each satellite, in order to obtain the time of signal transmission t_{st}, the carrier signal frequency f_{ca} and the Doppler frequency shift Δf_{ca}. The signals also contain information about the satellite's

orbit, so its position at the time of signal transmission, $p_s(t_{st})$, can be calculated.

Given the above data, the pseudorange for the j^{th} satellite, which is related to the distance the signal has traveled from the satellite to the receiver's antenna, can be defined as

$$\rho_j = c(t_{sa} - t_{st,j}) \tag{1.34}$$

where c is the speed of sound and t_{sa} is the time of signal arrival (which is unknown, yet).

Furthermore, the pseudorange rate, which is related to the receiver's and the satellite's relative velocity is

$$\dot{\rho}_j \approx -\frac{c}{f_{ca,j}} \Delta f_{ca,j} \tag{1.35}$$

The receiver's position p^e_{ea} in the ECEF frame is now related to the above measurable quantities using the nonlinear equation

$$\rho_j = \sqrt{\left[C^I_e(t_{st,j}) p^e_{es,j}(t_{st,j}) - p^e_{ea}(t_{sa})\right]^T \left[C^I_e(t_{st,j}) p^e_{es,j}(t_{st,j}) - p^e_{ea}(t_{sa})\right]} + \delta\rho_{rc} + \delta\rho_{r,j} \tag{1.36}$$

where $\delta\rho_{rc}$ is the receiver's clock offset and $\delta\rho_{r,j}$ is a residual error. In the above equation, the satellite's position p^e_{es} is calculated using the transmission time t_{st} and the known orbit characteristics.

The transformation matrix C^I_e is used to compensate for the Earth's rotation during the signal transmission from the satellite to the receiver.

Similarly, the pseudorange rate is related to receiver's velocity v^e_{ea} in the ECEF frame through

$$\dot{\rho}_j = {u^e_{as,j}}^T \left[C^I_e(t_{st,j}) \left(v^e_{es,j}(t_{st,j} + \Omega^e_{ie} p^e_{es,j}(t_{st,j})\right) - (v^e_{ea}(t_{sa}) + \Omega^e_{ie} p^e_{ea}(t_{sa}))\right] + \delta\rho_{rc} + \delta\rho_{r,j} \quad (1.37)$$

where $u^e_{as,j}$ is the unit vector from the j^{th} satellite to the receiver's antenna and the Ω^e_{ie} is used to compensate for the Earth's rotation.

In equation (1.36) there are four unknowns, the three dimensional position p^e_{ea} and the time of signal arrival t_{sa}. Therefore, in order to solve it, at least data from four satellites are needed. The same is true for the velocity equation (1.37).

GNSS and IMU integration

There are two ways in literature to handle GNSS equations (1.36) and (1.37) is a state estimation framework where inertial sensors are utilized as well [4], [3]. The first is referred to as "tight integration" and it involves using equations (1.36) and (1.37) inside a Kalman Filter (Chapter 2) that estimates all the above unknowns, along with all the other kinematic quantities of interest. The second approach or the "loose integration" scheme lets the GNSS receiver do this job and utilizes the computed position and velocity output. Tight integration can provide better performance but complicates the estimation algorithm very much and increases its computational cost considerably. Loose integration can provide results with acceptable accuracy for most small UAV applications, especially when used in an augmented GNSS environment (section 1.5.2.2) and this is the approach used in this text.

GNSS errors

Many error sources that contribute to GNSS positioning and velocity errors, of which the most prominent are [4]

- signal propagation delays, which result in local perturbation in the ionosphere and

the atmosphere that cause delays in the transmission of the GNSS signals

- clock errors, which refer to inconsistencies in the clocks used onboard the GNSS satellites
- receiver errors, which arise from the hardware used in GNSS receivers and are more common in lower cost receiver solutions
- multipath, which is caused by the local environment of the GNSS receiver that may reflect or degrade the GNSS signals coming from the satellite constellation
- selective availability, which consists of random errors in the GNSS system imported deliberately by the operating authority to hinder the use of the GNSS positioning service. Selective availability was used in the initial years of the GPS but it was disabled later in the 2000s.

GNSS today

Currently, two GNSS systems are fully operational, providing global coverage; the Global Positioning System (GPS), operated by the United States, and the GLObal NAvigation Satellite System (GLONASS), operated by the Russian Federation. Due to the difference in the Earth models and the time used in the two systems, they are not interoperable with each other as each system will calculate a different position for the same point on Earth. This text will only deal with the Global Positioning System and WGS84 Earth modeling. Dual GNSS implementations are, though, possible and rather interesting from a safety and redundancy point of view, however the appropriate transformation from each datum and vice versa must be taken care of, as well as the time difference, in the integration software [3].

1.5.2.1 Global Positioning System (GPS)

The Global Positioning System (GPS) is the most widely used GNSS service worldwide. It

is designed and operated by the United States.

The accuracy of GPS for positioning is in the range of a couple of meters for the majority of scenarios (with selective availability turned off). Receivers with quality hardware in standalone operation (without augmentation) can achieve a 3 to 5 meters 95% positioning accuracy.

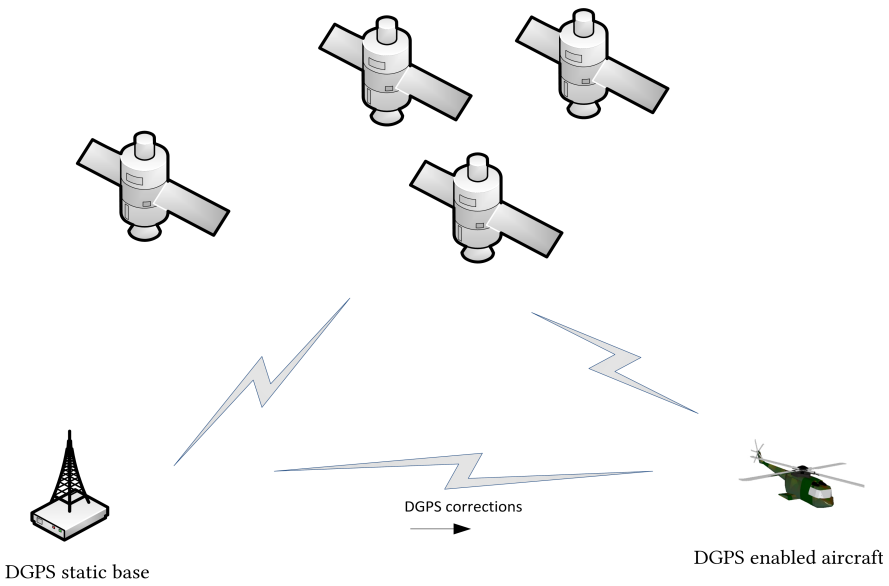
1.5.2.2 GPS augmentation

The performance of any GNSS system can be greatly enhanced if used together with a GNSS augmentation service. GNSS augmentation refers to various methods that attempt to compensate for the GNSS errors on a local basis. All error sources listed previously can be compensated with the various GNSS compensation schemes, except for receiver errors and multipath. GNSS augmentation can increase the accuracy of GNSS positioning to about one meter or even more.

Some GNSS augmentation schemes are tailored to cover a large area and they operate through satellites or ground stations that broadcast correction messages that contain information about the errors for each satellite (in particular, they contain corrections for each pseudorange for the visible / usable satellites). The rest of the GNSS augmentation techniques cover smaller areas of interest. In any case, the GNSS receiver used must be able to import and decode the correction messages send by the augmentation stations.

A GNSS augmentation technique which is suitable for the operation of small UAVs is Differential GPS (DGPS). DGPS, which is a local augmentation technique for GPS is based on the principle of one or more special stationary receivers that given their known position, they solve the inverse of equation (1.36) in order to obtain the pseudorange and clock residuals. At each iteration, the results are transmitted to the DGPS capable moving receivers that incorporate them to their calculations. DGPS offers the maximum possible

accuracy for positioning (in the decimeter range) and one station can cover an area of a couple of kilometers around it (of course, there is a need for an equally capable communication link to transmit the correction results to the nearby nonstationary GPS receivers).



1.5.3 Compass – Magnetometer

A compass measures the aircraft's magnetic heading. The magnetic heading ψ_m is different from the true heading ψ (also referred to as yaw) which is the bearing to the Earth's North pole and is the quantity of interest for use in navigation. The difference between the true and the magnetic heading is the magnetic declination δ, and it is heavily influenced from local fluctuations in the Earth's magnetic field.

The World Magnetic Model [24] provides a global approximation of the Earth's magnetic field components and other relevant quantities, including the magnetic declination, as a function of position and time (the date, actually), $\delta(p, t)$.

Using the convention that the magnetic declination is positive when the direction to which the compass points (the magnetic North) is west of the true North, the true heading is given by

$$\psi = \psi_m + \delta(p, t) \tag{1.38}$$

If a three axis magnetometer is used instead of a compass, as it is the case in many small UAVs that include a magnetometer inside the Inertial Measurement Unit, the magnetic heading is given by

$$\psi_m = \arctan 2(M_y, M_x) \tag{1.39}$$

where M_y and M_x are the magnetic field measurements along the body frame's y and x axes respectively (we assume that the magnetometer's axes are aligned with the respective body frame axes).

The magnetic heading output, if obtained as described bellow, should not be used in cases when the roll and / or pitch angle exceeds a value of about 20° as the Down component of the Earth's magnetic field will be accounted for in the measurements along the body frame axes x and y, degrading the result.

Chapter 2

State Estimation Theory Review

This chapter is concerned with the mathematical theory of state estimation. The objective of state estimation is to approximate the state of a dynamical system, taking into account the modeled system dynamics, known system inputs and any available noisy measurements. State estimation for linear systems is considered in section 2.2 where the linear Kalman Filter is described. In Section 2.3 nonlinear systems are discussed; the Extended Kalman Filter and the Unscented Kalman Filter are presented in sections 2.3.1 and 2.3.2 respectively.

2.1 System description

Due to the fact that state estimation algorithms are almost always implemented in real time as an integral part of digital control systems, here we only refer to their discrete time formulation.

Let x_k be an $(n \times 1)$ state vector to be estimated, with its dynamics given by the nonlinear system f_k,

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \tag{2.1}$$

where u_k is a $(p \times 1)$ vector of known system inputs and w_k is a $(r \times 1)$ white Gaussian noise vector with covariance Q_k accounting for process noise. In most state estimation solutions, the process noise accounts for unmodeled system dynamics. Equation (2.1) is referred to as the process model, in the state estimation context.

We suppose that a set of measurements are available that are related to the system's state through the nonlinear measurement equation

$$z_k = h_k(x_k, v_k) \tag{2.2}$$

where z_k is an $(l \times 1)$ measurement vector and v_k is an $(l \times 1)$ vector of white Gaussian noise with covariance R_k accounting for sensor noise.

