Research Internship Report

Attitude Determination of the Inspire-sat 7 satellite using Deep Learning Methods

Submitted by

Anthony Kalaydjian
Second year, Applied Mathematics Department
ENSTA Paris

Under the supervision of

LATMOS supervisors		ENSTA Paris referent
Dr. Mustapha Meftah	**Dr. Alain Sarkissian**	**Dr. Andrea Simonetto**
Astrophisicist & Principal Investigator	Assistant Physicist	Research professor

NON CONFIDENTIAL

LATMOS
Laboratoire Atmosphères, Milieux, Observations Spatiales
OVSQ Quartier des Garennes 11 Bld d'Alembert, 78280 GUYANCOURT

May – August 2023

Abstract

With the escalating global concern over climate change, it has never been more critical to accurately quantify its impact. The emergence of cubesats – small satellites equipped with miniaturized sensors – has democratized space access for public research. Leveraging the use of the right sensors, these satellites enable researchers to quantify the Earth Radiative Budget (ERB). Having accurate measurements is crucial. Since the smallest cubesats don't have any active pointing system, knowing their attitude – their orientation with respect to a known reference frame – is fundamental in determining the direction of both Earth and the Sun, and thus identifying the origin of the different radiation fluxes it receives. The following report focuses on the validation of the Multilayer Perceptron (MLP) Deep Learning model to process the signal from the different satellite's sensors in order to accurately compute its attitude. This validation is first done through the exploitation of a test in laboratory, on which an attitude error of 4° at 1σ is retrieved. A simulation is also implemented in order to generate more data and compare this new techniques with the currently used one. In the shadow zone, the MLP provides an error with a 2.24° standard deviation around a 1.29° mean, which outperforms the classic method providing a 6° standard deviation around 8° mean.

Acknowledgements

Foremost, I would like to extend my heartfelt appreciation to my supervisor, Dr. Mustapha Meftah for his insightful feedback throughout this project, and with whom I have gained invaluable knowledge. Special thanks to Dr. Alain Sarkissian who trustfully welcomed me among the team and on this exciting project.

My gratitude extends to Christophe Dufour, whose thoughtful insights enriched the depth of my work.

I am also grateful to my laboratory peers, whose feedback and encouragement bolstered both my work and spirit. A special tribute goes to Nicolas Caignard, Cannelle Clavier, Louis Dechaseaux, and Angèle Minet for their unwavering support.

Finally, I would like to thank my family for their unending support throughout the whole duration of this project.

Contents

Introduction

Climate change has undoubtedly emerged as one of the most pressing concerns shaping the course of the 21st century. The looming danger of global warming is a strong threat to humanity and is felt throughout the whole globe. Understanding the dynamics of climate change, and quantifying its evolution is a critical task. Whether it is on a global or regional scale, it has become urgent to measure the variables of climate change.

With the technological advancements of space technology, research laboratories have become able to launch exponentially more satellites. Capitalizing on the miniaturization of sensors and actuators and on the depreciating cost of Space launches, Scientists are now able to build low-cost Rubik's-cube sized satellites, called cubesats equipped with a multitude of sensors they wish to use on each face. With carefully chosen sensors and with the ability to launch a constellation of small satellites, these new tools have become a serious way to measure, with high global and regional precision, the Earth Radiative Budget (ERB) which quantifies the energy imbalance between the radiation that the Earth receives from the Sun, and the radiation it emits back in Space.

Knowing the orientation of a cubesat with respect to a given reference frame – its so called attitude – is an essential step in order to have more accurate measurements of the ERB. This is the case for the two general types of satellites that exist: for satellites with active pointing systems to accurately point towards the Earth, or any object of interest but also for satellites which do not have such systems, in order to identify the position of the Sun and the Earth in the satellite's body frame, and thus uncouple the different radiation fluxes received by its sensors.

This project will focus on Inspire-sat 7, a 2U cubesat (1U = 10cm x 10cm x 10cm) lauched by LATMOS in April 15th 2023 and whose goal is to measure the ERB at the Top Of the Atmosphere (TOA) and to create the first LATMOS satellite constellation, with its older counterpart UVSQ-sat. Not equipped with an active pointing system, determining the attitude of the satellite will allow to have more accurate measurements of the ERB. As classic Attitude Determination (AD) algorithms have their pros and cons, the goal of this report is to evaluate the performance of new Deep Learning (DL) methods to retrieve the attitude of Inspire-sat 7. The first section will be an introduction to the AD problem, outlining a few algorithms and presenting the research results of the field. The second part will then present the Multilayer Perceptron DL algorithm and introduce the problematic. The final part will apply this algorithm to two problems, first the data of a test made in laboratory at the LATMOS, and then a simulation which has also been implemented.

1 Attitude Determination Theoretical Outline

Before diving into the different attitude determination algorithms, it is important to parameterize this said attitude and choose an appropriate representation system.

1.1 Attitude representation

There exists multiple attitude representation systems. The most famous, Euler angles can be visualized intuitively as they represent the angles of the successive composition of 3 elementary rotations along three axis. A bunch of elementary rotation sequences exist, the most famous one being the Yaw, Pitch and Roll (YPR) sequence, which successively applies rotations along the X, Y and Z axis.

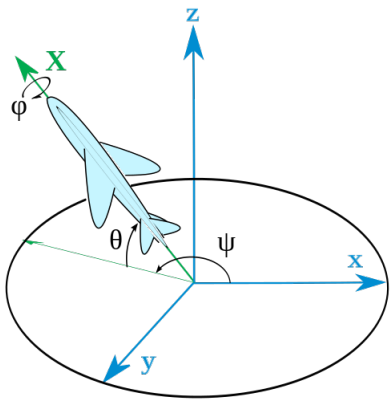


Figure 1: Yaw, Pitch, Roll (YPR) Euler sequence, figure taken from Juansempere, https://en.wikipedia.org/wiki/Euler_angles

Another commonly used representation system is the Direction Cosine Matrix (DCM). A DCM is a rotation matrix representing the rotation of a frame with respect to a reference frame. Euler angles can be converted to DCM using the following equation (for the YPR sequence):

$$R_{zyx} = R_x(\phi) \cdot R_y(\theta) \cdot R_z(\psi) \tag{1}$$

With:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

But Euler angles have a few issues. The most important being a singularity issue when the second angle of the Euler sequence is 90°.

An other system which eliminates this issue is quaternions. Any rotation in space can be represented by a unit quaternion (see appendix A). Besides eliminating the singularity issue, it is also shown in the appendix that successive rotations are easily represented by the quaternionic product, of which there is no equivalent in Euler's system.

1.2 Literature survey

There exists two types of algorithms used to estimate a satellite's attitude: direct methods and sequential ones. Direct methods rely solely on instantaneous sensor data to compute the satellite's attitude, while sequential methods go beyond and fuse these instantaneous data with historical attitude estimates.

1.2.1 Direct methods

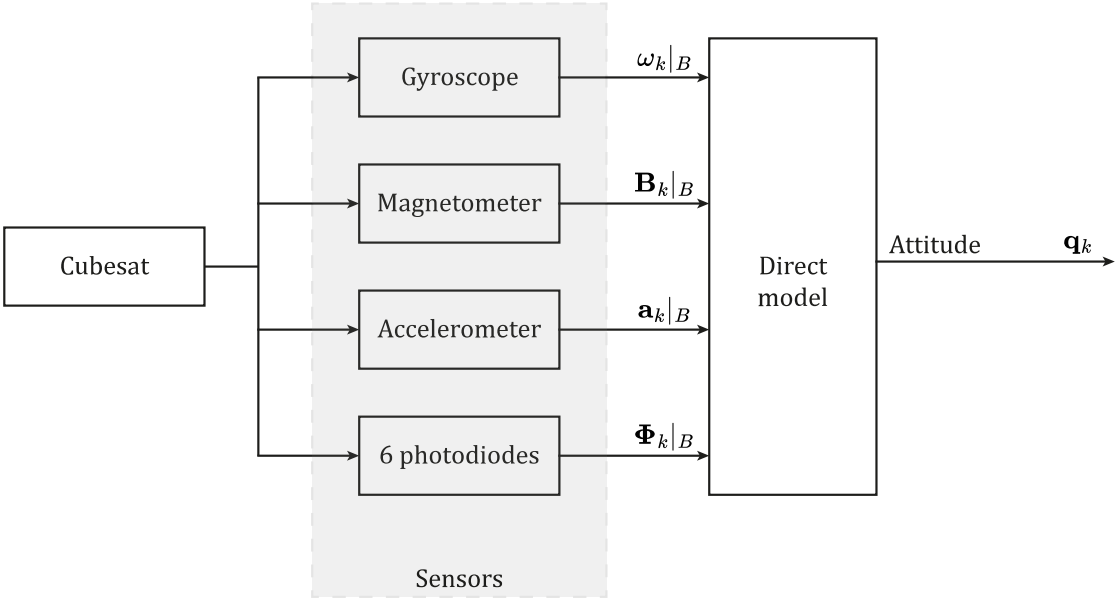


Figure 2: Direct method

A first algorithm, called TRIAD [4] is a direct algorithm that relies on the knowledge of 2 vectors in the body and reference frames to compute the attitude of the satellite.

When more vectors measurements are available, Attitude Determination (AD) can be rewritten as an optimisation problem, finding the rotation matrix that minimises the following cost function :

$$\underset{\mathcal{R}\in SO_3}{\operatorname{argmin}} \quad J(\mathcal{R}) = \frac{1}{2}\sum_{i=1}^{I} w_i ||\vec{v_i}^I - \mathcal{R}\vec{v_i}^B||^2 \tag{2}$$

Where $\vec{v_i}^I$ is the ith vector known in the Inertial reference frame, $\vec{v_i}^B$ is the ith vector in the satellite body frame measured thanks to the satellite's sensors, and $(w_i)_{i\in[\![1,I]\!]}$ is a barycentric sequence of weights giving more weight to the terms whose measurements have better accuracy.

This problem is known as Wahba's problem and has been developed in [5]. Analytical solutions to this problem were found such as the Singular Value Decomposition (SVD) methods, the Fast Optimal Attitude Matrix (FOAM) algorithm in [6], or using the quaternion representation as in Davenport's Q-method [7].

Direct methods based on image processing have also been developed lately. For instance, [8] makes use of the surface pattern on Earth to estimate the satellite's attitude and is able to reach an accuracy of 0.1-1° when the satellite camera's Field of View (FOV) contains land areas of Earth and the satellite is by the enlightened side of Earth.

A similar method is shown in [9], but this time only making use of the overall shape of the celestial body of reference, which projected on the image plane of the satellite's camera, is a conic.

While current direct methods provide good responsiveness, they are very sensitive to measurement errors. Sequential methods provide more stability in the estimates.

1.2.2 Kalman Filters

Most of the sequential methods developed until now are based on a form of Kalman Filter (KF).

Kalman filters rely on a two-step process where the state vector – containing for instance the attitude of the satellite – is updated by first predicting its value given the system dynamics' knowledge and the previous state estimates, and then correcting this prediction using the current measurements of the sensors. The regular KF has been introduced in appendix B.

While the initial Kalman Filter is suited for linear systems, The attitude determination problem requires its extended version to accommodate the non-linearity of its dynamics.

The Extended Kalman Filter (EKF) has thus been the seed of many algorithms created since to estimate attitude.

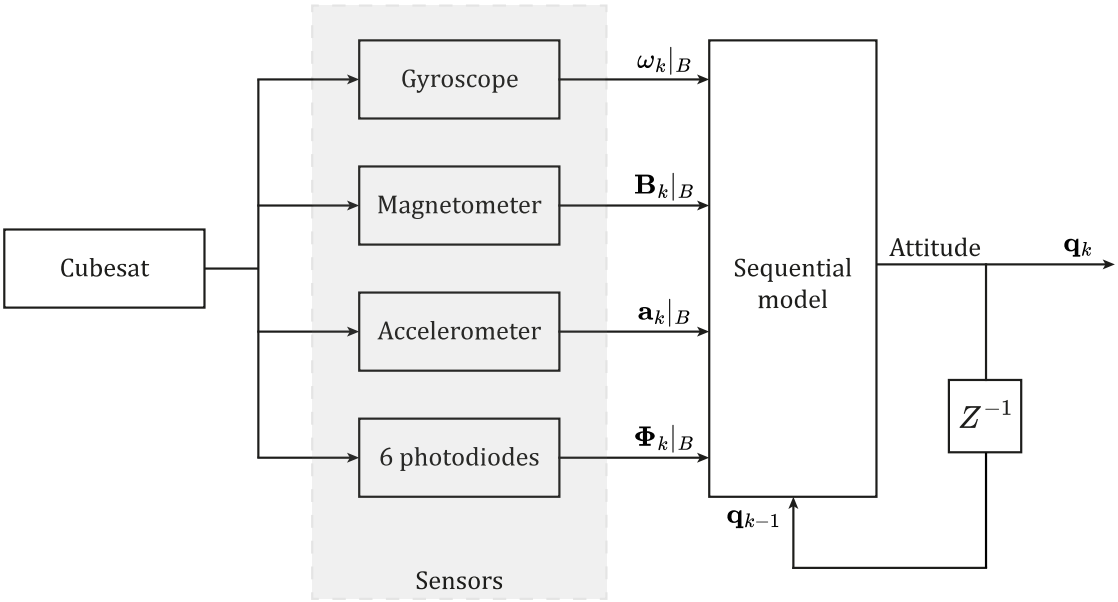


Figure 3: Sequential method

While the original EKF is additive, i.e. it updates the state vector at each step via summation, a multiplicative variant, the Multiplicative Extended Kalman Filter (MEKF) variant [3] updates the state vector using the quaternionic product. This way, the updating is able to preserve the attitude quaternion's norm, without needing an extra normalization step.

It can be noted that while most attitude methods rely on the same set of sensors, e.g. magnetometer, sun sensors and Inertial Measurement Units (IMU), a few papers have tested the use of more peculiar sensors, such as gravity gradient sensors in [10] or electrostatically Suspended Gyroscopes in [11].

A table commenting on a few conducted research papers is provided in table 1

While current direct methods provide good responsiveness, they are very sensitive to measurement errors. Sequential methods on the other hand provide more stable estimates, but are softer.

1.2.3 Currently used AD algorithms onboard Inspire-sat 7 and UVSQ-sat

The algorithm currently being used on the UVSQ-sat and Inspire-sat 7 missions is the TRIAD method. It is a direct method which computes the attitude of the satellite given its instantaneous sensor measurements.

Ref	Description	Method	Result
[?]	Multiplicative Unscented Filter (MUF) for AD without accurate gyro-measurements	Simulation	With reasonable initial estimates, MUF converges after a few steps to reach an error below 0.5°. When the initial estimates are less accurate, MUF is able to converge faster than MEKF
[10]	MEKF using a spaceborne gravity gradient sensor	Simulation & Test data	The algorithm converges in about 200 steps at a 1Hz sampling rate, and has an error of less than 100 arcsec at 3-sigmas
[12]	Comparison between gyro-less and gyro-based algorithms	Simulation	gyro-based algorithms perform better, especially when the sampling frequency increases
[13]	UKF using solar panels as Sun sensors	Simulation	1.1° precision in Euler angles. More computationally heavy compared to EKF, but 2 times better performance. Unreliable during Eclipses
[14]	AD using Adaptative EKF	Simulation	AEKF outperforms the SVD-EKF filter in the eclipse phase, with a maximum Euler angle error of 0.3° compared to the 12.7° error of SVD-EKF. It also slightly outperforms it in the Sunlight.
[11]	AD KF using Electrostatically Suspended Gyroscopes (ESG)	Simulation	0.005° error after 200 steps
[15]	Star tracker & MEKF	Test using the night sky	0.0203° error at 1-sigma
[16]	Decomposed Kalman filter: Decomposition of the KF into two filters: One for estimating the angular velocity, and one for estimating the attitude quaternion. A Star Tracker (ST) is also used	Simulation	Time invariant error dynamics. The proposed algorithm is superior until the ST's sampling time is superior to 500ms. Even with a sampling time of 125s, the algorithm is reaches an error median of just 0.005 rad
[17]	Improved Cubature Kalman filter. The Adaptive Huber filter based on the Multiple Strong Tracking (AHMFST) algorithm adresses the non truly gaussian aspect of measurement errors noises	Simulation in 5 scenari	AHFMST shows big improvements compared to MEKF and has better resilience than the other algorithms with increased errors, and is able to provide estimates with errors as low as 0.0009° at 1-sigma
[18]	AD Using EKF and only geomagnetic field data	Simulation	The filter converges for any initial attitude error and reaches 8° of accuracy in less than one and a half orbit
[?]	Particle Filter (PF) & EKF comparison	Simulation	The PF has superior attitude estimation accuracy compared to EKF, with a 0.01° accuracy for attitude estimation
[19]	QUatertion ESTimation (QUEST) aided EKF. Reduces the calculation cost of EKF by inputting a coarse attitude estimate with QUEST into EKF.	Simulation	0.78° error after 500 monte carlo simulations average
[2]	A new MLP method for attitude determination. The algorithm was trained on the TRIAD attitude estimats of UVSQ-sat's on orbit data.	On-orbit data	Because the true attitude is not known in orbit, training it on TRIAD estimates was conducted. But this is not optimal.

Table 1: Overview of recent research results in the AD field

TRIAD method [4]

The TRIAD method is a direct algorithm which solely makes use of the magnetometer and radiation sensors of the satellite.

It uses the sensor measurements to compute the attitude of the satellite with respect to an inertial frame of reference.

Given that the satellite's sensors allow to compute both the Sun's direction as well as the magnetic field's direction in the satellite's body frame, and that there are models that allow to retrieve these same two vectors in the inertial frame of reference, the TRIAD makes use of a third frame (hence the name TRIAD) to compute the transformation from the inertial to the body frame.

The vectors in the inertial frame of reference can be computed given the knowledge of the satellite's position in the orbit, with geomagnetic reference maps for the magnetic field, and with the Sun's known position for the Sun's Line Of Sight (LOS), i.e. the Sun's direction vector.

Algebraically speaking, given the notations of table 2 where $\vec{s}$ refers to the Sun's LOS and $\vec{m}$ refers to the magnetic field vector, the triad basis can be constructed.

Sensors	Model
$\vec{s}^{\,B}$	$\vec{s}^{\,I}$
$\vec{m}^{B}$	$\vec{m}^{I}$

Table 2: TRIAD available data

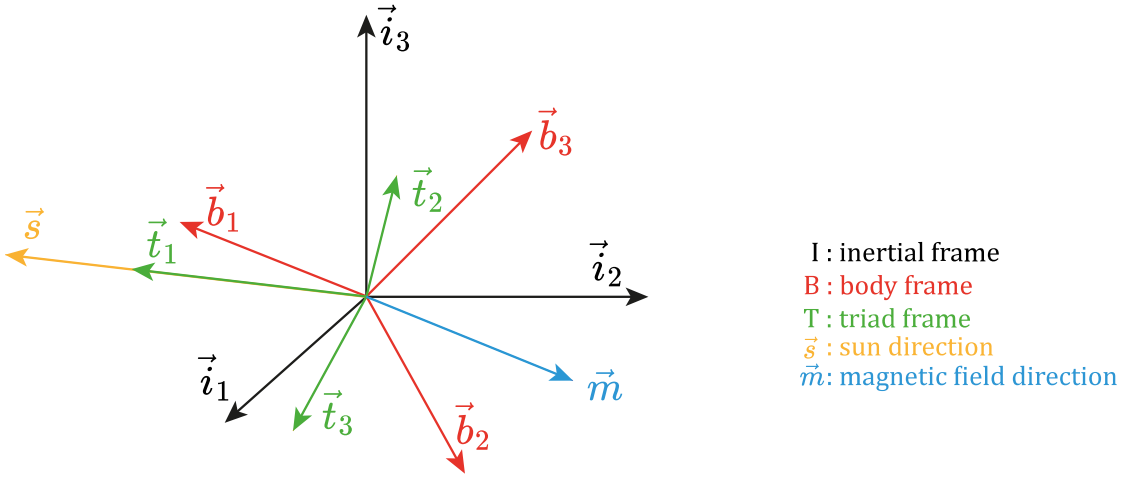


Figure 4: TRIAD method

The TRIAD basis vectors can be expressed in the Body and Inertial frames of reference as follows :

Body frame :	Inertial frame :
$\vec{t_1}^B := \frac{\vec{s}^{\,B}}{\|\vec{s}^{\,B}\|}$	$\vec{t_1}^I := \frac{\vec{s}^{\,I}}{\|\vec{s}^{\,I}\|}$
$\vec{t_2}^B := \frac{\vec{s}^{\,B} \times \vec{m}^{\,B}}{\|\vec{s}^{\,B} \times \vec{m}^{\,B}\|}$	$\vec{t_2}^I := \frac{\vec{s}^{\,I} \times \vec{m}^{\,I}}{\|\vec{s}^{\,I} \times \vec{m}^{\,I}\|}$
$\vec{t_3}^B := \vec{t_1}^B \times \vec{t_2}^B$	$\vec{t_3}^I := \vec{t_1}^I \times \vec{t_2}^I$

From these 6 vectors, one can compute the transition matrices between the body frame and the triad frame, as well as the one between the inertial frame and the triad frame.

$$\mathrm{P}_B^T := \begin{bmatrix} \vec{t_1}^B & \vec{t_2}^B & \vec{t_3}^B \end{bmatrix} \qquad \mathrm{P}_I^T := \begin{bmatrix} \vec{t_1}^I & \vec{t_2}^I & \vec{t_3}^I \end{bmatrix}$$

It follows:

$$\begin{aligned} \mathcal{P}_I^B &= \mathcal{P}_I^T \mathcal{P}_T^B \\ \mathcal{P}_I^B &= \mathcal{P}_I^T \left(\mathcal{P}_B^T\right)^T \end{aligned}$$

The attitude matrix of the satellite is finally defined as the transition matrix between the Body frame of reference and the inertial frame of reference.

$$\boxed{A := \mathcal{P}_I^B = \begin{bmatrix} \vec{t_1}^I & \vec{t_2}^I & \vec{t_3}^I \end{bmatrix} \begin{bmatrix} \vec{t_1}^B & \vec{t_2}^B & \vec{t_3}^B \end{bmatrix}^T} \tag{3}$$

This matrix is only an estimate, as the measurements from our different sensors are noisy, and feed noise forward into the calculations of the TRIAD basis vectors.

Having access to the real attitude of the satellite, say for instance when dealing with a simulation, enables the computation of the angular error between the estimated attitude matrix A and the real attitude matrix A_r.

Since A and A_r are both in the $SO(3)$ group, $A^T A_r$ is also in $SO(3)$. Thus, there exists a particular basis in which this matrix can be written as follows:

$$A^T A_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4}$$

Similar to the rotation quaternion representation, this matrix tells how any rotation matrix can be interpreted as a single rotation along a unitary axis. The angle of rotation along this axis is given by θ.

If the estimate is exact, one would have $A = A_r$, thus this matrix would be the identity matrix. The error angle, θ can be computed as follows, since the trace of a matrix is independent of the basis the matrix is represented in :

$$\boxed{\theta = \arccos\left[\frac{1}{2}\left(\mathrm{Tr}(A^T A_r) - 1\right)\right]} \tag{5}$$

When the satellite is in the shadow zone of Earth, the sun LOS determination is not possible, but the sun vector can be switched to the nadir vector, which can be estimated in the satellite's Body frame using the Earth Radiative Sensors equipped with Optical Sun Reflector (ERS with OSR), which sense the IR outgoing Earth radiation.

This still yields a consequent amount of error. Indeed, it is much easier to measure the Sun's LOS, which can be considered as emitted by a point source. The Earth being much closer to the satellite than the Sun, measurements of its IR radiation less accurately provide the nadir direction.

Optimized TRIAD

It has just been shown that the TRIAD algorithm chooses a particular vector as its first vector basis. In order to gain accuracy, one may consider also using the other vector as first TRIAD vector, and fuse the two obtained attitude matrices estimates.

Given the error precision of each, the fused attitude matrix can be written as explained in [20]:

$$A^* := \frac{\sigma_s^2}{\sigma_s^2 + \sigma_m^2} A_s + \frac{\sigma_m^2}{\sigma_s^2 + \sigma_m^2} A_m$$

This matrix is nearly orthogonal but can be orthogonalized as follows :

$$\boxed{A = \frac{1}{2}\left[A^* + \left(A^{*-1}\right)^T\right]} \tag{6}$$

The TRIAD method's accuracy has been quantified in [21], with a 3° uncertainty when the satellite is in the Sunlight and 14° of uncertainty when it is in the shadow zone.

The main drawback of this method is its imprecision when switching from one zone to the other.

2 Attitude Determination with a Multilayer Perceptron

As the different techniques for attitude determination have been presented, the purpose of the next sections is to evaluate the feasibility and performance of a ML-based attitude determination algorithm for Inspire-sat 7, the Multilayer Perceptron (MLP).

2.1 Multilayer Perceptron [1, 2]

The Multilayer Perceptron, illustrated in fig. 5, is a type of direct algorithm, used for regression and classification tasks.

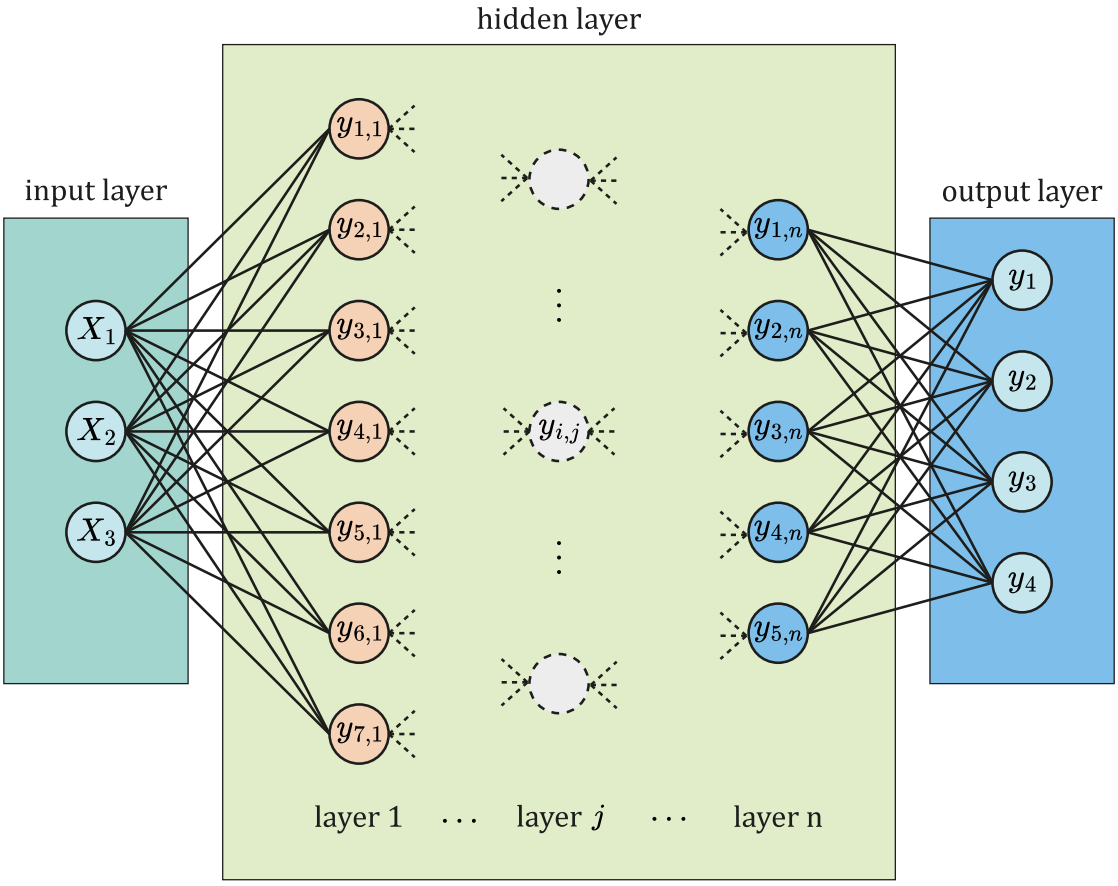


Figure 5: MLP with n hidden layers for a size 3 input layer and size 4 output layer

The MLP network is able to learn from the data it is provided. It is a function whose parameters are optimized with a training process so it minimizes the value

of a certain loss function, which quantifies the quality of the model on the sample training data. Given a training input vector sequence (X^k) and its relative output vector sequence (y^k), the training's goal is to find the best MLP network (from the whole MLP class) which best maps the inputs to the outputs and is also able to correctly map new input data to their real output.

The MLP is constituted of numerous layers and neurons. Given a MLP of n hidden layers, the relation between each neuron is given using:

$$y_{i,j} = g\left(\sum_{k=1}^{n_{j-1}} w_{k,j-1} y_{k,j-1} + b_{j-1}\right) \tag{7}$$

Where $y_{i,j}$ is the i$^{\text{th}}$ neuron of the j$^{\text{th}}$ layer (by convention $y_{i,0} = X_i$ and $y_{i,n+1} = y_i$), w are the weights, b the bias term, n_j is the size of the j$^{\text{th}}$ layer and g is the activation function which breaks the linearity of the algorithm. Typically, g is the Rectified Linear Unit (ReLU) function from eq. (8).

$$\text{ReLU}(x) := \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \tag{8}$$

The training is done through the minimization of a loss function. The classical loss function is the Mean Squared Error (MSE):

$$\mathcal{L}((\hat{y}_k), (y_k)) := \frac{1}{N} \sum_{i=1}^{N} ||\hat{y}_i - y_i||^2 \tag{9}$$

Where $\hat{y}_k = \text{MLP}(X^k)$ and N is the number of sample elements. This function is optimized with respect to the MLP's weights and biases. This optimization is done with the backpropagation algorithm, and gradient descent.

When it comes to attitude estimation, the inputs of the MLP will naturally be all of the satellite sensors' measurements, as well as the position of the satellite in space and the different vector values in the reference frame. The output will be the attitude quaternion.

Such an algorithm had already been tested by Finance et al, but on the in-orbit data from the previously launched UVSQ-sat satellite. Given this, the algorithm was trained to retrieve the attitude estimates of the TRIAD method, and not the true attitude of the satellite. Hence, rather than finding the best model for

estimating the attitude of the satellite, this algorithm searched to reproduce the TRIAD method.

2.2 Contribution description

The following section will tackle two problems. First, the exploitation of a ground-test made on the newly launched Inspire-sat 7. And then the implementation and exploitation of a simulation made to reproduce the in orbit conditions of the satellite.

Both tests will verify the ability of a Multilayer Perceptron (MLP) to retrieve the attitude of the satellite.

Listing everything done, my contribution to the project is the following:

- Exploitation of the raw test dataset, using a MLP.
- Implementation of a simulation, modeling the behaviour of Inspire-sat 7 in orbit.
- Implementation of a simple graphical visualization of the simulated datasets.
- Exploitation of the simulation's results.

3 Contribution

3.1 Real test data exploitation

3.1.1 Description of the test bench

A specific test designed to evaluate the responsiveness of the different radiation sensors of Inspire-sat 7 was conducted in laboratory before its launch.

The satellite was put on a rotating platform along its z-axis, making it rotate at a low angular rate. A light source was also placed near the satellite to excite its different sensors. The sensor measurements were recorded for a few revolutions, and were retrieved in the form of an excel sheet.

The data from this test can be used to retrieve the satellite's attitude. In this controlled environment, the attitude of the satellite can be represented by the angle of the satellite's body frame with respect to the light source. Given that the position of the light source is unknown, this angle could still be known, at a constant, by using the magnetometer data, as the measured magnetic field always points towards a fixed direction.

The attitude determination algorithm could then be trained to calculate the computed angle using every sensor measurement but the magnetometer data as input features.

3.1.2 Dataset exploitation

Data pre-processing

The raw form of the dataset is not friendly to use. The first important step of the pre-processing is to match the data from the different sensors as they don't necessarily have the same sampling frequency, and don't sample at the same timestamps. The different variable values are identified using a maximum delta of 5 timestamps. The remaining missing data can be interpolated with respect to time.

The discontinuity in the magnetic field data, as seen in fig. 6, which is due to the encoding of the data are then adressed.

Once this is done, the rows of aberrant magnetic field measurements can be deleted from the dataset, and the features can be standardised. The final size of the dataset is noted N. Projecting the magnetic field in the X-Y plane shows it traces an ellipse. A calibration of the magnetometer data is then performed, first fitting an ellipse to the data samples using a least square approach.

Given M^x and M^y, two row vectors containing the components of the magnetic

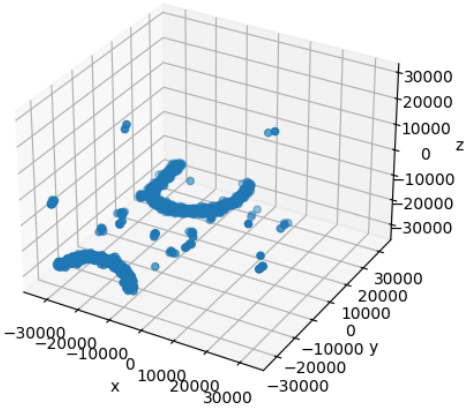


Figure 6: Brute magnetometer data

field in the X-Y plane at each timestamp, the coefficients of the fitting ellipse of equation $ax^2 + bxy + cy^2 + dx + ey + 1 = 0$ are calculated by solving the problem:

$$\underset{x \in \mathbb{R}^5}{\operatorname{argmin}} \, ||Ax - b||^2 \tag{10}$$

Where:

$$A := \begin{bmatrix} M^x \odot M^x \\ M^x \odot M^y \\ M^y \odot M^y \\ M^x \\ M^y \end{bmatrix}^T \in M_{N,5}(\mathbb{R})^1 \qquad b = -1 \in \mathbb{R}^N \tag{11}$$

The data-points can then be transformed into a circle by first centering the ellipse, rotating it so its semi major axis is aligned with the X-axis, and scaling each point's x and y coordinate appropriately, as seen in fig. 7 where the green shape is the unit disk.

Once the magnetometer is calibrated, the angle of the magnetic field in the satellite's body frame at index k can be retrieved using :

$$\theta_k = \arctan\left(\frac{M_k^y}{M_k^x}\right) \qquad \forall k \in [\![1, N]\!] \tag{12}$$

[1] $\odot$ is the element-wise product

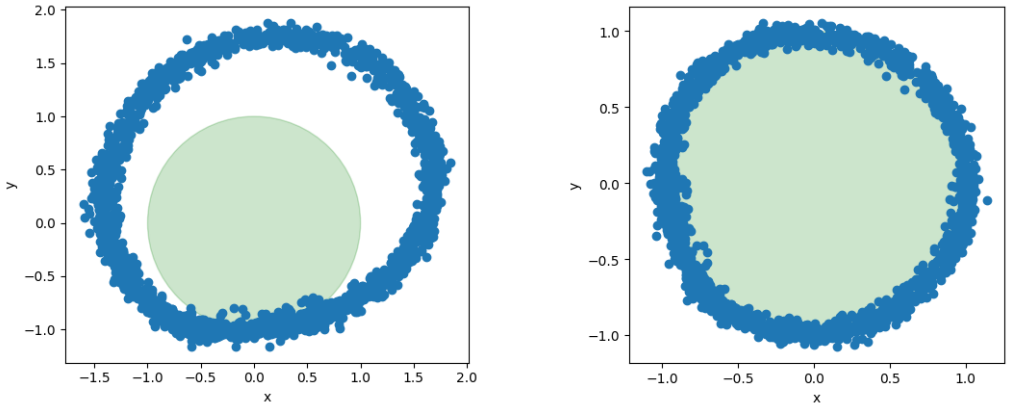


Figure 7: Magnetometer calibration

Looking at fig. 8 confirms how the rotation rate was mostly constant, as it is piecewise linear, except for a time interval just before step 1000, where the angle stops. This could be due to a momentary failure of the motor which could be explained as it was driven at a low voltage.

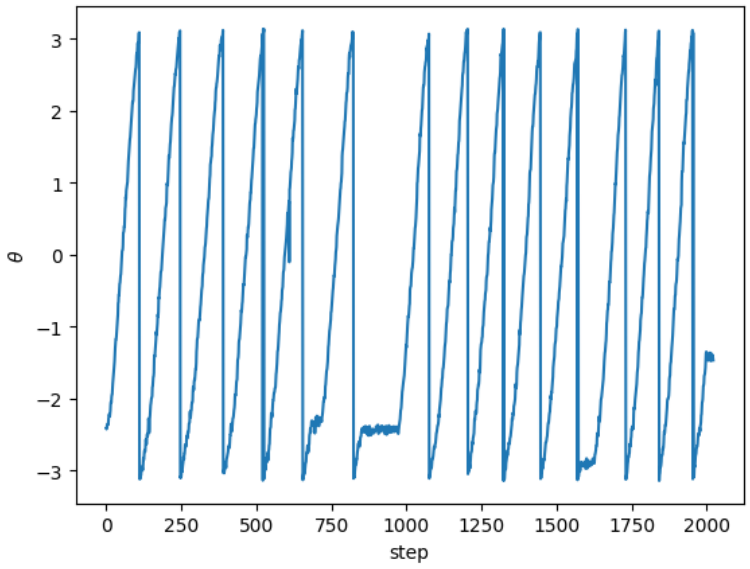


Figure 8: Attitude angle

Multivariate analysis

A few variables could be eliminated right off. First the ISIS sensors – a type of radiation sensors – which do not bring sufficient information, as the light source was not powerful enough.

Temperature data can also be eliminated as convection of the air changes the sensor's measurements, while it could be used in Space, as it holds information about the incoming radiation.

Plotting the correlation matrix in fig. 10a of the variables mostly correlated with the angle θ interestingly shows that the sensor measurements along the X axis of the satellite are more correlated to the satellite's attitude than the ones along the Y axis.

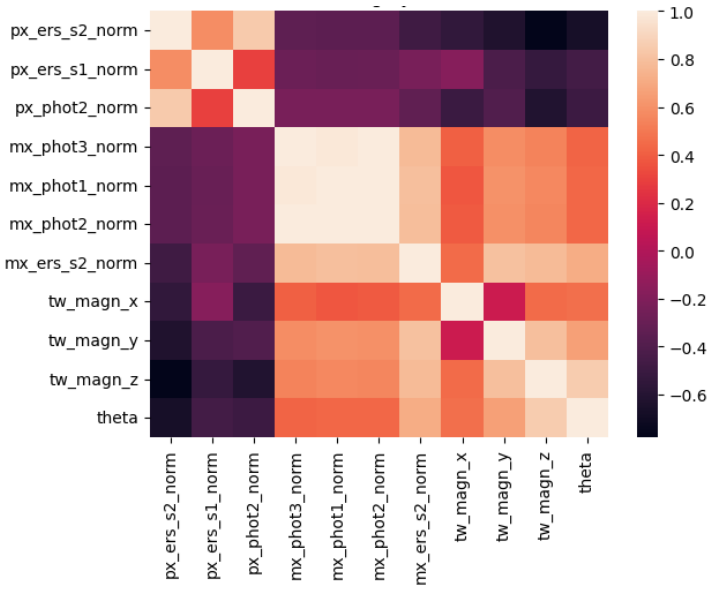


Figure 9: Correlation matrix of the variables most correlated with θ

Attitude estimation

Attitude estimation is done via a Multilayer Perceptron.

To evaluate the model properly, the dataset was split into three parts:

- A training set – 70% of the dataset
- A testing set, used to choose the right model – 15% of the dataset
- A validation set, used to validate the generalization of the chosen algorithm to new data – 15% of the dataset

Rather than training the neural network directly on the angle, which does not take into account the discontinuity at $\theta = \pm 2\pi$, the training output was encoded as follows : $y = \begin{bmatrix}\sin(\theta) & \cos(\theta)\end{bmatrix}$. This new output format does not suffer from the discontinuity, and after training, the estimated output angle can then be easily retrieved with:

$$\hat{\theta} = \arctan\left(\frac{\hat{y}_1}{\hat{y}_2}\right) \tag{13}$$

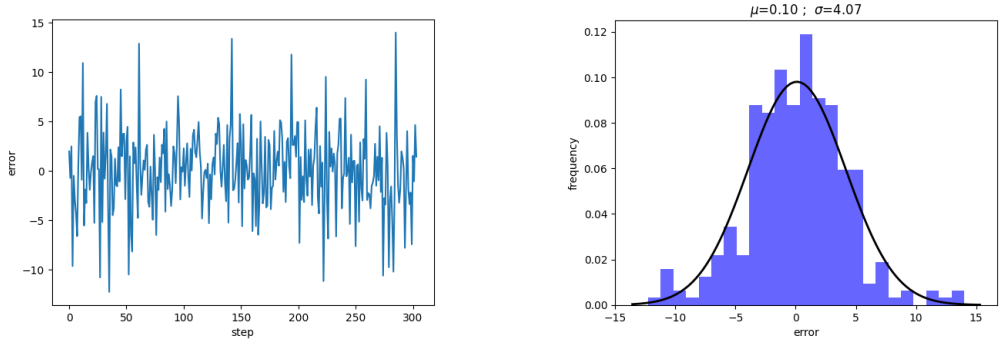
Where $\hat{y}$ is the output estimation from the model.

The loss function has also been modified, adding the the regular MSE a normalization term:

$$\mathcal{L}\left((\theta_k), (\hat{\theta}_k)\right) := \frac{1}{N}\sum_{i=1}^{N} |\theta_i - \hat{\theta}_i|^2 + \beta|1 - \hat{\theta}_i|^2 \tag{14}$$

Where $\hat{\theta}_k$ is an estimate of θ_k $\forall k \in [\![1, N]\!]$ with N being the number of datapoints, and β is an arbitrary positive number.

The right model was chosen after training a bunch of models, choosing the one that minimised the test error. The residual of this model on the validation set has been plotted in fig. 10.



(a) Estimation residuals

(b) Histogram of the estimation residuals

Figure 10: Error plots

The estimation residual seems gaussian, which is confirmed by the plotted histogram in fig. 10b. The trained model is able to reach an error of 4.07 at 1-sigma.

This result is satisfactory especially when comparing the conditions of the test with orbital conditions, as with the closeness of the light source and the lack of usage of numerous sensors.

Although the results obtained are promising, the nature of the dataset, conducted on a rotation along a single axis, has constrained the MLP to only work in this particular setting. Thus, it is not able to generalize to rotations along other axis.

Being able to generate massive amounts of data to test this new method is thus relevant. The following section will focus on the modelisation and implementation of a simulation of Inspire-sat 7 in a Low Earth Orbit (LEO).

3.2 Simulation

3.2.1 Modeling the system

The modelisation of the system is done in a few different steps. First the kinematics of the satellite (translation and rotation) and then the modelisation of the satellite's sensors (radiation sensors & magnetometer).

The satellite's motion can be split into two parts. A first part that describes the movement of the satellite's center of mass, and a second that describes the satellite's rotation. Given that the satellite is in Space, these two movements don't interact, and can be computed separetly and composed together later on.

Center of mass kinematics

The Inspire-sat 7 satellite's orbit is a sun-synchronous Low Earth Orbit (LEO), as seen in fig. 14, meaning that the orbit plane of the satellite stays aligned with the vector pointing from Earth's center of mass to the Sun. This means that while the Earth rotates arround the Sun over the year, the satellite's orbit also rotates, but in the reverse direction and at a rate compensating for Earth's rotation.

Given that the satellite's center of mass kinematics are not particularly useful for attitude determination, the motion of the satellite will simply be modelled as a rotation at a constant rate in the orbital plane, which itself rotates along its z axis.

Rotational dynamics

The rotational dynamics are important, as they not as easy as simple rotations along a certain axis. They are also useful for the gyroscope's data, and should be modeled more accurately. As such, they will be integrated using Euler's rotation equations:

$$\begin{aligned}(I_z - I_y)\omega_2\omega_3 + I_x\dot{\omega}_1 &= 0\\ (I_x - I_z)\omega_3\omega_1 + I_y\dot{\omega}_2 &= 0\\ (I_y - I_x)\omega_1\omega_2 + I_3\dot{\omega}_3 &= 0\end{aligned} \tag{15}$$

Where I_x, I_y and I_z are the inertial moments of the satellite in its orthogonal principal axes of inertia coordinates, drawn in fig. 12.

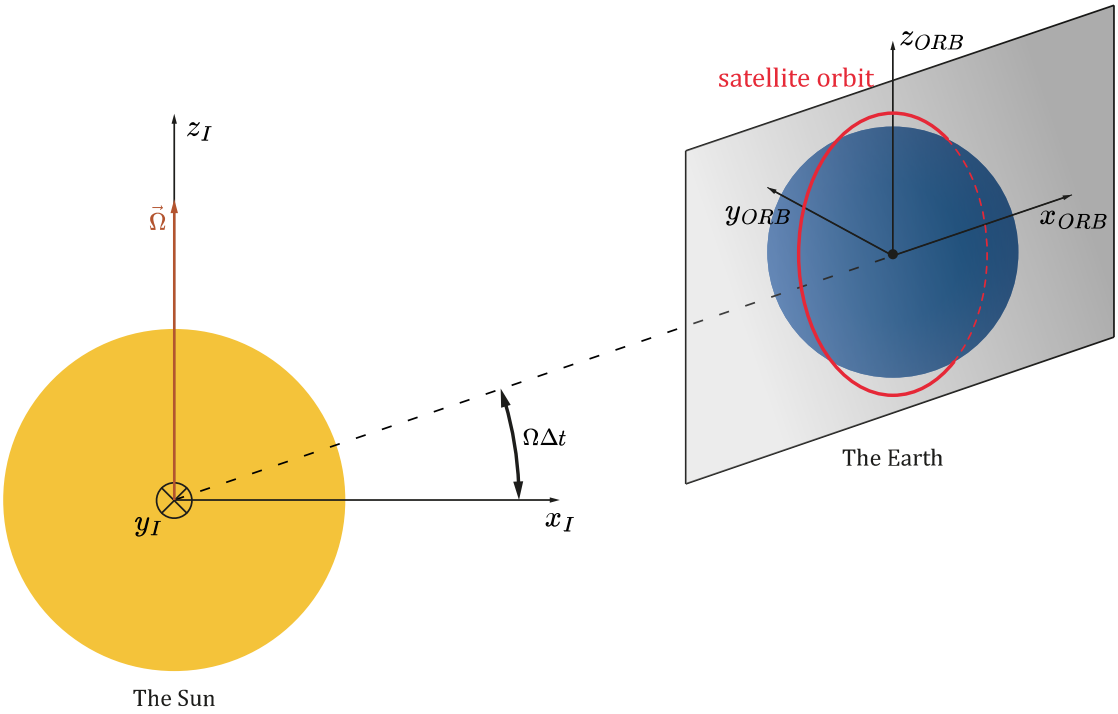


Figure 11: Orbital frame

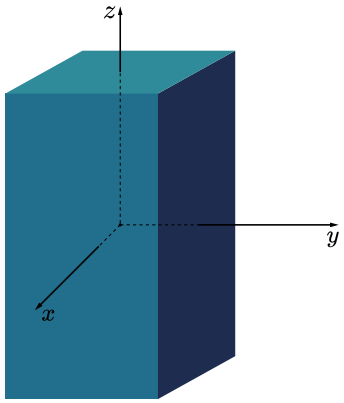


Figure 12: Satellite body frame

For inspire-sat 7, the inertial moments are the following :

$$\begin{cases} I_x = 0.0123 \text{ kg m}^2 \\ I_y = 0.0119 \text{ kg m}^2 \\ I_z = 0.0046 \text{ kg m}^2 \end{cases}$$

Integrating eq. (15) provides a sequence of rotation rates $(\omega_k)_{k\in[\![1,N_{iter}]\!]}$. the atti-

tude quaternion of the satellite can then be updated using eq. (34) at each time-step:

$$q_{k+1} = q_k + \frac{1}{2} dt \Omega_k q_k$$

Earth magnetic field modelisation

The Earth's magnetic field is modelled as a magnetic dipole whose magnetic moment is directed from the magnetic South to the magnetic North.

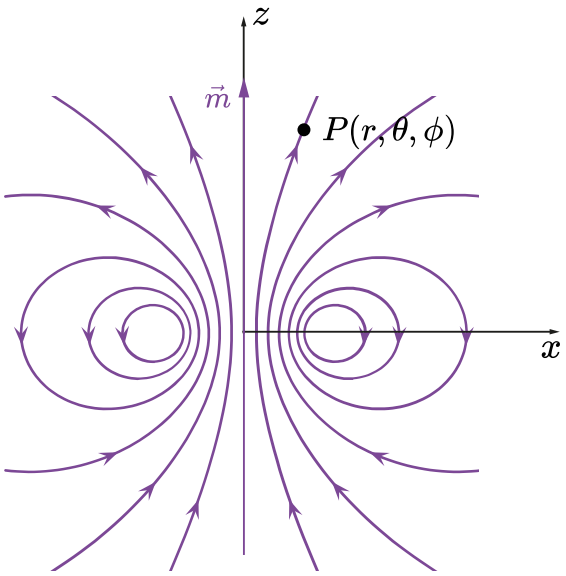


Figure 13: Magnetic dipole

As such, the equations used to calculate the magnetic field situated at a point $P(r, \theta, \phi)$ of space in spherical coordinates are the following :

$$\begin{cases} B_r = \frac{\mu_0 M \cos(\theta)}{2\pi r^3} \\ B_\theta = \frac{\mu_0 M \sin(\theta)}{4\pi r^3} \\ B_\phi = 0 \end{cases}$$

But one must be careful as the magnetic North pole is not aligned with the rotation axis of earth, which is itself tilted with respect to the ecliptic plane's normal.

In the Earth Central Inertial frame (ECI), the magnetic moment moves forming a cone centered on Earth, whose angle is 11.

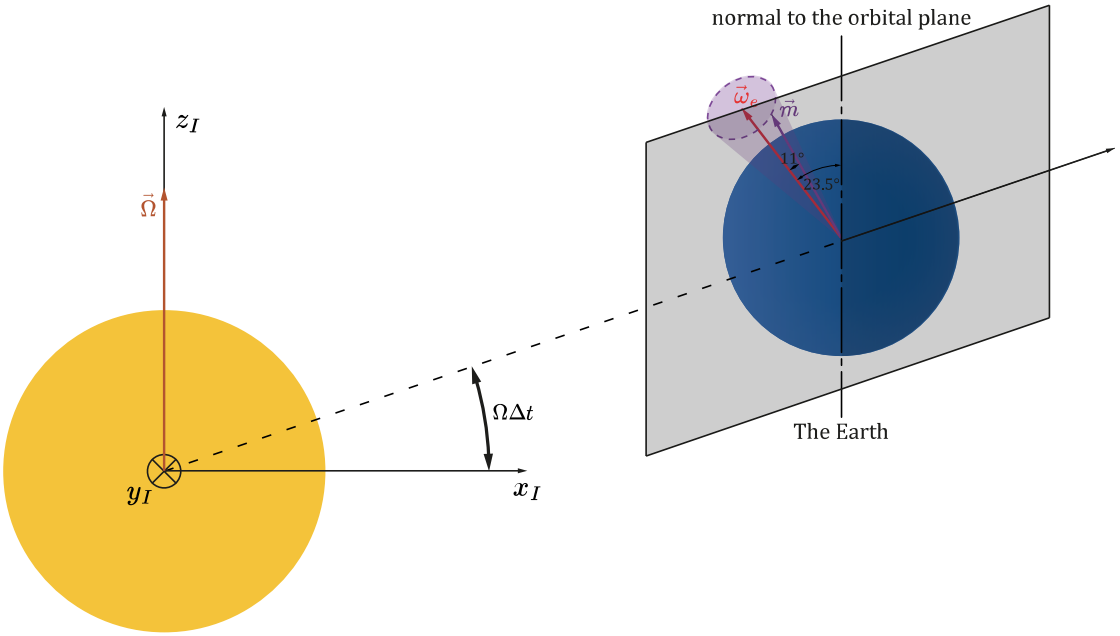


Figure 14: Earth magnetic moment motion

With the numerous frames that have been defined, it is important to define the transformation matrices that enable the transporting a vector from one frame to another. A good thing is that these transformations are all rotations, thus their corresponding matrices can be inverted by taking their transpose :

$$\begin{aligned} P_I^{ORB}(t) &:= \mathcal{R}_{\mathbf{e_z}}(\Omega t) \\ P_I^{ROT}(t) &:= \mathcal{R}_{\mathbf{e_y}}(23.5°)\mathcal{R}_{\mathbf{e_z}}(\omega t) \\ P_{ROT}^{MAG} &:= \mathcal{R}_{\mathbf{e_y}}(11.0°) \\ P_I^{MAG}(t) &:= P_I^{ROT}(t)P_{ROT}^{MAG} \end{aligned}$$

Where $\mathcal{R}_{\mathbf{u}}(\theta) \longleftrightarrow q_{\mathbf{u},\theta}$.

Radiation flux modelisation

The radiations flux that receives the satellites can be decomposed into three components:

- Q_p^s : Direct Sun radiation.
- Q_p^{ref} : Sun radiation reflected by Earth and its atmosphere.
- Q_p^{bb} : Earth Outgoing Longwave Radiation (OLR), emitted by Earth's black body.

The direct Sun radiation is computed using Lambert's cosine law, given the assumption that the Sun's radiation flux is constant throughout the orbit of the satellite, when it is not in Earth's shadow zone. The radiation received by a face P of the satellite is the following :

$$\boxed{Q_p^s = Q_0 \cos(\theta)} \tag{16}$$

Where $\vec{n}_P$ is the normal to the satellite's P face, $\vec{s}$ is the unitary Sun rays' direction vector, $\theta = -\langle \vec{s}, \vec{n}_P \rangle$, and Q_0 is the mean radiation flux emitted by the Sun and received near Earth's atmosphere.

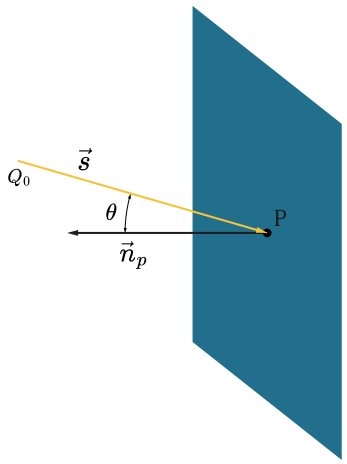


Figure 15: Direct Sun radiation

The reflected radiation is more complicated to compute, and a differential approach will thus be used. The first thing to note is that at first order, the Earth's atmosphere and surface absorbs about 70% of the radiation it receives, and only reflects a mean of 30% of it. The amount of radiation it reflects back into space is called albedo.

Albedo is not easily modelled as it depends on many parameters, such as the nature of the land's surface (Are there trees? Is their water? ...), and the meteorological state (Are there any clouds?).

By trying to fit to fig. 17, the function chosen to model the albedo is a flat function around the equator, whose parameters have been chosen to have a minimum of 0.14 at the equator, a maximum of 0.63 at the poles, and a mean of 0.3. Formally, it is written as:

$$a(\theta) := a_{min} + (a_{max} - a_{min}) \frac{e^{-\frac{1}{(\theta - \pi/2)^2}}}{e^{-4/\pi^2}} \tag{17}$$

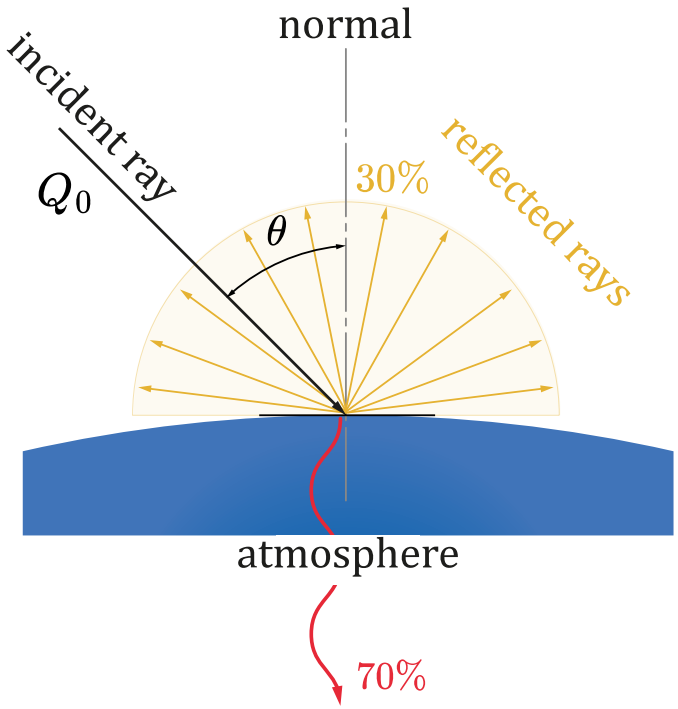


Figure 16: Absorbed and reflected energy

Where θ is the angle between a point on Earth's surface, and the rotation axis of Earth.

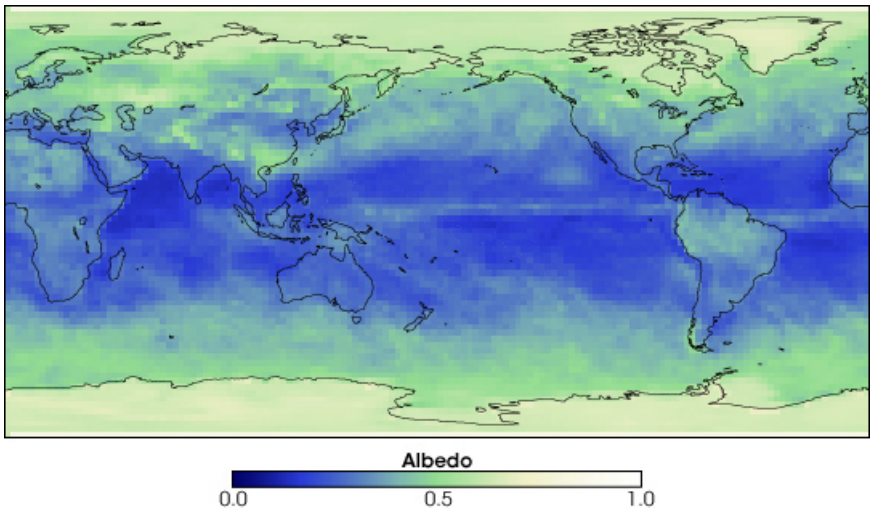


Figure 17: Albedo map processed with the NASA CERES mission's data [22]

Now that the albedo is modelled, the approach to compute the reflected solar flux

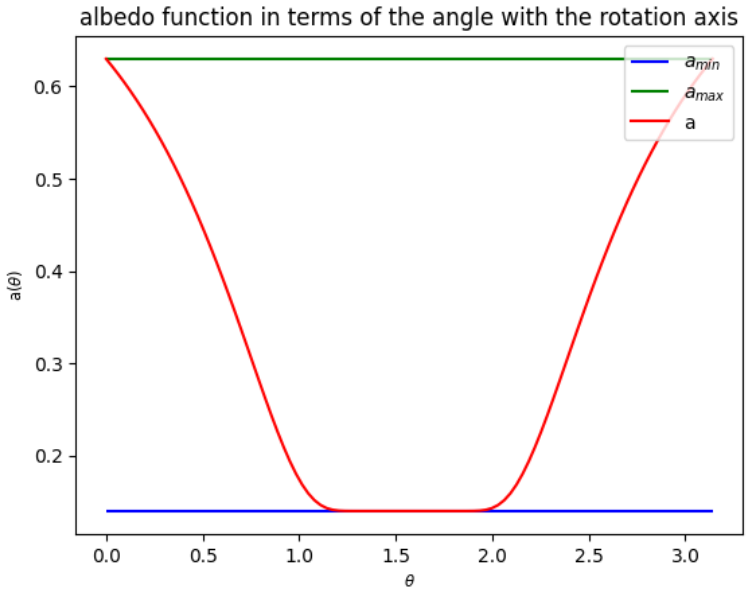


Figure 18: Albedo function

received on a face of the satellite will be the integration of the differential flux the face receives from a differential element of surface on Earth's surface. A common tool to compute these differential fluxes is the view factor.

The view factor between two surfaces represents the fraction of the radiation that exits the first surface and that is received by the other one.

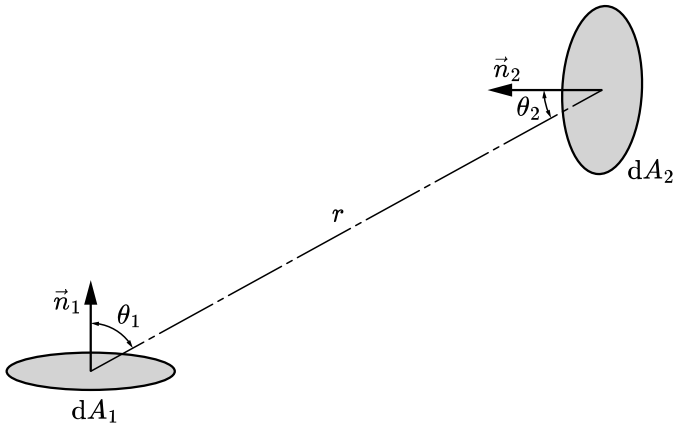


Figure 19: View factor

Given the two differential surfaces seen in fig. 19, the fraction of the radiation that exits surface 1 and that is absorbed by surface 2 is :

$$dF_{12} = \cos(\theta_1)\frac{d\omega_{12}}{\pi} \tag{18}$$

Where $d\omega_{12}$ is the solid angle subtended by surface 1 and viewed from surface 2:

$$d\omega_{12} := \frac{\cos(\theta_2)dA_2}{r^2}$$

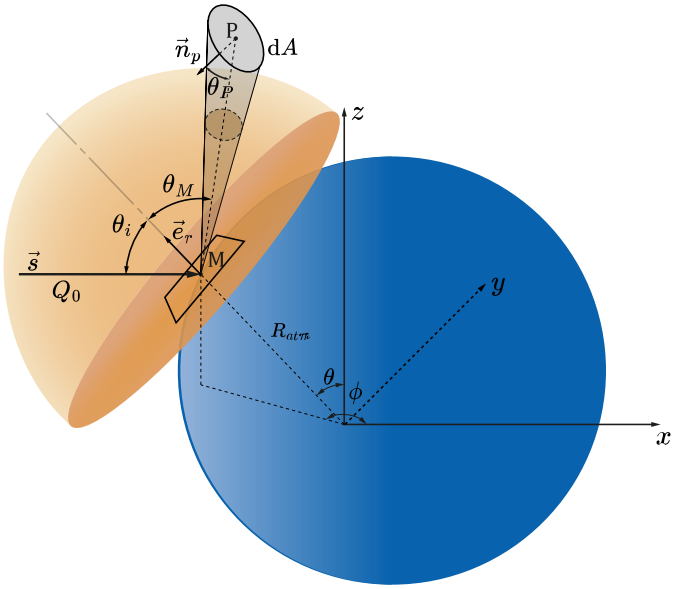


Figure 20: Reflected flux received by a satellite face

Taking the notations from fig. 20, and using the expression of the view factor between two differential plates, one has :

$$\begin{aligned}
E_i &= Q_0 \cos(\theta_i)\sin(\theta)R_{atm}^2 d\theta d\phi \\
E_r &= a(\theta,\phi)\ E_i \\
E_p &= E_r \underbrace{\cos(\theta_M)\frac{\cos(\theta_P)dA_P}{\pi R^2}}_{dF_{MP}} \\
dQ_p^{ref} &= \frac{E_p}{dA_p}
\end{aligned}$$

Where E_i is the incident energy absorbed by the elementary surface M, E_r is the energy reflected isotropically by this surface and E_p is the energy received by the satellite's face P. dQ_p^{ref} is finally the differential reflected flux received on the satellite's surface that is emitted by the differential surface M.

By integrating on Earth's surface, the total radiation flux that is reflected by Earth and its atmosphere is :

$$\boxed{Q_p^{ref} = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} a(\theta, \phi)\ Q_0 \cos(\theta_i) \cos(\theta_M) \cos(\theta_P) \sin(\theta) \frac{R_{atm}^2}{\pi R^2} d\theta d\phi} \tag{19}$$

The last radiation flux needed to calculate the total received radiation is the Earth OLR. This radiation is emitted as if the Earth was a black body, emitting a mean constant flux of $Q_{IR} = 239 W/m^2$ in all directions. Once again, the view factor is of use. This problem is simpler than the former as the emission doesn't depend on the orientation of each surface element, and is independent of the satellite's position. Hence, analytical expressions have been computed as in [23].

This solutions distinguishes 3 cases: when the differential plate is oriented towards the Earth, when the plane of the plate intersects the Earth, and when the plate doesn't face the Earth.

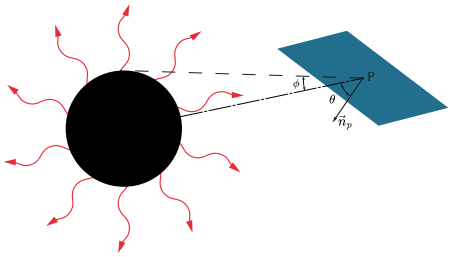


Figure 21: Black body Earth radiation

Given the angle θ between the plate's normal and the vector pointing from the center of the plate towards the center of the Earth, and ϕ the angle drawn in fig. 21, the value of the view factor is the following:

$$F = \begin{cases} 0 \text{ if } \theta > \pi/2 + \phi \\ \frac{\cos(\theta)}{H^2} \text{ if } \theta < \pi/2 - \phi \\ \frac{1}{2} - \frac{1}{\pi} \sin^{-1}\left[\frac{(H^2-1)^{1/2}}{H \sin(\theta)}\right] + (1/(\pi H^2)) \left(\cos(\theta) \cos^{-1}\left[-\left(H^2-1\right)^{1/2} \cot(\theta)\right] - (H^2-1)^{1/2}\left[1 - H^2 \cos^2\theta\right]\right) \text{ else} \end{cases} \tag{20}$$

Finally, one has:

$$\boxed{Q_p^{bb} := F Q_{IR}} \tag{21}$$

Sensor modelisation

The light sensors are photodiodes, Earth Radiative Sensor (ERS) sensors with Optical Sun Reflector (OSR) and ERS sensors with carbon nanotube (CNT) coating. They each measure on different spectrums of light. Using the different fluxes defined in eqs. (16), (19) and (21), the sensor's measurements are expressed in [21] as follows:

$$\begin{aligned} &\text{– Photodiodes :} && Q_{phot} = \left[Q_p^s + Q_p^{ref}\right]_{[400-1100]\text{nm}} \\ &\text{– ERS with OSR :} && Q_{OSR} = 6\%(Q_p^s + Q_p^{ref}) + 84\%Q_p^{bb} \\ &\text{– ERS with CNT coating :} && Q_{CNT} = Q_p^s + Q_p^{ref} + Q_p^{bb} \end{aligned} \tag{22}$$

The magnetometer's measurements are disturbed by a gaussian white noise η_B :

$$\hat{B} = B_{true} + \eta_B \tag{23}$$

The gyrometer's modelisation is more complicated as it accounts for the gyro bias drift. The continuous model is :

$$\hat{\omega(t)} = \omega(t) + \beta(t) + \eta_\omega(t)$$

$$\dot{\beta}(t) = \eta_\beta(t)$$

The discretised model, developed in [3] has been implemented as follows:

$$\hat{\omega}_{k+1} = \omega_{k+1} + \frac{1}{2}(\beta_{k+1} + \beta_k) + \left(\frac{\sigma_v^2}{\Delta t} + \frac{1}{12}\sigma_u^2 \Delta t\right)^{1/2} \mathbf{N}_{vk} \tag{24}$$

Where

$$\beta_{k+1} = \beta_k + \sigma_u \Delta t^{1/2} \mathbf{N}_{uk}$$

With $\mathbf{N}_{vk}$ and $\mathbf{N}_{uk}$ white noises with the identity matrix as their covariance matrix.

3.2.2 Data generation

Now that the modelisation of the system is done, after having implemented the necessary functions, the simulation outputs a dataset, with each row representing a given timestep. The variables of the dataset include the various sensor measurements, the true attitude of the satellite, the position of the satellite, as well as the true sun Line of Sight (LOS), Magnetic Field and Nadir directions.

3.2.3 Simulation exploitation

Quaternions and loss function

The quaternion representation has one major issue when dealing with a MLP model as both $q \in \mathbb{H}$ and $-q$ represent the same rotation[2]. