Software Requirements Specification for Attitude Check: IMU-based Attitude Estimation

Adrian Sochaniwsky

April 3, 2024

Contents

Revision History

Date	Version	Notes
2024/02/02	1.0	Initial Release.
2024/04/03	2.0	Addressed reviewer comments.

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	unit	SI
m	length	metre
rad	angle	radian
s	time	second
T	magnetic field	tesla

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the attitude and heading reference systems literature and existing documentation. The symbols are listed in alphabetical order.

symbol	unit	description
a	m/s^2	linear acceleration
m	µT	magnetic field strength
θ	rad	angular position
ω	rad/s	angular velocity
Δt	ms	sampling time
$\mathbf{g}_0$	m/s^2	gravitational acceleration vector
${}^{E}\mathbf{b}$	µT	local magnetic field vector
γ	-	Complimentary filter parameter
${}^{S}_{E}\mathbf{q}$	rad	Attitude (orientation) of the sensor-frame relative to the earth-frame
η	-	Fraction of development time

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
API	Application Programming Interface
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
TM	Theoretical Model
IMU	Inertial Measurement Unit
MEMS	Micro-electromechanical System
NED	North-East-Down
WMM	World Magnetic Model.

1.4 Mathematical Notation

Vector: Vectors will be distinguished via **bold** face lowercase letters such as, $\mathbf{b}$.

Matrix: Matrices will be distinguished via **bold** face upper case letters such as, $\mathbf{R}$.

Time: Variables for a given time step, t will be denoted with a sub-postscript such as, $\mathbf{s}_t$.

Frame: The reference frame for a variable will be denoted by a super-prescript, such as ${}^{E}\mathbf{s}$. Where the vector $\mathbf{s}$ taken with respect to the Earth frame.

Transform: The rotational transform between two frames will be denoted using pre-scripts. For example, ${}^{B}_{A}\mathbf{T}$ describes the orientation of frame A relative to B.

2 Introduction

The goal of attitude estimation is to determine the rotation of an object relative to a reference frame using sensor measurements. Attitude estimation is essential for many applications, such as navigation and control of spacecraft, drones, or robots. Once only possible with expensive hardware, advances in Micro-Electro-Mechanical systems (MEMS) allow an inexpensive Inertial Measurement Unit (IMU) to provide the necessary measurements for attitude estimation. Typical IMU sensors contain a 3-axis accelerometer, 3-axis magnetometer, and 3-axis gyroscope. However, challenges arise from the presence of noise, bias, drift, and uncertainty in the sensor data, as well as the nonlinearity of the object dynamics [2].

2.1 Purpose of Document

This document defines the scope, requirements, and functionality, of a software product. It serves as a contract between the developers and the stakeholders, ensuring that both parties have a clear and common understanding of what the software should do and how it should perform. This SRS also helps the developers to plan, design, test, and maintain the software according to the agreed-upon requirements. This document can reduce the risk of errors, delays, and conflicts during the software development process. [3]

2.2 Scope of Requirements

The scope of this document will only consider operating conditions where all measured values are within the range of the sensors, and the system is at or near the surface of the Earth. Furthermore, timescale of the system's operation is small compared to time varying changes in the Earth's magnetic field. Since this program is designed for robotics and drone applications, all speeds are much slower than the speed of light, thus relativistic effects are outside the scope of this project.

2.3 Characteristics of Intended Reader

The reader should have the following understanding: Linear Algebra Level II, Numerical Methods, Newtonian Mechanics and Dynamics Level I, State Estimation and Filtering, Robot Kinematics. The reader should have some familiarity with digital sensors.

2.4 Organization of Document

The rest of this document is structured as follows. Section 3 discusses the general context and description of the system. Section 4 details the specific system description, goals, and definitions. Section 5 covers system requirements. Section 6 outlines likely changes for the system. Section 7 discusses unlikely changes. Section 8 covers the traceability of the requirements. Section 9 discusses the system development plan.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

Attitude estimation is a lower-level operation in an overall robotics or aerospace application. Once the hardware is installed and the software is compiled, there is no direct human interaction with the system in the context of measuring or estimating orientation. Figure 1 shows the general system context of Attitude Check, the IMU measurements are filtered and the orientation estimate is calculated for downstream use. Higher level functions that perform guidance, navigation, and control consume orientation to complete higher level objectives.