State estimation algorithms provide an estimate of the state vector, noted $\hat{x}_k$, using the available measurements z_k and taking into account the systems dynamics f_k, as given by (2.1). As [5] points out, this estimate should be unbiased ($\mathbb{E}[\hat{x}_k]$ should equal $\mathbb{E}[x_k]$), of minimum error variance ($\min\left[\mathbb{E}[\tilde{x}_k^2] - (\mathbb{E}[\tilde{x}_k])^2\right]$ where $\tilde{x}_k$ is the error) and consistent (converging to the true value with more measurements). In the case of linear systems, the above three requirements can be fulfilled through an optimality criterion, yielding the optimal Kalman Filter. In nonlinear systems, there is no way to come up with an optimal estimator as the statistics a random variable undergoing nonlinear transformations cannot be approximated by its mean and covariance only; we rely on certain approximations instead.

2.2 Linear state estimation – the Kalman Filter

The Kalman Filter has been proposed by R.E. Kalman in 1960 in [9]. Other references are [5], [6] and [10]. The development of this section follows the Kalman Filter derivation as

presented in [5].

Equations

In the special case where the process and measurement models are linear, they can be written respectively as

$$x_k = F_{k-1}x_{k-1} + L_{k-1}u_{k-1} + w_{k-1} \tag{2.3}$$

and

$$z_k = H_k x_k + v_k \tag{2.4}$$

where F_{k-1} is an $(n \times n)$ matrix, L_{k-1} is an $(n \times p)$ input coupling matrix and H_k is an $(l \times n)$ measurement matrix.

A basic assumption of the Kalman filter is that the process and the measurement noise are uncorrelated,

$$\mathbb{E}[w_k v_j^T] = 0, \forall k, j \in \mathbb{N} \tag{2.5}$$

In order for the estimation process to be viable given the time and memory restrictions imposed by real time applications, a solution of linear recursive form is sought, namely

$$\hat{x}_k(+) = K'_k \hat{x}_k(-) + K_k z_k \tag{2.6}$$

where K'_k and K_k are matrices to be determined. The a priori $(-)$ symbol refers the value of the respective quantity (here the state vector) before a measurement is taken, while a posteriori $(+)$ is used to refer to the value after a measurement has been taken and

processed. It should be noted that equation (2.6) is a linear combination of the a priori state and the newest measurement – there is no need to store and process past states and measurements.

The state estimate can be written as a sum of the real value and an error term,

$$\hat{x}_k = x_k + \tilde{x}_k \tag{2.7}$$

Using (2.6), equation (2.7) becomes

$$x_k(+) + \tilde{x}_k(+) = K'_k \left(x_k(-) + \tilde{x}_k(-)\right) + K_k z_k \tag{2.8}$$

Now, incorporating the measurement model (2.4) we get an expression for the estimation error,

$$\tilde{x}_k(+) = (K'_k + K_k H_k - I)x_k + K'_k \tilde{x}_k(-) + K_k v_k \tag{2.9}$$

Due to the requirement for an unbiased estimate, $\mathbb{E}[\tilde{x}_k(+)] = 0$ and $\mathbb{E}[\tilde{x}_k(-)] = 0$. Since also $\mathbb{E}[v_k] = 0$ (due to the definition of v_k as white Gaussian noise), we get

$$K'_k + K_k H_k - I = 0 \tag{2.10}$$

The estimate and its error become, respectively,

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k(z_k - H_k \hat{x}_k(-)) \tag{2.11}$$

$$\tilde{x}_k(+) = (I - K_k H_k)\tilde{x}_k(-) + K_k v_k \tag{2.12}$$

Defining the covariance of the estimate as the expected value of the estimation error,

$$P_k(+) = \mathbb{E}[\tilde{x}_k(+)\tilde{x}_k^T(+)] \tag{2.13}$$

we obtain the following expression for the a posteriori covariance

$$\begin{aligned} P_k(+) =& \mathbb{E}\{(I - K_k H_k)\tilde{x}_k(-)[\tilde{x}_k(-)^T(I - K_k H_k)^T + v_k^T K_k^T] \\ &+ K_k v_k(\tilde{x}_k(i)^T(I - K_k H_k)^T + v_k^T K - k^T)) \end{aligned} \tag{2.14}$$

Taking into account that the estimation error and the measurement noise are uncorrelated,

$$\mathbb{E}[\tilde{x}_k(-)v_k^T] = \mathbb{E}[v_k \tilde{x}_k(-)^T] = 0 \tag{2.15}$$

The optimal Kalman gain matrix K_k is obtained through the minimization of the cost function

$$J_k = \mathrm{tr}(P_k(+)) \tag{2.16}$$

Setting

$$\frac{\partial J_k}{\partial K_k} = 0 \tag{2.17}$$

yields

$$K_k = P_k(-)H_k^T[H_k P_k(-)H_k^T + R_k]^{-1} \tag{2.18}$$

The a posteriori mean is

$$\hat{x}(+) = \hat{x}(-) + K_k[z_k - H_k \hat{x}_k(-)] \tag{2.19}$$

An alternative, simpler form for K_k is suggested by [5],

$$K_k = P_k(+)H_k^T R_k^{-1} \tag{2.20}$$

Using the above expression for K_k, the a posteriori covariance is given by

$$P_k(+) = (I - K_k H_k)P_k(-) \tag{2.21}$$

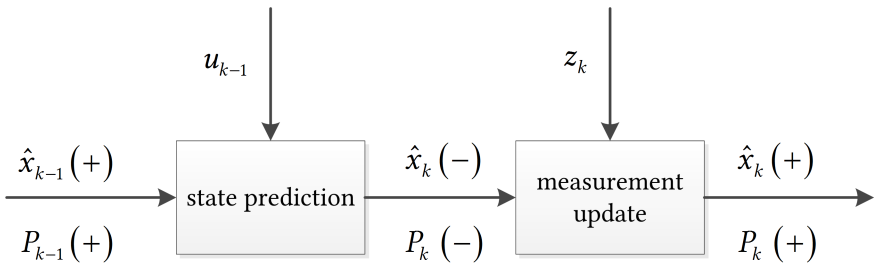
Finally, by using the process model (2.1) it is possible to obtain the a priori (predicted) state and covariance,

$$\hat{x}_k(-) = F_{k-1}\hat{x}_{k-1}(+) + L_{k-1}u_{k-1} \tag{2.22}$$

$$P_k(-) = F_{k-1}P_{k-1}(+)F_{k-1}^T + Q_{k-1} \tag{2.23}$$

Summary

Summarizing, the Kalman Filtering process for the optimal state estimation of linear systems can be broken up in two parts, prediction and correction. For a Gaussian distribution, the mean and the covariance uniquely determine the distribution itself. The Kalman Filter provides a way to transform the distribution using the linear process and measurement models; due to the linearity of the equations, the distribution remains Gaussian and the state estimation solution is optimal and unique.



In the prediction step, the process model is used to propagate the state vector $\hat{x}_{k-1}(+)$ into

time yielding $\hat{x}_k(-)$, using the known system dynamics F_{k-1} and the deterministic inputs u_{k-1}. Its covariance $P_k(-)$ is also calculated, using the previous covariance value $P_{k-1}(+)$, the system dynamics F_{k-1} and the known process noise covariance matrix Q_{k-1}. The prediction step can be repeated as many times as a prediction of the state is needed, although in any practical application the presence of process noise will render the prediction highly inaccurate in a number of iterations if no measurements are available.

The measurement update step, which is invoked after the prediction step and whenever there are measurements z_k available, involves the computation of the Kalman gain matrix K_k which is subsequently used to correct the predicted state estimate and covariance, yielding $\hat{x}_k(+)$ and $P_k(+)$.

2.3 Nonlinear state estimation

In the nonlinear case, the system is described as in section 2.1. Now the probability distribution of the state vector is not always Gaussian, thus it cannot be described fully by its first two moments (mean and covariance). Higher order moments may have significant effects on the distribution. Furthermore, the process and measurement models are not necessarily in closed form, so they may have to be approximated to be used in a state estimation framework. In any case, there is no unique, optimal solution to the nonlinear state estimation problem.

Two solutions for nonlinear state estimation are presented here. The Extended Kalman Filter, which is a derivative based method that linearizes the process and measurement models, attempting to approximate the respective nonlinear transformations, and the Unscented Kalman Filter, which follows a different approach by approximating the probability distribution function before and after the nonlinear transformations, applying the nonlinear models in a black box way.

2.3.1 Extended Kalman Filter

This section follows the Extended Kalman Filter (EKF) presentation found in [5] and [7]. We consider a system with process and measurement models of the form

$$x_k = f_{k-1}(x_{k-1}) + w_{k-1} \tag{2.24}$$

$$z_k = h_k(x_k) + v_k \tag{2.25}$$

where f_k and h_k are nonlinear functions of the system's state, while w_k and v_k are white Gaussian noise vectors with covariance Q_k and R_k respectively.

Note that the system formulation used here to present the EKF doesn't include known inputs. Inclusion of known inputs u_k will be covered in the next section discussing the Unscented Kalman Filter.

The prediction step of EKF involves propagating the state using the nonlinear process model,

$$\hat{x}_k(-) = f_{k-1}(\hat{x}_{k-1}(+)) \tag{2.26}$$

and predicting the measurement that would correspond to the propagated state $\hat{x}_k(-)$,

$$\hat{z}_k = h_k(\hat{x}_k(-)) \tag{2.27}$$

The nonlinear process model now is linearized around the previous estimate,

$$F_{k-1}^{[1]} = \frac{\partial f_k}{\partial x} \Big|_{x=\hat{x}_{k-1}(+)} \tag{2.28}$$

The superscript in $F_{k-1}^{[1]}$ indicates the order of the linearization. It is also possible to derive the EKF using higher order linearizations as in [5].

The a priori covariance can now be calculated, concluding the prediction step,

$$P_k(-) = F_{k-1}^{[1]} P_{k-1}(+) {F_{k-1}^{[1]}}^T + Q_{k-1} \tag{2.29}$$

If measurements are available, the measurement model is similarly linearized around the a priori state mean, yielding the Jacobian matrix

$$H_k^{[1]} = \frac{\partial h_k}{\partial x} \Big|_{x=\hat{x}_k(-)} \tag{2.30}$$

in order to calculate the Kalman gain

$$K_k = P_k(-) H_k^{[1]T} [H_k^{[1]} P_k(-) H_k^{[1]T} + R_k]^{-1} \tag{2.31}$$

It should be noted here that the Kalman gain equation is rather similar to that of the linear Kalman Filter (2.18), as the Jacobian of the measurement model $H_k^{[1]}$ takes the place of the measurement matrix H_k.

Again, in a similar way with the Kalman Filter, the a posteriori mean and covariance are given by

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k(z_k - \hat{z}_k) \tag{2.32}$$

$$P_k(+) = (I - K_k H_k^{[1]}) P_k(-) \tag{2.33}$$

2.3.2 Unscented Kalman Filter

In contrast to the EKF, the Unscented Kalman Filter (UKF) attempts to approximate the statistics of the random variable before and after the nonlinear transformation, instead of approximating the nonlinear transformations themselves through linearizations. It is a derivative free method, as no calculation of Jacobian matrices is required. The most comprehensive reference for the UKF is the review [11] prepared by its original authors.

2.3.2.1 State prediction

The UKF is based on a deterministic sampling principle that utilizes a Cholesky decomposition of the state covariance matrix. Taking the state vector with mean $\hat{x}_k$ and covariance P_{xx}, a set of weighted state vectors called sigma points are generated around $\hat{x}_k$ so that they have mean $\hat{x}_k$ and covariance P_{xx}. Each of the sigma points is then propagated in time using the nonlinear process model f_k and subsequently the a priori mean and covariance are calculated. This procedure, which concludes the prediction step, is known as the Unscented Transform (UT). The UT offers second order accuracy in the calculation of the state vector first moment (mean) and first order accuracy in the second order moment (covariance). According to [11], some information about the random variable's higher moments can be captured under specific distribution assumptions and appropriate sigma point weighting, however we will not elaborate further on that.

2.3.2.2 Measurement update

In the measurement update step, which is invoked whenever there are measurements available, the sigma points are used similarly in an Unscented Transform sense but with the nonlinear measurement model h_k this time, in order to predict the mean and the covariance of the upcoming measurements. After the calculation of the system's cross

covariance matrix, the Kalman gain is obtained as a function of the cross covariance and the measurement covariance matrices. The a posteriori state update is given by an equation similar to that of the original Kalman Filter and the Extended Kalman Filter; however the predicted measurement is replaced by the weighted mean of the predicted measurements set.

2.3.2.3 Computational cost considerations

The most computationally intensive elements of the UKF algorithm in comparison with the linear Kalman Filter and the Extended Kalman Filter are the process and measurement models function evaluations required during the Unscented Transforms and the Cholesky decomposition involved in the sigma point generation. However, as the sigma points are sampled in a deterministic way and not in random (as would be the case in particle filtering) their number is eventually low ($n+2$ or $2n+2$, where n is the state vector's length), keeping thus the number of process model and measurement model function evaluations required low enough not to prohibit real time implementation. In navigation applications, like the aircraft state estimation problem addressed here, the UKF's computational cost is considered to be of the same order of magnitude with that of the EKF [18].

2.3.2.4 Implementation equations

Since the UKF algorithm was first presented, various improvements have been proposed by the two original authors. A complete discussion on the UKF and all the available options can be found in [11]. The most suitable version for the aircraft state estimation problem addressed later is chosen and presented here. In particular, two sigma point selection strategies are presented, a scaled symmetric sigma point selection scheme [11] and a simplex sigma point selection scheme [11], [12] that produces almost half the sigma points

than the former but still captures the first two moments of the random variable, offering a great computational advantage for real time implementation. Secondly, an augmented state vector approach is used, incorporating the process and measurement noise into the state vector. That way it is possible to compensate much better care for noise, even in the system's inputs.

The system is described by the process model

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \tag{2.34}$$

and the measurement model

$$z_k = h_k(x_k, v_k) \tag{2.35}$$

It should be noted here that the process and measurement noise vectors need not be additive as in the linear Kalman Filter of Extended Kalman Filter equations; this UKF model formulation allows for multiplicative noise to be treated as well.

For this purpose, we define the augmented state vector

$$x_k^a = \begin{bmatrix} x_k^x \\ x_k^n \end{bmatrix} = \begin{bmatrix} x_k^x \\ w_k \\ v_k \end{bmatrix} \tag{2.36}$$

where x_k^x is the state vector to be estimated and whose dynamics are given by (2.34) and $x_k^n = \begin{bmatrix} w_k^T & v_k^T \end{bmatrix}^T$ is the noise part, containing the zero mean Gaussian process and measurement noise.

As mentioned before, the UKF is based on a deterministic sampling of "sigma points". The sigma points can be thought of as a set of disturbed state vectors that have the given mean and covariance.

Symmetric sigma points

Given the previous state estimate $\hat{x}^a_{k-1}(+)$ and its covariance $P^a_{xx,k-1}(+)$, the symmetric sigma point set consists of $(2n^a + 1)$ weighted "state vectors" scattered around the zeroth sigma point $\hat{x}^a_{k-1}(+)$, where n^a is the length of the augmented state vector,

$$X^a_{k-1}(+) = \left[\begin{array}{ccc} \hat{x}^a_{k-1}(+) & \hat{x}^a_{k-1}(+) + \sqrt{\gamma P^a_{xx,k-1}(+)}|_i & \hat{x}^a_{k-1}(+) - \sqrt{\gamma P^a_{xx,k-1}(+)}|_i \end{array} \right] \quad (2.37)$$

The weight W_0 of the zeroth sigma point can be adjusted to tune the algorithm, while the weight of the other sigma points is

$$W_i = W_{i+n^a} = \frac{1 - W_0}{2n^a} \quad (2.38)$$

The scalar factor γ is given by

$$\gamma = \frac{1}{2W_i} = \frac{1}{2W_{i+n^a}} = \frac{n^a}{1 - W_0} \quad (2.39)$$

In (2.37), $\sqrt{\gamma P^a_{xx,k-1}(+)}|_i$ refers to the i^{th} column of the matrix square-root, as this is given by the lower Cholesky factorization of the form $P = LL^T$. If the upper Cholesky factorization is used, namely $P = U^T U$, the i^{th} row of the matrix square-root is taken instead.

The weights can be any real number, however,

$$\sum_{i=0}^{2n^a} W_i = 0 \tag{2.40}$$

must hold for an unbiased estimate [11].

Spherical simplex sigma points

The spherical simplex sigma points selection algorithm offers an alternative way to generate sigma points that produces only $(n+2)$ sigma points, greatly reducing the computational cost of the upcoming estimation operations. The spherical simplex points lie on a hypersphere centered around the zeroth point. The have been proposed in [12].

Without loss of generality, we consider a system with mean $x = 0$ and covariance $P_{xx} = I$.

The weight of the zeroth point is chosen in the $[0, 1]$ range,

$$0 \leq W_0 \leq 1 \tag{2.41}$$

Using W_0, the weights for the rest of the sigma points are calculated,

$$W_i = \frac{1 - W_0}{n^a + 1} \tag{2.42}$$

For the purpose of deriving the spherical sigma points, a special notation will be used in this section only: sigma points will be denoted as X_i^j, where the superscript j refers to the system's size (it equals n^a), while the subscript i runs from 0 to $(j+1)$.

Three points are defined as

$$X_0^1 = \left[\ 0\ \right], X_1^1 = \left[\ -\frac{1}{\sqrt{2W_1}}\ \right], X_2^1 = \left[\ \frac{1}{\sqrt{2W_1}}\ \right] \tag{2.43}$$

The sigma points are then expanded as follows for $j = 2, ..., n^a$

$$X_i^j = \begin{cases} \begin{bmatrix} X_0^{j-1} \\ 0 \end{bmatrix} & \text{for} \quad i = 0 \\ \begin{bmatrix} X_i^{j-1} \\ -\frac{1}{\sqrt{j(j+1)W_i}} \end{bmatrix} & \text{for} \quad i = 1, ..., j \\ \begin{bmatrix} 0_{j-1} \\ \frac{j}{\sqrt{j(j+1)W_i}} \end{bmatrix} & \text{for} \quad i = j+1 \end{cases} \tag{2.44}$$

The sigma points set generated here for the zero mean case is transformed for the case with arbitrary mean $\bar{z}$ and covariance P_{zz} using

$$Z_i^j = z + \sqrt{P_{zz}} X_i^j \tag{2.45}$$

State prediction

After generating the sigma points with any of the available selection algorithms, we apply the process model to every sigma point, yielding

$$X_k^x(-)|_i = f_{k-1}(X_{k-1}^a(+)|_i, u_{k-1}) \tag{2.46}$$

Using the propagated sigma point set $X_k^x(-)$ the a priori mean is calculated

$$\hat{x}_k^x(-) = \sum_{i=0}^{n^a+1} W_i X_k^x(-)|_i \tag{2.47}$$

while the a priori covariance is given by

$$P_{xx,k}(-) = \sum_{i=0}^{n^a+1} W_i [X_k^x(-)|_i - \hat{x}_k^x(-)][X_k^x(-)|_i - \hat{x}_k^x(-)]^T \tag{2.48}$$

The state prediction procedure, as well as all the next UKF operations are independent of the sigma point generation algorithm used. The only thing that differs according to the generation algorithm selected is the summation index limit in all equations involving summations (like (2.47) and 2.48)). Here, without loss of generality, we assume that the spherical simplex sigma point algorithm has been selected, so $i = 0...(n^a + 1)$.

Measurement update

Since the process model yields only the unaugmented state vector part, the predicted augmented sigma point set is assembled manually as

$$X_k^a(-) = \begin{bmatrix} X_k^x(-) \\ X_{k-1}^n \end{bmatrix} \tag{2.49}$$

The measurement model is used on each propagated sigma point to produce a set of predicted measurements,

$$Z_k|_i = h_k(X_k^a(-)|_i) \tag{2.50}$$

Now the predicted measurement mean and covariance can be similarly calculated,

$$\hat{z}_k = \sum_{i=0}^{2n^a} W_i Z_k|_i \tag{2.51}$$

$$P_{zz,k} = \sum_{i=0}^{2n^a} W_i (Z_k|_i - \hat{z}_k)(Z_k|_i - \hat{z}_k)^T \tag{2.52}$$

The cross covariance is given by

$$P_{xz} = \sum_{i=0}^{2n^a} W_i [X_k^x(-)|_i - \hat{x}_k^x(-)](Z_k|_i - \hat{z}_k)^T \tag{2.53}$$

and it is used to calculate the Kalman gain,

$$K_k = P_{xz} P_{zz}^{-1} \tag{2.54}$$

Finally, the state and the covariance are updated using the familiar equations

$$\hat{x}_k^x(+) = \hat{x}_k^x(-) + K_k(z_k - \hat{z}_k) \tag{2.55}$$

and

$$P_{xx,k}(+) = P_{xx,k}(-) - K_k P_{zz} K_k^T \tag{2.56}$$

Chapter 3

Mathematical Problem Statement

After reviewing the general requirements of state estimation for small unmanned aircraft and the mathematical fundamentals and interrelations of the estimated quantities (Chapter 1), as well as the main state estimation solutions for linear and nonlinear systems (Chapter 2), we are now in a position to define the state estimation problem in a more rigorous way.

The quantities that should be estimated are the aircraft's attitude quaternion q, its curvilinear position $p_{\phi,\lambda,h}$ and its NED velocity v_{NED}. Thus, the state vector of interest is

$$x = \begin{bmatrix} q \\ p_{\phi,\lambda,h} \\ v_{NED} \end{bmatrix} \tag{3.1}$$

Equation (1.20) is the continuous time propagation equation for the quaternion, while (1.29) is the continuous time propagation equation for curvilinear position. The time propagation of the NED velocity is given by

$$\dot{v}_{NED} = a_{NED} = R_b^{NED}(q)a_{ib}^b \tag{3.2}$$

where $R_b^{NED}(q)$ is a function of the attitude q, as given by equation (1.19).

Equations (1.20) (1.29) and (3.2) will be used to derive the discretized estimator's nonlinear process model, which will use as inputs the gyro measurements of body frame angular velocity ω and acceleration a. However, the measurements of the gyro and the accelerometer are both noisy and biased (section 1.5.1) so we must compensate for these two effects before using the measurements as inputs to the process model.

The measurements that can be used in the measurement update step are the magnetic heading as given by the compass or the magnetometer (section 1.5.3) and the position and the velocity given by the GNSS receiver (section 1.5.2). If the compass / magnetometer is embedded into the IMU containing the gyros and the accelerometers, its output will be available concurrently with the estimator's inputs, so at every cycle the compass measurement model will be invoked. GNSS receivers typically produce output at a much lower frequency than IMUs (5-10 Hz for GNSS vs. 100 Hz for IMUs), so the GNSS measurement model will be invoked at more infrequent intervals.

As mentioned previously in Chapter 1, one can integrate the IMU and the GNSS in the "tight" way, using the GNSS receiver's pseudoranges to solve the respective equations (section 1.5.3) in a state estimation framework. The alternative approach, or "loose" integration is to use the position and the velocity computed by the GNSS receiver, without interfering with the internal GNSS operations. The latter approach is followed here.

The small aircraft state estimation problem is highly nonlinear (there are nonlinearities both in the process and the measurement models) and a solution using the Unscented Kalman Filter will be described next. However, as the state vector contains both constrained quantities (q) and vector quantities ($p_{\phi,\lambda,h}$, v_{NED}) the Unscented Kalman Filter has to be modified for this particular problem. The quaternion part of the state vector will be updated in a multiplicative manner, while its covariance will be expressed in terms of a rotation vector, using one less dimension; the non quaternion part of the state

vector will be treated normally, as in the equations of Chapter 2. It should be emphasized here that although the two parts of the state vector are treated differently, this happens concurrently and the attitude estimation is not decoupled from the estimation of position and velocity. To allow for a better and more intuitive presentation of the separation of constrained and vector quantities, the attitude estimation case will be considered first in Chapter 4. The full state estimation case with the state vector (3.1) will be presented next in Chapter 5, building on the method presented in Chapter 4.

More details on the implementation of the Chapter 5 algorithm follows in Chapter 6, along with simulation results and a discussion.

Chapter 4

Attitude Estimation

This chapter deals with the estimation of the aircraft's attitude in an Unscented Kalman Filtering framework. First of all, the shortcomings of the standard UKF formulation that result from the attitude quaternion representation and from the presence of circular measurements are discussed and the respective solutions are presented. In the next sections, the process and measurement models used are described. Figures presenting the attitude estimation results are provided in Chapter 6.

4.1 Estimation strategy

The quantity to be estimated is the attitude quaternion $q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$, thus it will be included in the state vector x. However, since in all small unmanned aircraft applications the measurements are obtained from MEMS gyros, the gyro biases (section 1.5.1) have to be estimated as well. The state vector will therefore be

$$x = \begin{bmatrix} q^T & b^T \end{bmatrix}^T = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 & b_{g_1} & b_{g_2} & b_{g_3} \end{bmatrix}^T \tag{4.1}$$

Since we will use a kinematic process model, the inputs of the estimator are the gyro measurements, namely

$$u = \begin{bmatrix} \omega_{g1} \\ \omega_{g2} \\ \omega_{g3} \end{bmatrix} \tag{4.2}$$

which are considered a Gaussian random variable with covariance corresponding to the gyro noise. The subscripts $g1$, $g2$, $g3$ correspond to the three sensitive axes of the gyro around which the rotational rates are measured. As pointed out in section 1.5.1, in general these axes do not precisely correspond to the axes of the body frame; however since the deviation is very small we ignore it for our application. Thus, we assume that

$$u = \begin{bmatrix} \omega_{g1} \\ \omega_{g2} \\ \omega_{g3} \end{bmatrix} \cong \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4.3}$$

In section 4.4 the state vector x will be augmented to account for the gyro input noise as well, in accordance with the UKF formulation presented in section 2.3.2.

As far as measurements are concerned, there are two approaches followed in literature and in practice. The first one uses a 3D magnetometer or, equivalently, a compass in order to obtain the magnetic heading. The true heading (yaw) can be calculated subsequently from the magnetic heading after accounting for the magnetic declination through the World Magnetic Model [24], as in section 1.5.3. The second approach deals with vector matching [19], [20]. Vector matching uses measurements and reference values of the magnetic field and the gravitational field of the Earth to provide measurement equations for use in the UKF framework. The gravitational field measurements are obtained through the accelerometers' specific force output, after filtering out the acceleration by velocity differentiation. Here a significant issue arises, as in highly dynamic conditions, it is impossible to sufficiently isolate the gravity terms in the accelerometer measurement; resulting in poor attitude estimation.

To avoid imposing any limits in the vehicle's maneuverability, it has been decided to follow the compass approach, relying only on the magnetic heading measurement. The results presented in Chapter 6 show that this successfully bounds the attitude estimation error.

4.2 Handling the attitude quaternion

Reviewing the UKF algorithm, one can easily see that many of the operations involved, like the calculation of the a priori mean and covariance, the state update, etc. involve summations and subtractions that are only valid for a vector space. Although this may be the case for the vast majority of state estimation applications, like in the estimation of range or velocity of targets, it doesn't hold for the attitude estimation case; the attitude quaternion is not a vector quantity since it is constrained (unity norm constraint) in the Special Orthogonal group (3) – SO(3) as it is shown in section 1.3.2. As a result, the various operations of the UKF cannot be applied when the attitude quaternion is the estimator's state vector or part of it.

By referring to the definition of the attitude quaternion (section 1.3.2), we note that although it consists of four parameters, it obeys a normalization constraint, limiting thus its degrees of freedom. The attitude quaternion has three degrees of freedom, as many as the three components of the rotation vector ϕ in equation (1.13).

Following a similar approach to [15], [16] and [17], and taking into account the limited degrees of freedom of the quaternion that actually correspond to a rotation vector, the following modifications are made to the UKF algorithm in order to handle the quaternion:

- The covariance of the quaternion is expressed in terms of a rotation vector $\tilde{\phi}$, using three parameters. Thus, the covariance matrix for a quaternion random variable has dimension (3×3).

- The quaternion update is facilitated in a multiplicative manner, through quaternion premultiplication with a correction quaternion term that corresponds to a correction rotation vector $\phi_k(+)$ calculated using the standard Kalman gain equations.
- If the state vector consists of both a quaternion and other terms that are vector quantities, each part is handled separately; the quaternion part is updated using the special method outlined above, while the vector part is treated in the usual UKF way.

The quaternion treatment is presented in detail in section 4.4.

4.3 Handling magnetic heading measurements

The heading angle is a circular quantity with obvious periodicity (for example $10°$ equals $370°$, etc.). Therefore, first of all there is a need to define the range to work with heading. Here we choose $[-180°, 180°)$.

Regardless of the chosen heading range, the averaging operation of the UKF (section 2.3.2.4) can easily fail, as for example the mean of $179°$ and $-179°$ is $-180°$, while the usual averaging equation would produce a totally incorrect result, as $179° + (-179)° = 0°$. The same condition can be observed around $0°$ if the $[0°, 360°)$ heading range is chosen. Similarly, the covariance calculation also fails around certain points.

For the above reasons, the mean and the covariance of the heading measurement cannot be calculated using the standard UKF formulas; methods from circular statistics [8] are required.

The mean of n samples of a circular quantity a_i is given by

$$\bar{a} = \text{atan2}\left(\frac{1}{n}\sum_{i=1}^{n}\text{sin}a_i, \frac{1}{n}\sum_{i=1}^{n}\text{cos}a_i\right) \tag{4.4}$$

For the weighted mean case, with weights W_i such that $\sum_{i=1}^{n} W_i = 1$, the above equation is transformed into

$$\bar{a} = \text{atan2}\left(\frac{1}{n}\sum_{i=1}^{n}W_i\text{sin}a_i, \frac{1}{n}\sum_{i=1}^{n}W_i\text{cos}a_i\right) \tag{4.5}$$

Equations (4.4) and (4.5) resolve almost all periodicity issues. There is only one case in which they provide an ambiguous result, when there are two measurements that are 180° apart. However, for the purposes of UKF state estimation where the averaging takes place in small area around the mean this is impossible to happen, because it would require a very big value for the covariance matrix entries to scatter predicted heading measurements that far apart and that covariance matrix would have already rendered the system inoperable.

Circular statistics provide only dimensionless measures of dispersion, so a custom solution for the covariance calculation is needed, obtained through subtraction and expressed in degrees or radians. Equation (2.52) of the UKF requires the calculation of the distance of each sigma point Y from the mean $\hat{y}$. As far as the heading is concerned, this distance is given by

$$d = Y - \hat{y} \tag{4.6}$$

Depending on the resulting value of d, a correction is made to resolve the periodicity issue and provide a result in the $[-180°, 180°)$ range,

- if $d < -180°$, then $d \leftarrow d + 360°$
- if $d \geq 180°$, then $d \leftarrow d - 360°$

4.4 Process model

The augmented state vector is

$$x^a = \begin{bmatrix} x^x_{(7\times1)} \\ x^n_{(7\times1)} \end{bmatrix} = \begin{bmatrix} q_{(4\times1)} \\ b_{g(3\times1)} \\ n_{g(3\times1)} \\ n_m \\ n_{gbp(3\times1)} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ b_{g1} \\ b_{g2} \\ b_{g3} \\ n_{g1} \\ n_{g2} \\ n_{g3} \\ n_m \\ n_{gbp_1} \\ n_{gbp_2} \\ n_{gbp_3} \end{bmatrix} \tag{4.7}$$

where $x^n = \begin{bmatrix} n_g & n_m & n_{gbp} \end{bmatrix}^T$ is the zero mean Gaussian noise part containing the gyro and the magnetometer noise components, as well as the gyro bias parameter n_{gbp}, and x^x corresponds to the state vector that it is of interest to be estimated, as given in equation (4.1).

The process model has the form $x^x_k = f(x^a_{k-1})$, as the noise terms are not propagated.

The quaternion propagation equation in continuous time is given in section 1.3.2. Its discrete time equivalent is

$$q_k = e^{\frac{1}{2}\Omega_{k-1}T} q_{k-1} \tag{4.8}$$

where T is the discretization interval with

$$\Omega_{k-1} = \begin{bmatrix} 0 & -\omega_{k-1,x} & -\omega_{k-1,y} & -\omega_{k-1,z} \\ \omega_{k-1,x} & 0 & \omega_{k-1,z} & -\omega_{k-1,y} \\ \omega_{k-1,y} & -\omega_{k-1,z} & 0 & \omega_{k-1,x} \\ \omega_{k-1,z} & \omega_{k-1,y} & -\omega_{k-1,x} & 0 \end{bmatrix} \tag{4.9}$$

where

$$\omega_k = \omega_{g,k} - n_{g,k} - b_{g,k} \tag{4.10}$$

in order to compensate for the gyro measurement noise.

Equation (4.8) utilizes the exponential matrix; a more numerically efficient implementation that makes use of the special form of Ω_k is given in [2].

The gyro biases are propagated according to

$$b_{g,k} = b_{g,k-1} + n_{gbp,k-1}T \tag{4.11}$$

This concludes the process model for the attitude estimation case, which consists of equations (4.8) and (4.11).

4.5 Measurement model

The measurement model has the form $z = h(x^a)$ and relates the system's state to the magnetic heading, as the magnetic compass is the only sensor used in the measurement update of the attitude estimation case.

So,

$$z_k = \psi_{m,k} \tag{4.12}$$

with

$$\psi_{m,k} = \arctan 2\left(2(q_{0,k}q_{3,k} + q_{1,k}q_{2,k}), 1 - 2(q_{2,k}^2 + q_{3,k}^2)\right) - \delta(p_k) + n_{m,k} \tag{4.13}$$

where $\delta(p_k)$ is the magnetic declination, obtained from the World Magnetic Model [24]. The magnetic declination is a function of the vehicle's position p_k, however in the attitude estimation case examined in this chapter the aircraft's position is not included in the state vector and thus it is not estimated. As the magnetic declination is a slowly changing quantity (taking into account the usual speed range for the aircraft class considered here) we can assume that a low frequency, noisy position measurement from a GPS receiver (with few meters accuracy) is more than sufficient for the calculation of $\delta(p_k)$. In the more complete, full state estimator presented in the next chapter this issue is solved by including the position in the estimator's state vector. If there is no GPS receiver available at all and an attitude estimate is still needed, $\delta(p_k)$ can be obtained offline from the World Magnetic Model beforehand, assuming it to be constant for the duration of the flight.

4.6 Attitude estimator equations

Here the complete Unscented Kalman Filter attitude estimation algorithm is presented. The spherical simplex sigma point selection algorithm is used to reduce the computational cost. For better clarity of the special treatment of the quaternion part of the state vector, we will use the notation

$$x_k^x = \begin{bmatrix} x_{k_{(4\times1)}}^{xq} \\ x_{k_{(3\times1)}}^{xv} \end{bmatrix} \tag{4.14}$$

separating the quaternion x_k^{xq} from the vector part x_k^{xv} of the state vector.

4.6.1 Spherical sigma points generation

Given the (7×1) vector $\hat{x}_{k-1}^x(+)$ containing the previous estimate and its (6×6) covariance matrix $P_{k-1}^x(+)$ (the quaternion covariance is expressed in terms of a rotation vector, hence the reduced dimension covariance matrix) the spherical simplex sigma points are generated as follows:

The augmented covariance matrix is assembled, using the known measurement and process noise covariance terms,

$$P_{k-1}^a(+) = \begin{bmatrix} P_{k-1}^x(+) & 0 \\ 0 & R \end{bmatrix} \tag{4.15}$$

where the (7×7) R matrix corresponds to the process, measurement and input noise covariance.

We calculate the lower Cholesky factor C of the augmented covariance, through the lower Cholesky decomposition,

$$P_{k-1}^a(+) = CC^T \tag{4.16}$$

The zero mean / unit covariance spherical sigma points Y are generated as described in section 2.3.2.4 for $n = n^a - 1 = 14 - 1 = 13$ (equations (2.41)-(2.44)).

Then, to get the sigma points set that have mean $\hat{x}^a_{k-1}(+) = \begin{bmatrix} \hat{x}^x_{k-1}(+)^T & 0^T_{(1\times 7)} \end{bmatrix}^T$ and covariance $P^a_{k-1}(+)$, for each i^{th} column $(i = 1....(n+2) = 1...15)$ of the zero mean / unit covariance spherical sigma points set Y (calculated by the algorithm given in section 2.3.2.4) we do the following:

We calculate the term

$$T_i = CY|_i = \begin{bmatrix} \delta_{\tilde{\phi}_{k-1,i}(3\times 1)} \\ \delta_{xv_{k-1,i}(3\times 1)} \\ \delta_{n_{k-1,i}(7\times 1)} \end{bmatrix} \tag{4.17}$$

where

- $\delta_{\tilde{\phi}_{k-1,i}}$ is a rotation vector that corresponds to a quaternion q_{Δ_i} calculated using equation (1.13)
- $\delta_{xv_{k-1,i}}$ is a sigma point "scattering" term that corresponds to the vector part x^{xv} of the system's state vector
- $\delta_{n_{k-1,i}}$ is related to the noise part of the augmented state vector

The i^{th} sigma point's part corresponding to the quaternion is now updated as

$$X^{xq}|_i = q_{\Delta_i} \otimes \hat{x}^{xq}_{k-1}(+) \tag{4.18}$$

The above operation rotates the mean quaternion around each of the deterministically sampled rotation vectors $\delta_{\tilde{\phi}_{k-1,i}}$.

The vector part is treated using the typical UKF equations, as

$$X^{xv}|_i = \hat{x}^{xv}_{k-1} + \delta_{xv_{k-1,i}} \tag{4.19}$$

and

$$X^{n}|_i = \delta_{n_{k-1,i}} \tag{4.20}$$

The weights W_i of the sigma points X_i are given by equation 2.42.

4.6.2 State prediction

Using now the process model of section 4.5 on each sigma point i yields the predicted sigma points set,

$$X^x_k(-)|_i = f_{k-1}\left(X^a_{k-1}(+)|_i, u_{k-1}\right) \tag{4.21}$$

where u is the input vector containing the measured angular rates.

The mean of the predicted sigma points is calculated separately for the quaternion and the vector parts. The mean quaternion according to [17] is given by

$$\hat{x}_k^{xq}(-) = \frac{\sum_{i=1}^{n+2} W_i \left[X_k^{xq}(-)|_i\right]}{||\sum_{i=1}^{n+2} W_i \left[X_k^{xq}(-)|_i\right]||} \tag{22}$$

while the mean for the vector part of the state vector is

$$\hat{x}_k^{xv}(-) = \sum_{i=1}^{n+2} W_i \left[X_k^{xv}(-)|_i\right] \tag{4.23}$$

For the calculation of the covariance, we have to reduce the sigma point quaternions to rotation vectors. We find the quaternion $\delta q_{k,i}$ which corresponds to the rotation between the i^{th} sigma point quaternion and the mean predicted quaternion $\hat{q}_k(-)$,

$$\delta q_{k,i} = X_k^{xq}(-)|_i \otimes \hat{x}_k^{xq}(-)^{-1} \tag{4.24}$$

where -1 refers to the quaternion inversion operation as given by equation (1.15).

Using equation (1.13) we find the rotation vector $\delta_{\tilde{\phi}_{k,i}}$ that corresponds to the $\delta q_{k,i}$ quaternion.

Now, as far as the non quaternion part of the sigma points set is concerned, we find its difference from the previously calculated predicted mean $\hat{x}_k^{xv}(-)$,

$$\delta_{xv_{k,i}} = X_k^{xv}(-)|_i - \hat{x}_k^{xv}(-) \tag{4.25}$$

Combining the calculated difference terms in one vector,

$$\delta_{x^x_{k,i}} = \begin{bmatrix} \delta_{\tilde{\phi}_{k,i}} \\ \delta_{xv_{k,i}} \end{bmatrix} \tag{4.26}$$

the covariance matrix is given by

$$P^x_k(-) = \sum_{i=1}^{n+2} W_i \delta_{x^x_{k,i}} \delta^T_{x^x_{k,i}} \tag{4.27}$$

4.6.3 Measurement prediction

As the noise part of the sigma points is not propagated, but it is used in the measurement prediction step, we manually assemble the new augmented sigma point set, containing the propagated part $X^x_k(-)$ and the previously generated noise part $X^n_{k-1}(+)$, as in equation (2.49),

$$X^a_k(-) = \begin{bmatrix} X^x_k(-) \\ X^n_{k-1}(+) \end{bmatrix} \tag{4.28}$$

The measurement model of section 4.5 is used on each predicted sigma point $X_k(-)|_i$ to yield the corresponding predicted measurement vector $Z_k(-)|_i$,

$$Z_k(-)|_i = h_k(X^a_k(-)|_i) \tag{4.29}$$

Here there is only one measurement (heading), however in general $Z_k(-)|_i$ is a vector, as will be the case in the next Chapter where multiple sensors may produce output in the same measurement update cycle.

According to section 4.3, the mean and the covariance of the heading are calculated using equations (4.5) and (4.6).

Therefore the mean is

$$\hat{z}_k(-) = \text{atan2}\left(\frac{1}{n+2}\sum_{i=1}^{n+2} W_i \left(\text{sin}Z_k(-)|_i\right), \frac{1}{n+2}\sum_{i=1}^{n+2} W_i \left(\text{cos}Z_k(-)|_i\right)\right) \tag{4.30}$$

and the covariance

$$P_{zz} = \sum_{i=1}^{n+2} W_i \delta z_i \delta z_i^T \tag{4.31}$$

where the difference δz_i is the angular difference of the i^{th} predicted measurement from the mean measurement prediction, as given by equation (4.6) to resolve the periodicity issues.

4.6.4 Cross covariance calculation

The cross covariance is given by

$$P_{xz} = \sum_{i=1}^{n+2} W_i \delta_{x^x_{k,i}} \delta z_i^T \tag{4.32}$$

where $\delta_{x^x_{k,i}}$ is the difference of the i^{th} sigma point from the predicted mean, as given by equation (4.26) and δz_i is the difference of the i^{th} predicted measurement from the predicted measurement mean, as given by (4.6).

4.6.5 Kalman gain and state update

The Kalman gain is

$$K_k = P_{xz} P_{zz}^{-1} \tag{4.33}$$

The state update is facilitated separately for the quaternion and the vector part of the state vector.

The Kalman gain can be divided in two parts, K_k^{xq} that corresponds to the quaternion part and K_k^{xv} that corresponds to the rest of the state vector,

$$K_k = \begin{bmatrix} K^{xq}_{k_{(3\times 1)}} \\ K^{xv}_{k_{(3\times 1)}} \end{bmatrix} \tag{4.34}$$

As far as the quaternion is concerned, a correction rotation vector is calculated using the Kalman gain and the actual measurement vector z_k,

$$\phi_k(+) = K_k^{xq}(z_k - \hat{z}_k(-)) \tag{4.35}$$

The angle subtraction above must be facilitated as in section 4.3 due to the heading angle periodicity issues. This rotation vector corresponds to a quaternion $\delta q_k(+)$ given by equation (1.13). Now the quaternion part of the predicted mean state vector is updated in a multiplicative manner,

$$\hat{x}_k^{xq}(+) = \delta q_k(+) \otimes \hat{x}_k^{xq}(-) \tag{4.36}$$

The rest of the state vector is updated using the standard UKF equation,

$$\hat{x}_k^{xv}(+) = \hat{x}_k^{xv}(-) + K_k^{xv}(z_k - \hat{z}_k(-)) \tag{4.37}$$

where again the angle subtraction must follow section 4.3.

4.6.6 Covariance update

The covariance is updated by

$$P_k^x(+) = P_k^x(-) - K_k P_{zz} K_k^T \tag{4.38}$$

Chapter 5

Full State Estimation

This chapter deals with the estimation of the aircraft's position and velocity, in addition to its attitude, using accelerometers and gyroscopes as inputs, and GNSS and compass output as measurements in an Unscented Kalman Filtering estimation framework. Since accelerometers and gyroscopes can exhibit significant biases, these biases will be part of the estimated state vector in order to be subtracted from the measured values before they are used as an input to the estimator.

The development of this chapter and the respective algorithm follows a similar approach to the previous chapter dealing with attitude estimation and the special treatment of the quaternion part of the state vector remains the same. It was chosen to present the attitude on its own in a separate chapter, to emphasize the quaternion handling in that more simple case.

The (16×1) state vector to be estimated in the full state estimation case is

$$x^x = \begin{bmatrix} q^T & p^T_{\phi,\lambda,h} & v^T_{NED} & b^T_g & b^T_a \end{bmatrix}^T \tag{5.1}$$

To compensate for noise, we define the (19×1) noise state vector,

$$x^n = \begin{bmatrix} n^T_g & n^T_a & n^T_m & n^T_{GPS_p} & n^T_{GPS_v} & n^T_{gbp} & n^T_{abp} \end{bmatrix}^T \tag{5.2}$$

where each term is Gaussian white noise with mean 0 and the respective covariance.

Again, the state vector x^x is split into the quaternion part x^{xq} and the vector part x^{xv} to allow for separate treatment of the attitude quaternion.

5.1 Process model for full state estimation

The discrete process model for the curvilinear position $p_{\phi,\lambda,h}$ is

$$\begin{bmatrix} \phi \\ \lambda \\ h \end{bmatrix}_k = \begin{bmatrix} \phi \\ \lambda \\ h \end{bmatrix}_{k-1} + \begin{bmatrix} \frac{v_{N\,k-1}}{R_N(\phi_{k-1})+h_{k-1}} \\ \frac{v_{E\,k-1}}{(R_E(\phi_{k-1})+h_{k-1})\cos\phi_{k-1}} \\ -v_{D\,k-1} \end{bmatrix} T \tag{5.3}$$

where T is the discretization interval. Equation (5.3) is a first order discretization of (1.29).

The NED velocity is propagated according to

$$\begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix}_k = \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix}_{k-1} + \left[R_b^{NED}(q_{k-1})(f^b_{ib,k-1} - n_{a,k-1} - b_{a,k-1} + g(p_{\phi,\lambda,h_{k-1}}) \right] T \tag{4.4}$$

where f^b_{ib} is the accelerometer measurement and g is the gravity vector calculated using the WGS84 model [23].

The accelerometer biases are propagated as the gyro biases in section 4.4. The attitude quaternion is also propagated as in section 4.4.

5.2 Measurement model for UKF full state estimation

In the full state estimation case the sensors used are the compass (as in Chapter 4) and the GNSS / GPS receiver that provides measurements of the aircraft's position and velocity.

The compass measurement model has been discussed in section 4.5.

The GNSS measurement model for position and velocity is

$$z_k^{GPS_p} = p_{\phi,\lambda,h_k} + R_b^{NED} r_{GPS} + n_{GPS_p} \tag{5.5}$$

$$z_k^{GPS_v} = v_{NED_k} + R_b^{NED} {\omega_{ib}}_k^b \times r_{GPS} + n_{GPS_v} \tag{5.6}$$

where r_{GPS} is the GPS antenna lever arm in body frame axes (this should be taken into account when the lever arm is large compared to the specific GNSS configuration accuracy).

Depending on the measurements available at every instance, the appropriate combination of models is used to assemble the predicted measurements vector.

5.3 Full state estimation algorithm

The estimation algorithm for the full state case is identical to the one presented in section 4.6 for the attitude case, if the appropriate full state process model and the appropriate measurement models are used, of course.

Care must be taken to correctly distinguish the quaternion and the vector parts of the

state vector and the sigma points set.

5.4 A note on curvilinear position periodicity

Since the latitude and the longitude are circular quantities, they exhibit periodicity issues near the limits of their definition interval.

Longitude periodicity issues occur near the 180° East / West longitude area. In the unlike event that the unmanned aircraft in question operates near that meridian line, the UKF algorithm may fail as the respective covariance values will increase dramatically. If flying in this area is of interest to the algorithm's designer, the method of section 3.5 could be used at every operation involving the longitude to avoid any issues.

A similar problem is encountered near the poles, where averaging and subtracting curvilinear coordinates may again cause a failure of the UKF algorithm.

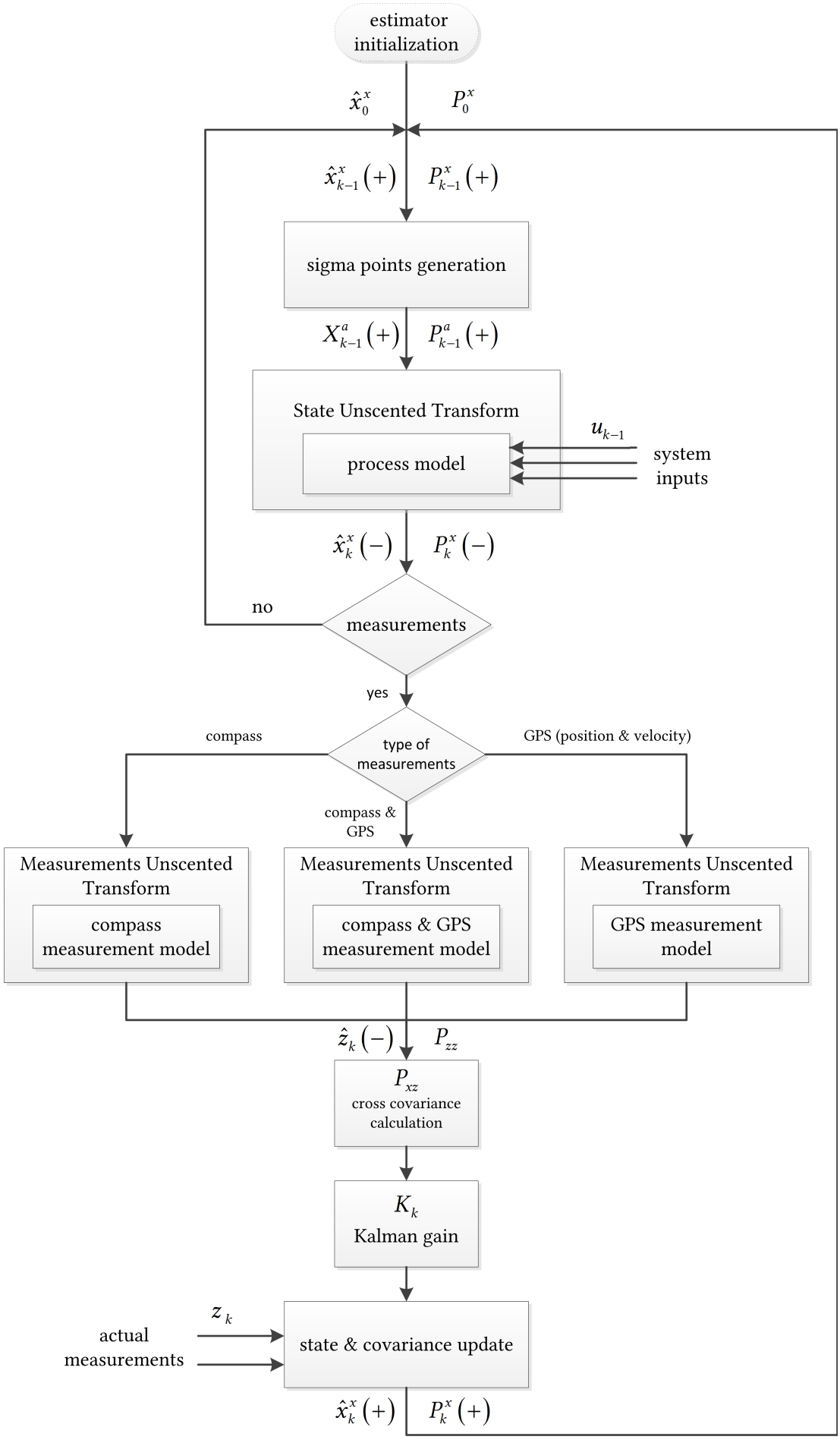
Chapter 6

Implementation – Discussion

This chapter presents a modular implementation of the full state estimation algorithm developed in Chapter 5. In section 6.1 the various components and their interdependence are presented. Next, the initialization of the estimator is dealt with in section 6.2. Simulation results are provided in section 6.3 to verify the proposed estimator's accuracy. A discussion of the results and suggestions for further research follows, concluding the chapter.

6.1 Algorithm

The state estimation solutions presented in this text are of a recursive form, since only the mean and the covariance of the previous estimate are required at the beginning of every estimator iteration. A block diagram that shows the estimator's loop is provided next. All the blocks correspond to the respective equations found in Chapters 4 and 5, except for the algorithm's initialization which is discussed in the next section. The MATLAB function prototypes of the major components shown here are given in Appendix A. The source code files for the simulation example are included in the accompanying CD; alternatively, they can be obtained by contacting the author.


estimator initialization
$\hat{x}_0^x$
P_0^x
$\hat{x}_{k-1}^x(+)$
$P_{k-1}^x(+)$
sigma points generation
$X_{k-1}^a(+)$
$P_{k-1}^a(+)$
State Unscented Transform
process model
u_{k-1}
system inputs
$\hat{x}_k^x(-)$
$P_k^x(-)$
no
measurements
yes
compass
type of measurements
GPS (position & velocity)
compass & GPS
Measurements Unscented Transform
compass measurement model
Measurements Unscented Transform
compass & GPS measurement model
Measurements Unscented Transform
GPS measurement model
$\hat{z}_k(-)$
P_{zz}
P_{xz}
cross covariance calculation
K_k
Kalman gain
z_k
actual measurements
state & covariance update
$\hat{x}_k^x(+)$
$P_k^x(+)$

6.2 Initialization

At the first iteration, the algorithm requires an initial value of the estimated quantities. This chapter presents some practical solutions to the initialization problem, for both attitude and position.

6.2.1 Attitude initialization

For attitude initialization, usually averaging of the compass output for a couple of minutes is enough, since for most stationary aircraft, the roll and pitch angles are 0. Alternatively, if a three axis magnetometer is present, one of the attitude matching algorithms in [19] or [20] may be used, to solve the dual vector field attitude matching problem, given the magnetic field measurement and the accelerometer's specific force output (which for a stationary or a slowly / steadily moving vehicle equals the gravity field components) and the known reference values (using the gravity [23] and magnetic field [24] models with the curvilinear position as input).

6.2.2 Position and velocity initialization

Position can be initialized by averaging the output of a GNSS receiver over a couple of minutes. The initialization period depends on the accuracy of the GNSS positioning output (Differential GPS systems, for example, already have a couple of decimeters accuracy so no lengthy averaging is necessary).

As far as the velocity is concerned, this can be also averaged from the GNSS velocity output, although almost always the aircraft remains stationary during the estimator's

initialization, so the North – East – Down velocity is 0.

6.2.3 In-flight restarts

In the event of a software / hardware malfunction, the estimator may have to be restarted in flight. For that purpose, it is sufficient to provide the last estimated quantities as initial values if the duration of the downtime hasn't been too long. UKF based estimators are robust in terms of initialization uncertainty [18].

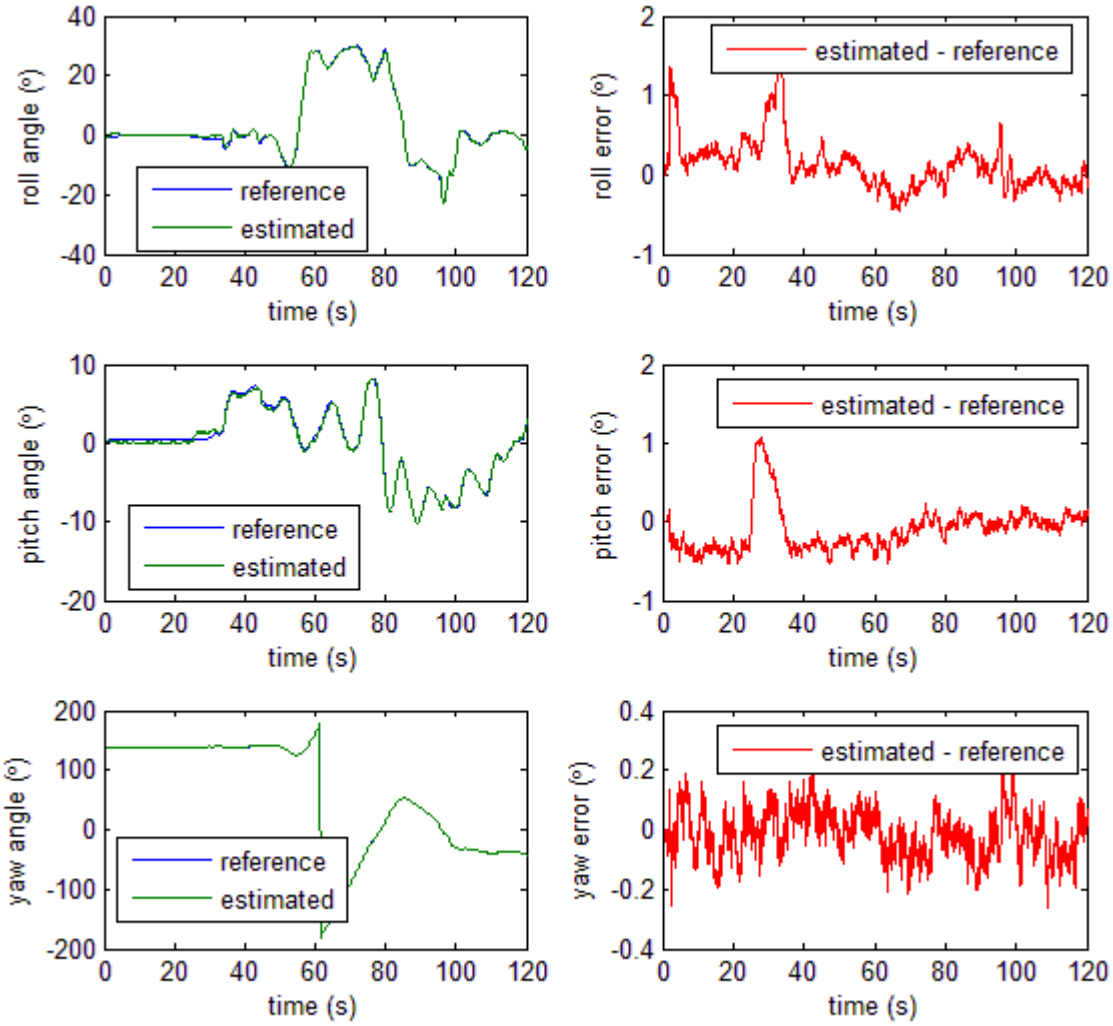
6.3 Results

The following results correspond to the simulated flight of a small aircraft, using the full state estimation algorithm of Chapter 5. The estimator runs at a frequency of 50 Hz, or equivalently whenever there are measurements from the IMU. GPS measurements are available every 5 cycles (at 10 Hz).

The noise characteristics for the IMU inputs correspond to a very small, light and cheap MEMS IMU, with standard deviations $\sigma_{gyro} = 0.0087 rad/s$ and $\sigma_{acc} = 0.02 m/s^2$. The magnetic compass noise is $\sigma_{compass} = 0.7^\circ$ and its measurements are available at 50 Hz.

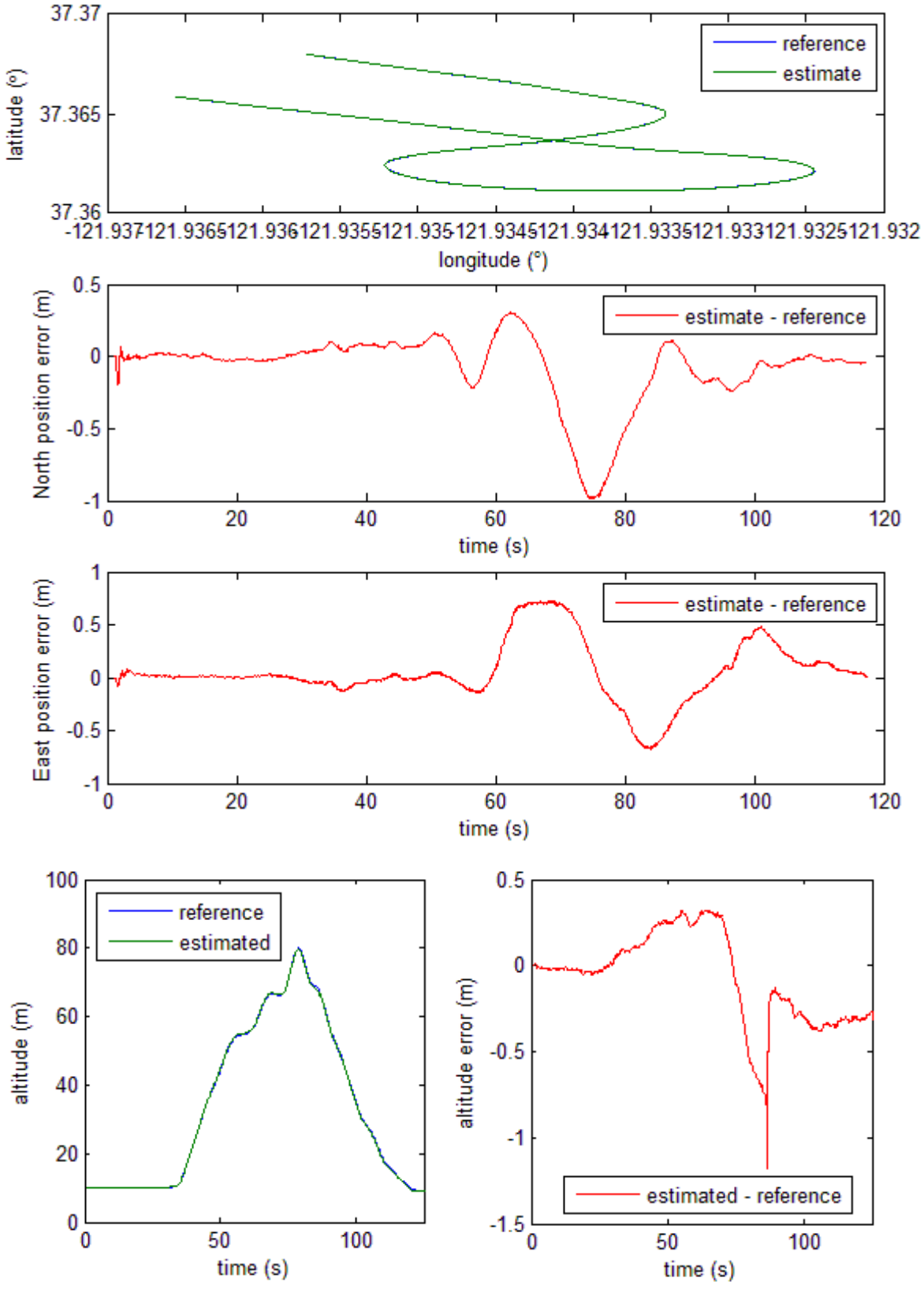
As far as the GPS receiver is concerned, a Differential GPS setup is simulated, with $\sigma_{pos_{\phi,\lambda}} = 0.2m$ and $\sigma_{pos_h} \approx 3 \times \sigma_{pos_{\phi,\lambda}}$.

6.3.1 Attitude estimation results

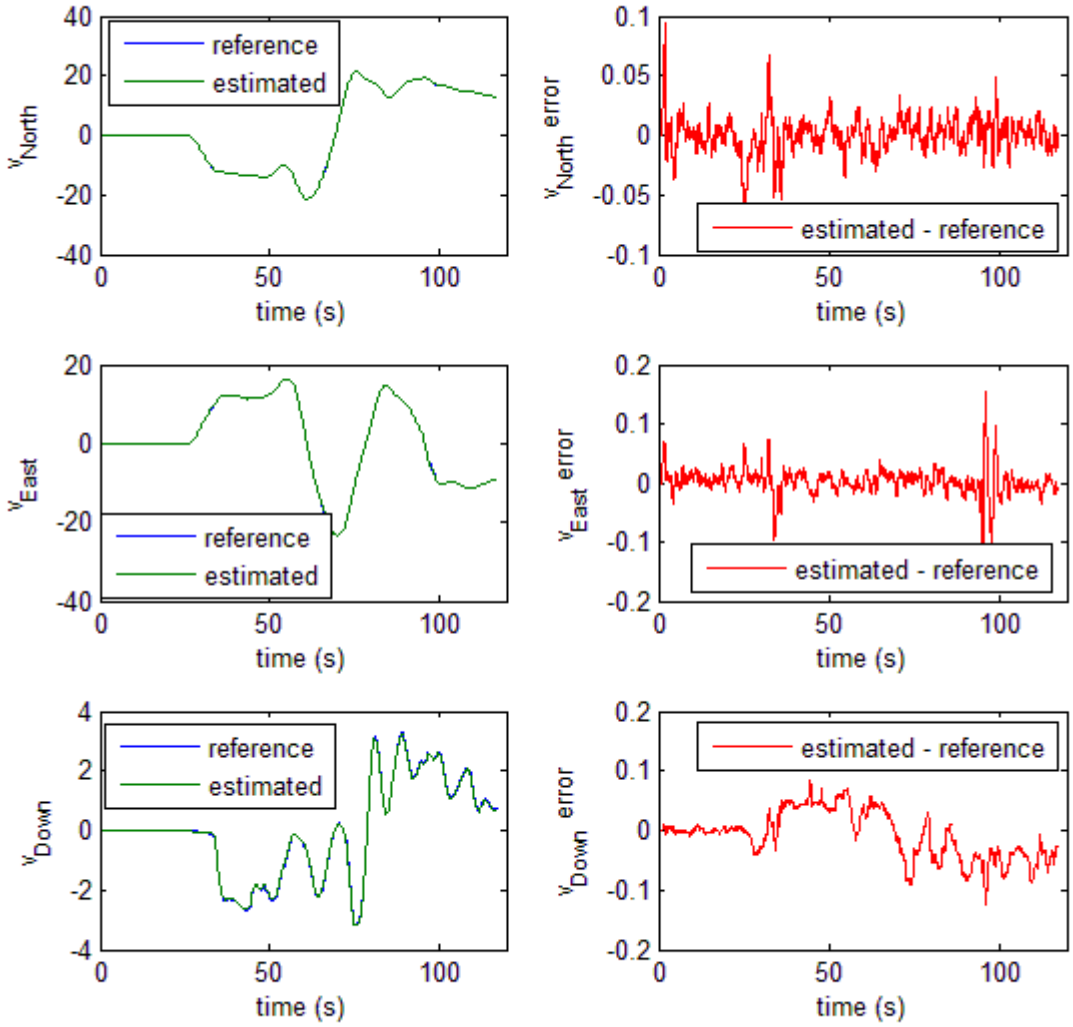


We note that with the exception of the moments when the vehicle first starts to move, the estimate's error seems to behave well, being sufficiently close to zero.

6.3.2 Position estimation results



6.3.3 Velocity estimation results



6.4 Discussion

In this text, a solution to the state estimation problem for small unnamed aircraft is presented and verified through a simulation example. The algorithm is based on a version of the Unscented Kalman Filter, modified appropriately to handle the attitude quaternion and heading measurements. The state vector is split in two parts, one that corresponds to the quaternion and the other that contains vector quantities. The quaternion is transformed into a rotation vector in order to express its covariance, while a rotation vector is again used to update the quaternion part of the state, after the calculation of the Kalman gain. The vector part of the state vector is treated in the usual UKF way. The algorithm is simple and robust, retaining all the benefits of the quaternion attitude representation.

Suggestions for further research include:

- comparison of this estimator's performance with an Extended Kalman Filter solution, in terms of accuracy and computational requirements
- comparative development of a Central Differences based Kalman Filter ([21], [22]) which bears many similarities to the UKF but offers greater accuracy in the estimation of covariance
- derivation of safety limits for GPS denied operation
- verification of the estimator's performance through actual flight experiments

Appendix A

Function Prototypes

The prototypes of the proposed algorithm's implementation functions are given here. The complete source code is included in the accompanying CD, or alternatively can be obtained by contacting the author.

These functions are called from the main loop (**UKF_FG_estimator**.m) to perform various operations described previously in this text.

[y] = **process_model_full**(x)

Full (attitude, position and velocity) kinematic and bias (gyros and accelerometers) process model for the augmented state vector. The inputs are the gyro and accelerometer output, which are a global variable in MATLAB for the purpose of this simulation, hence they are not provided as input to the function.

[Z,W,Wc] = **QuatVSphericalSimplexSigmaPoints**(x,P,W0)

Generates the Spherical Simplex Sigma Point set for noise augmented state vector x with covariance P. The first four elements of the state vector correspond to an attitude quaternion. W0 is a free tuning parameter (see section 2.3.2.4) without noticeable effects in the results of the estimation algorithm, though.

[Y_mean,Y,P,dx] = **UnscentedTransformQuatV**(X,f,Wm,Wc)

Propagates the sigma point set X with weights Wm (for mean) and Wc (for covariance), using the process model function handle f. Returns the propagated sigma point set Y and its mean Y_mean, their covariance and the difference dx of each sigma point from the mean (to calculate the cross covariance matrix later)

[yaw_pred_mean,yaw_pred,P,diff] = **UnscentedTransformMYaw**(X,f,Wm,Wc)

Predicts the magnetic heading measurements and the difference from their mean to assist in the cross covariance calculation later. The measurement model handle f is used.

[meas_pred_mean,meas_pred,P,diff] = **UnscentedTransformMFull**(X,f,Wm,Wc)

Predicts the full measurement set (magnetic heading, GPS position and velocity), according to the measurement model handle f.

[pred_meas] = **measurement_model_full**(x)

Measurement model for the compass + GPS measurement case, taking an augmented vector as input.

[pred_meas] = **measurement_model_MH**(x)

Measurement model for the compass case, taking an augmented vector as input.

[a_avrg] = **angle_avrg**(a,W)

Calculates the weighted (W) average of a vector of angles (a)

[a] = **angle_diff**(a1,a2)

Calculates the difference in rad between 2 angles taking into consideration the angle periodicity on a circle

[R] = **Rbn**(q)

Quaternion rotation matrix (body to NED)

[f] = q**uat_findrotation_custom**(q)

Finds the rotation vector corresponding to the given quaternion

[dq] = **quat_rotation_custom**(f)

Finds the quaternion corresponding to the given rotation vector

[q] = **quatinv_custom**(q)

Finds the inverse quaternion for the given quaternion q

[p] = **quatmult_custom**(q,r)

Performs the quaternion multiplication operation for two given quaternions q,r

[RE] = **WGS_RE**(L)

Calculates the transverse radius of curvature of the Earth given the geodetic latitude

[RN] = **WGS_RN**(L)

Calculates the meridian radius of curvature of the Earth given the geodetic latitude

MATLAB's built in functions **wrldmagm** and **gravitywgs84** are used as well to calculate

the magnetic declination and the gravity as a function of position (and time for the magnetic model case).

The quaternion related functions include the word “custom” in their name in order to distinguish them from the MATLAB built in equivalent functions.

Appendix B

References

Books

[1] M. Musial, "System Architecture of Small Autonomous UAVs," VDM Verlag Dr. Müller, 2008

[2] J. A. Farell, "Aided Navigation," McGraw Hill, 2008

[3] P. D. Groves, "Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems," Artech House, 2008

[4] M. S. Grewal, L. R. Weill, A. P. Andrews, "Global Positioning Systems, Inertial Navigation, and Integration," Wiley, 2007

[5] A. Gelb (ed), "Applied Optimal Estimation," MIT Press, 1974

[6] P. S. Maybeck, "Stochastic Models, Estimation and Control," vol. I, II & III, Academic Press, 1979

[7] M. S. Grewal, A. P. Andrews, "Kalman Filtering: Theory and Practice Using MATLAB®," Wiley, 2008

[8] K. V. Mardia, P. E. Jupp, "Directional Statistics," Wiley, 2009

Journal and Conference Publications

[9] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” ASME Journal of Basic Engineering, vol. 82D, 1960

[10] G. T. Schmidt, “Practical Aspects of Kalman Filtering Implementation,” NATO Advisory Group for Aerospace Research and Development (AGARD) Lecture Series No. 82, 1976

[11] S. J. Julier, J. K. Uhlmann, "Unscented filtering and nonlinear estimation," Proceedings of the IEEE, vol.92, no.3, pp. 401- 422, Mar 2004

[12] S. J. Julier, "The spherical simplex unscented transformation," American Control Conference, 2003. Proceedings of the 2003, vol.3, pp. 2430- 2434, 4-6 June 2003

[13] R. Van der Merwe, E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on, vol.6, pp.3461-3464, 2001

[14] M. D. Shuster, “A Survey of Attitude Representations,” Journal of the Astronautical Sciences, Vol. 42, No. 4, October-December 1993

[15] E. Kraft, "A quaternion-based unscented Kalman filter for orientation tracking," Information Fusion, 2003. Proceedings of the Sixth International Conference of, vol.1, pp. 47- 54, 2003

[16] B. J. Sipos, "Application of the manifold-constrained unscented Kalman filter," Position, Location and Navigation Symposium, 2008 IEEE/ION, pp.30-43, 5-8 May 2008

[17] W. Khoder, B. Fassinut-Mombot, M. Benjelloun, "Quaternion Unscented Kalman Filtering for integrated Inertial Navigation and GPS," Information Fusion, 2008 11th International Conference on, pp.1-8, June 30 2008-July 3 2008

[18] R. Van der Merwe, E. A. Wan, "Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion – Applications to Integrated Navigation," Guidance, Navigation and Control, 2004 AIAA Conference on, August 2008

[19] F. Landis Markley, D. Mortari, "How to Estimate Attitude from Vector Observations," AAS/AIAA Astrodynamics Specialist Conference, August 1999

[20] D. Gebre-Egziabher, G. H. Elkhaim, "MAV Attitude Determination by Vector Matching," Aerospace and Electronic Systems, IEEE Transactions on, vol.44, no.3, pp.1012-1028, July 2008

[21] M. Nørgaard, N. K. Poulsen, O. Ravn, "New Developments in State Estimation for Nonlinear Systems," Automatica, vol.36, pp. 1627-1638, 2000

[22] K. Ito, K. Xiong, "Gaussian Filters for Nonlinear Filtering Problems," Automatic Control, IEEE Transactions on, vol.45, no.5, pp.910-927, May 2000

Reports

[23] US National Geospatial-Intelligence Agency, World Geodetic System WGS84

[24] US National Geospatial-Intelligence Agency, World Magnetic Model WMM2010