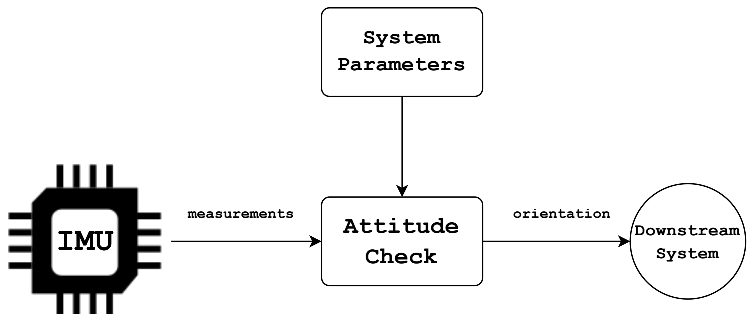

Figure 1: System Context

- User Responsibilities:
  - Provide IMU measurements.
  - Provide system parameters.
- Attitude Check Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number.
  - Return orientation value for each set of measurements.

3.2 User Characteristics

The typical user of Attitude Check is expected to have an understanding of the purpose, inputs, and output of an attitude estimator. This project is designed for users looking to process IMU data. High-school level kinematics is required. For optimal results the user will be required to adjust the gain parameter, this may require experimentation.

3.3 System Constraints

Attitude Check is often ported to microcontrollers or used in high performance applications. Attitude check should be implemented in a language that is compatible with common microcontrollers and existing robotics code bases.

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

Attitude Check is intended to estimate the attitude of an IMU sensor, given noisy measurements.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Attitude: Also referred to as orientation, is the angle of an object relative to another frame of reference
- Coordinate frame: the 3D frame of reference for a particular rigid body, comprised of 3 orthogonal vectors that follow the right-hand rule.
- Euler angles: An intuitive method of representing rotations in 3D space, Euler angles use 3 rotations about the x, y, and z axes. This representation is susceptible to gimbal lock, when 2 axes are parallel, the 3rd axis is immobile.
- Requirement: A statement of a functional or non-functional property that a software system should accomplish.
- Rotation Matrix: a $n \times n$ matrix used to perform rotation transformations in Euclidean space. They are also used to describe the orientation of a rigid body.
- Transform: a computation that converts data from one coordinate frame to another.
- Quaternion: A quaternion is a compact mathematical notation for representing 3D rotations using a scalar component, $q_w \in \mathbb{R}$, and a vector component $(q_x, q_y, q_z) \in \mathbb{R}^3$ to create $q = [q_w \quad q_x \quad q_y \quad q_z]^T$. Unlike Euler angles, quaternions numerically stable, and compact compared to rotation matrices. Quaternion multiplication uses the Hamilton product, refer to [13] for the definition. See [1] for more information.

4.1.2 Physical System Description

The physical system of Attitude Check, as shown in Figure 2 includes the following elements:

PS1: Problem Domain is $\mathbb{R}^3$. The Cartesian coordinate system used for all mathematical operations.

PS2: Coordinate Frames:

- **PS2a** Sensor frame: This is a non-inertial reference frame that aligns with the IMU output [8]. Figure 2 shows the details:
  - * The origin is at the mass center of the IMU.
  - * The x-axis points to the front of the IMU.
  - * The y-axis points to the right of the IMU.
  - * The z-axis points down, perpendicular to the x-y plane.
- **PS2b** Navigation Frame: This is a reference frame used to describe the IMU orientation relative to the earth surface, shown in Figure 2 as the n-frame. The origin is below the mass center of the IMU, on the Earths surface, the axes follow the North-East-Down (NED) convention, see [14] for details.
- **PS2c** Earth Frame: This is a coordinate system attached to the Earth. Also referred to as the e-frame, it is defined by:
  - * The origin is at the mass center of the Earth.
  - * The x-axis points to the Greenwich meridian.
  - * The y-axis is 90 degrees of the Greenwich meridian.
  - * The z-axis is the rotational axis of the Earth.

PS3: Forces:

- **PS3a:** Gravitational Force. All objects in the domain that have mass are subject to gravitational acceleration that is perpendicular to the surface of the earth. This force may vary with position on the Earth's surface.
- **PS3b:** Magnetic Force. The strength of the Earth's magnetic field can be measured to assist attitude estimation [15]. It may vary with position on the Earth's surface or time.

PS4: Initial conditions. The initial condition is the orientation of the system at startup. It is an optional value provided by the user if magnetometer measurements are present.

Note: This document will refer to the Navigational frame as the Earth frame, since this follows the original documentation [9] and is more intuitive for the reader.

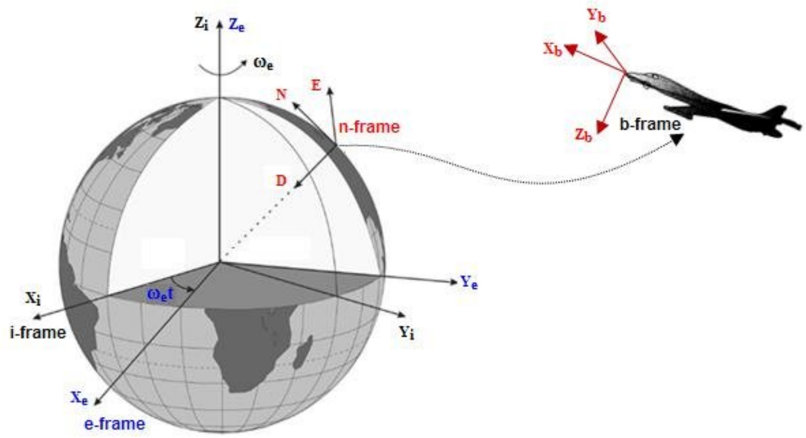

Figure 2: Physical System Context [10].

4.1.3 Goal Statements

The goal statement is:

GS1 : Convert sequential IMU measurements into an orientation relative to the Earth.

4.2 Solution Characteristics Specification

The instance models that govern Attitude Check are presented in Subsection 4.2.7. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Types

- All scalar values in this system are $\in \mathbb{R}$.
- Each sensor produces a measurement vector $\in \mathbb{R}^3$.
- All calculations with quaternions are $\in \mathbb{R}^4$, the quaternion space. And thus must follow quaternion algebra [1].

4.2.2 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [TM], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: Gravitational acceleration, g_0, is assumed to be constant in the z-axis relative to the Navigation frame.

A2: The east component of the magnetic field is assumed to be negligible.

A3: The Earth frame is considered inertial, and the rotation of the earth has no effect on the IMU.

A4: Sensor non-linearities are not considered.

A5: The sensor bandwidth is adequate for measuring the change in state at each time step. I.e. the frequency of the change in state is less than the bandwidth of the system.

A6: Accelerometer g-sensitivity is not considered. Under constant linear acceleration, the bias of an accelerometer may shift over time.

A7: Quantization and analog to conversion error is ignored. The error from measuring of continuous phenomena with digital circuits will be ignored.

A8: The IMU will not experience phenomena outside the sensor's measurement range. I.e. the system will not accelerate beyond the accelerometer's measurement range.

A9: The gyroscope bias is assumed to be constant, and the sensor is calibrated and internally compensates for the bias.

A10: The magnetometer is assumed to experience no local disturbance or slow time varying changes.

A11: The body/IMU is assumed to be a rigid body.

A12: The system is assumed to be in a quasi-static state with little to no angular or linear acceleration.

4.2.3 Theoretical Models

This section focuses on the general equations and laws that Attitude Check is based on.

RefName: TM1

Label: Sensor Measurement Model

Equation: $\mathbf{s} = \mathbf{s}_m + \mathbf{b}_s + \boldsymbol{\mu}_s \in \mathbb{R}^n$

Description: The above equation gives the model for a sensor measurement. Where s_m is the true measurement, b_s is a bias, and μ_s is an additive measurement noise.

For this equation to apply, other forms of error such as non-linearities A4, bandwidth limits A5, quantization error A7, out-of-range measurements A8 are assumed to be negligible.

Notes: The symbol for bias, b, should not be confused with a magnetic field measurement.

Source: http://dx.doi.org/10.1109/TITS.2016.2617200

Ref. By: DD1, DD2, DD3

Preconditions for TM1: None

Derivation for TM1: Not Applicable

RefName: TM2

Label: Angular Velocity

Equation: $\omega = \dot{\theta}$

Description: The above equation gives the definition of angular velocity. Where ω is the derivative of the angular position, θ, with respect to time.

Notes: None.

Source: [11]

Ref. By: GD1, DD2

Preconditions for TM2: None

Derivation for TM2: Not Applicable

RefName: TM3

Label: Rotational Kinematics

Equation: $\theta_t = \theta_{t-1} + \omega_t \Delta t$

Description: The above equation gives the definition of angular kinematics. Where θ_t is the current angular position, θ_{t-1} is the previous position, Δt is the period, and ω_t is the average angular velocity.

This formula assumes that ω is constant or nearly constant during the Δt (A12).

Notes: None.

Source: [11]

Ref. By: IM1

Preconditions for TM3: None

Derivation for TM3: Not Applicable

RefName: TM4

Label: Linear Acceleration

Equation: $a = \ddot{x}$

Description: The above equation gives the definition of linear acceleration. Where a is the second derivative of the linear position, x, with respect to time.

Notes: None.

Source: [11]

Ref. By: DD1

Preconditions for TM4: None

Derivation for TM4: Not Applicable

RefName: TM5

Label: Complimentary Filter

Equation: $s_f = \gamma s_1 + (1 - \gamma) s_2$

Description: The complimentary filter is a simple method to fuse two values with the same units together. Where s_f is the final result of fusing the two values s_1 and s_2. The parameter $\gamma \in [0.0, 1.0]$, where a value of 0 completely trusts s_2 and a value of 1 completely trusts s_1.

Notes: None.

Source: [4]

Ref. By: IM4, IM5.

Preconditions for TM5: None

Derivation for TM5: Not Applicable

RefName: TM6

Label: Orientation from vector observations.

Equation:

$$
\begin{aligned}
{}^S_F\mathbf{q}_d &= \min_{{}^S_F\mathbf{q} \in \mathbb{R}^4} f({}^S_F\mathbf{q},\, {}^F\mathbf{d},\, {}^S\mathbf{s}) \\
f({}^S_F\mathbf{q},\, {}^F\mathbf{d},\, {}^S\mathbf{s}) &= {}^S_F\mathbf{q}\mathbf{p}({}^F\mathbf{d})\mathbf{p}({}^S\mathbf{s}) - {}^S\mathbf{s}
\end{aligned}
$$

Description: Given a measurement of a vector in a non-intertial sensor frame, ${}^S\mathbf{s}$, and a known field direction relative to a fixed-frame, ${}^F\mathbf{d}$, the quaternion for this orientation can be approximated. However, quaternions require a complete solution [9]. Thus, the minimization problem presented solves for ${}^S_F\mathbf{q}_\alpha$, the quaternion that represents the orientation of the sensor frame relative to the fixed-frame. Where the subscript d denotes the quaternion is minimized using ${}^F\mathbf{d}$.

Notes: The function $\mathbf{p}(\mathbf{v})$ creates a pure quaternion, $(0, v_x, v_y, v_z) \in \mathbf{R}^4$ from $\mathbf{v} \in \mathbf{R}^3$. Quaternion multiplication uses the Hamilton product [13].

Source: [9]

Ref. By: IM2 IM3

Preconditions for TM6: None

Derivation for TM6: Not Applicable

4.2.4 General Definitions

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**Angular velocity as a quaternion.**
SI Units	$\mathrm{rad\,s^{-1}}$
Equation	$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q}\mathbf{p}(\boldsymbol{\omega})$
Description	The above equation gives the model for the evolution of the orientation over time. Where $\mathbf{q}$ is a unit quaternion, $\dot{\mathbf{q}}$ is the rate of change of the quaternion, and $\mathbf{p}(\boldsymbol{\omega})$ is the unitary pure quaternion associated to the angular velocity $\mathbf{p}(\boldsymbol{\omega}) = (0, \omega_x, \omega_y, \omega_z)$. For this equation to apply, the object/IMU must be a rigid body (A11). **Note:** Quaternion multiplication uses the Hamilton product [13].
Source	[6]
Ref. By	IM1.

4.2.5 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Acceleration Measurement**
Symbol	a
SI Units	$\mathrm{m\,s^{-2}}$
Equation	$\mathbf{a} = \mathbf{a}_m + \mathbf{b}_a + \boldsymbol{\mu}_a \in \mathbb{R}^3$
Description	$\mathbf{a_m}$ is the true acceleration vector defined by TM4. $\mathbf{b_a}$ is the bias for each axis of the accelerometer. μ_a is the additive noise vector. The accelerometer measurement vector $\mathbf{a}$ is the sum of the components above. Often, acceleration measurements will have the reference frame and time step added. Example: ${}^S\mathbf{a}_t$.
Sources	
Ref. By	IM2, IM3, IM4.

Number	DD2
Label	**Gyroscope Measurement**
Symbol	$\boldsymbol{\omega}$
SI Units	$\mathrm{rad\,s^{-1}}$
Equation	$\boldsymbol{\omega} = \boldsymbol{\omega}_m + \mathbf{b}_g + \boldsymbol{\mu}_g \in \mathbb{R}^3$
Description	$\boldsymbol{\omega}_m$ is the true gyroscopic rate vector defined by TM2. $\mathbf{b}_g$ is the bias for each axis of the gyroscope. μ_g is the additive noise vector. The gyroscope measurement vector $\boldsymbol{\omega}$ is the sum of the components above. Often, gyroscope measurements will have the reference frame and time step added. Example: ${}^S\boldsymbol{\omega}_t$.
Sources	
Ref. By	IM1, IM4, IM5.

Number	DD3
Label	**Magnetometer Measurement**
Symbol	$\mathbf{m}$
SI Units	T
Equation	$\mathbf{m} = \mathbf{m}_m + \mathbf{b}_m + \boldsymbol{\mu}_m \in \mathbb{R}^3$
Description	$\mathbf{m}_m$ is the true magnetometer vector. $\mathbf{b}_m$ is the bias for each axis of the magnetometer. μ_m is the additive noise vector. The magnetometer measurement vector $\mathbf{m}$ is the sum of the components above. Often, magnetometer measurements will have the reference frame and time step added. Example: ${}^{S}\mathbf{m}_t$.
Sources	
Ref. By	IM3, IM5.

4.2.6 Data Types

This section is not applicable.

4.2.7 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.5 to replace the abstract symbols in the models identified in Sections 4.2.3 and 4.2.4.

The goal GS4.1.3 is solved by combining the results from IM5 when all three sensor measurements are provided, and IM4 when the magnetometer is omitted.

Number	IM1
Label	**Orientation from Gyro measurements**
Input	$\mathbf{q}_{t-1}, {}^S\boldsymbol{\omega}_t, \Delta t$ The input is constrained so that $\lvert\omega_t\rvert \leq \omega_{\max}$ (A8)
Output	$\mathbf{q}_{\omega,t} = \mathbf{q}_{t-1} + \dot{\mathbf{q}}_t \Delta t$ $= \mathbf{q}_{t-1} + \frac{1}{2}\mathbf{q}_{t-1}\mathbf{p}({}^S\boldsymbol{\omega}_t)\Delta t$
Description	The current orientation $\mathbf{q}_{\omega,t}$ is computed using TM3. Where $\mathbf{q}_{t-1}$ the previous orientation and $\dot{\mathbf{q}}_t$ from GD1, which requires ${}^S\boldsymbol{\omega}_t$, the current gyroscope measurement in the sensor frame. Where Δt is the period since the last calculation. The additional subscript ω denotes that the quaternion is computed from angular rate,
Sources	[5], [9]
Ref. By	IM4, IM5.

Number	IM2
Label	**Orientation from Accelerometer measurements**
Input	${}^S_E\mathbf{q}, {}^E\mathbf{g}_0, {}^S\mathbf{a}$ The gravity vector is normalized to $\mathbf{g} = (0,0,1)^T$ (A1). The acceleration measurement is in the sensor frame, and constrained to $\lvert a\rvert \leq a_{\max}$ (A8).
Output	${}^S_E\mathbf{q}_g = \min_{{}^S_E\mathbf{q} \in \mathbb{R}^4} f({}^S_E\mathbf{q}, {}^E\mathbf{g}_0, {}^S\mathbf{a})$
Description	Using the known gravitational field in the Earth frame, and a measured sensor-frame acceleration under A12, the orientation, ${}^S_E\mathbf{q}_g$, can be computed using TM6. Since the gravity vector only has one component, this method cannot accurately estimate the heading angle, only pitch and role are accurate. The Earth frame is assumed to be inertial A3.
Sources	[5], [9]
Ref. By	IM4.

Number	IM3
Label	**Orientation from Accelerometer and Magnetometer**
Input	${}^S_E\mathbf{q}$, ${}^E\mathbf{g}_0$, ${}^S\mathbf{a}$, ${}^E\mathbf{b}$, ${}^S\mathbf{m}$ The gravity vector is normalized to ${}^E\mathbf{g}_0 = (0,0,1)^T$ (A1). Following A2 the magnetic field vector is simplified to $\mathbf{b} = (b_x, 0, b_z)^T$. The acceleration and magnetometer measurements are in the sensor frame.
Output	${}^S_E\mathbf{q}_{g,b} = \min\limits_{{}^S_E\mathbf{q}\in\mathbb{R}^4} f_{g,b}({}^S_E\mathbf{q}, {}^E\mathbf{g}_0, {}^S\mathbf{a}, {}^E\mathbf{b}, {}^S\mathbf{m})$ $f_{g,b}({}^S_E\mathbf{q}, {}^E\mathbf{g}_0, {}^S\mathbf{a}, {}^E\mathbf{b}, {}^S\mathbf{m}) = \begin{bmatrix} f_g({}^S_E\mathbf{q}, {}^E\mathbf{g}_0, {}^S\mathbf{a}) \\ f_b({}^S_E\mathbf{q}, {}^E\mathbf{b}, {}^S\mathbf{m}) \end{bmatrix}$
Description	Combining IM2 with the known magnetic field in the Earth frame and a measured sensor-frame magnetic field, the orientation, ${}^S_E\mathbf{q}_{g,b}$, can be computed using TM6.
Sources	[5], [9]
Ref. By	IM5.

Number	IM4
Label	**Orientation from Accelerometer and Gyroscope**
Input	${}^S_E\mathbf{q}_{\omega,t}, {}^S_E\mathbf{q}_{g,t}, {}^S\mathbf{a}_t, {}^E\mathbf{g}_0, {}^S\boldsymbol{\omega}_t, \Delta t$ The sensor measurements are from the current time step, and the quaternion estimate is from the previous time step.
Output	${}^S_E\mathbf{q}_t = \gamma\, {}^S_E\mathbf{q}_{\omega,t} + (1-\gamma)\, {}^S_E\mathbf{q}_{g,t}$
Description	Combining estimates from IM1 and IM2 using the complementary filter from TM5 produces a quaternion output using accelerometer and gyroscope measurements.
Sources	[9]
Ref. By	R4.

Number	IM5
Label	**Orientation from Accelerometer, Gyroscope, and Magnetometer**
Input	${}^S_E\mathbf{q}_{\omega,t}$, ${}^S_E\mathbf{q}_{(g,b),t}$, ${}^S\mathbf{a}_t$, ${}^E\mathbf{g}_0$, ${}^S\boldsymbol{\omega}_t$, Δt The sensor measurements are from the current time step, and the quaternion estimate is from the previous time step.
Output	${}^S_E\mathbf{q}_t = \gamma\, {}^S_E\mathbf{q}_{\omega,t} + (1-\gamma)\, {}^S_E\mathbf{q}_{(g,b),t}$
Description	Combining estimates from IM1 and 3 using the complementary filter from TM5 produces a quaternion output using accelerometer and gyroscope measurements.
Sources	[9]
Ref. By	R3.

Number	IM6
Label	**Initial orientation from Accelerometer**
Input	${}^S\mathbf{a}_t$ The sensor measurements are from the current time step, and the quaternion estimate is from the previous time step.
Output	$\theta = \arctan(\frac{a_y}{a_z})$ $\phi = \arctan\left(\frac{-\,a_x}{\sqrt{a_y^2 + a_z^2}}\right)$ $\psi = 0$
Description	Using accelerometer data, roll and pitch Euler angles can be computed. Yaw cannot be computed and is set to 0.
Sources	https://ahrs.readthedocs.io/en/latest/filters/tilt.html
Ref. By	R2.

Number	IM7
Label	**Initial orientation from Accelerometer, and Magnetometer**
Input	${}^{S}\mathbf{a}, {}^{S}\mathbf{m}$ Sensor measurements.
Output	$\theta = \arctan(\frac{a_y}{a_z})$ $\phi = \arctan\left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}\right)$ $\psi = \text{arctan2}\big(m_z \sin\phi - m_y \cos\phi,\ m_x \cos\theta + \sin\theta(m_y \sin\phi + m_z \cos\phi)\big)$
Description	Using accelerometer and magnetometer data, an initial orientation in Euler angles can be computed.
Sources	https://ahrs.readthedocs.io/en/latest/filters/tilt.html
Ref. By	R2.

4.2.8 Input Data Constraints

Table 1 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 1 are listed in Table 2.

4.2.9 Properties of a Correct Solution

A correct solution must be a valid orientation representation, calculation results are unit quaternions and can be optionally converted to another attitude representation.

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
Δt	$\Delta t > 0$	$\Delta t_{\text{min}} \leq \Delta t \leq \Delta t_{\text{max}}$	10 ms	-
a	$-\infty < a < \infty$	$\lvert a \rvert \leq a_{\text{max}}$	9.81 m/s^2	See sensor datasheet.
m	$-\infty < m < \infty$	$\lvert m \rvert \leq m_{\text{max}}$	25 µT	See sensor datasheet.
ω	$-\infty < \omega < \infty$	$\lvert \omega \rvert \leq \omega_{\text{max}}$	1.0 rad/s	See sensor datasheet.
γ	-	$0.0 \leq \gamma \leq 1.0$	0.041	-

Table 2: Specification Parameter Values

Var	Value
Δt_{min}	1 ms
Δt_{max}	100000 ms
a_{max}	See sensor datasheet [$\text{m}\,\text{s}^{-2}$].
m_{max}	See sensor datasheet [T].
ω_{max}	See sensor datasheet [$\text{rad}\,\text{s}^{-1}$].

Table 3: Output Variables

Var	Physical Constraints
$\mathbf{q}_t$	$\lvert \mathbf{q}_t \rvert = 1$

5.1 Functional Requirements

R1: The user shall provide system inputs at initialization and every time step as listed below.

Initialization: None are required. Optional parameters are: the gain parameter γ, and the initial orientation.

Each time step: temporally synchronized (accelerometer, magnetometer, gyroscope) measurements or (accelerometer, gyroscope) measurements, and the sensor sampling time Δt.

Refs IM4, IM5.

R2: If an initial orientation is not provided Attitude Check shall use the initial sensor measurements to produce an initial orientation. Refs IM6, IM7.

R3: Attitude Check shall calculate the orientation relative to the Earth frame when provided with accelerometer, gyroscope, and magnetometer measurements. Refs IM5.

R4: Attitude Check shall calculate the the orientation relative to the Earth frame when provided with accelerometer and gyroscope measurements. Refs IM4.

R5: Attitude Check shall output the orientation in quaternion format. Refs IM4, IM5.

5.2 Nonfunctional Requirements

NFR1: **Accuracy** The level of accuracy achieved by Attitude Check shall be described in Section 5.2.1 of the Verification and Validation plan [12].

NFR2: **Understandability**: Attitude Check shall follow standard coding conventions such that the Application programming Interface (API) is clearly understood by a typical user.

NFR3: **Maintainability**: The effort required to make any of the likely changes listed for Attitude Check should be less than η of the original development time.

NFR4: **Portability**: Attitude Check shall compile (if necessary) and run on Ubuntu 20.04, macOS 14 Sonoma, and Win 10/11.

5.3 Rationale

The scope and assumptions in this document reflect realistic assumptions for a real-world sensors and systems. For small drones and robotics applications, the accuracy and quality of the sensors are too low to require accounting for the rotation of the Earth. Typical values reflect values in recreational and hobby drones and robots.

6 Likely Changes

LC1: Detect significant linear accerleration, and ignore accelerometer measurements during large accerleration events. This removes A12.

LC2: Detect siginifiacnt magnetic disturbance and ignore magnetmeter measurements. This removes A10.

7 Unlikely Changes

None.

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 4 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions. Table 5 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 6 shows the dependencies of instance models, requirements, and data constraints on each other.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
TM1				X	X		X	X				
TM2											X	
TM3											X	X
TM4												
TM5												
TM6												
GD1											X	
DD1				X	X	X	X	X				
DD2			X	X	X		X	X	X			
DD3				X	X		X	X		X		
IM1			X	X	X		X	X	X		X	X
IM2	X			X	X	X	X	X				X
IM3	X	X		X	X	X	X	X		X		X
IM4	X		X	X	X	X	X	X	X		X	X
IM5	X	X	X	X	X	X	X	X	X	X	X	X
IM6	X				X	X		X			X	X
IM7	X	X			X	X		X		X	X	X
LC1												X
LC2										X		

Table 4: Traceability Matrix Showing the Connections Between Assumptions and Other Items

	TM1	TM2	TM3	TM4	TM5	TM6	GD1	DD1	DD2	DD3	IM1	IM2	IM3	IM4	IM5	IM6	IM7
TM1																	
TM2																	
TM3		X															
TM4																	
TM5																	
TM6																	
GD1		X															
DD1	X			X													
DD2	X	X															
DD3	X																
IM1	X	X	X				X										
IM2	X			X		X		X									
IM3	X			X		X				X							
IM4	X	X	X	X	X	X	X	X	X		X	X					
IM5	X	X	X	X	X	X	X	X	X	X	X		X				
IM6	X			X				X									
IM7	X			X				X		X							

Table 5: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM2	IM3	IM4	IM5	IM6	IM7	R1	R2	R3	R4	R5
IM1												
IM2												
IM3												X
IM4								X			X	X
IM5								X		X		
IM6									X			
IM7									X			
R1												
R2												
R3												
R4												
R5												

Table 6: Traceability Matrix Showing the Connections Between Requirements and Instance Models

9 Development Plan

N/A.

10 Values of Auxiliary Constants

Table 7: Table of Constants

Var	**Value**
η	$\frac{1}{21}$
$\mathbf{g}_0$	$(0, 0, 9.81)^T$
${}^E\mathbf{b}$	$(16676.8, -3050.9, 49916.9)^T$

Note: Magnetic field values were calculated from the World Magnetic Model (WMM) [7] for Toronto, ON. Other locations may have different values.

References

[1] Attitude representation: Quaternion. https://ahrs.readthedocs.io/en/latest/quaternion/quaternion.html. Accessed: 2024-01-23.

[2] Bing Chat with GPT-4. "can you create a problem statement for a attitude estimation project". https://sl.bing.net/hakggcilB2y. Accessed: 2024-01-16.

[3] Bing Chat with GPT-4. "write a paragraph that descibes th purpose of a software requirements specification for software engineering". https://sl.bing.net/e5UwAYRf5wa. Accessed: 2024-01-23.

[4] Complementary Filter — AHRS 0.3.1 documentation.

[5] Madgwick Orientation Filter. https://ahrs.readthedocs.io/en/latest/filters/madgwick.html, 2023.

[6] Mahoney Orientation Filter. https://ahrs.readthedocs.io/en/latest/filters/mahony.html, 2023.

[7] World Magnetic Model. https://ahrs.readthedocs.io/en/latest/wmm.html#module-ahrs.utils.wmm, 2023.

[8] Hussein Al-Jlailaty and Mohammad M. Mansour. Efficient Attitude Estimators: A Tutorial and Survey, December 2020. arXiv:2012.04075 [cs, eess].

[9] Sebastian O H Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays.

[10] Hossein Nourmohammadi and Jafar Keighobadi. Fuzzy adaptive integration scheme for low-cost sins/gps navigation system. *Mechanical Systems and Signal Processing*, 99:434–449, 01 2018.

[11] Raymond A Serway and John W Jewett. *Physics for scientists and engineers*. Cengage learning, 2018.

[12] Adrian Sochaniwsky. Verification and validation plan for attitude check. https://github.com/adrian-soch/attitude_check/blob/main/docs/VnVPlan/VnVPlan.pdf, 2024.

[13] Wikipedia. Hamilton product. https://en.wikipedia.org/wiki/Quaternion#Hamilton_product, 2023.

[14] Wikipedia. Local tangent plane coordinates. https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates#Local_north,_east,_down_(NED)_coordinates, 2023.

[15] Wikipedia. Earth's magnetic field. https://en.wikipedia.org/wiki/Earth%27s_magnetic_field, 2024.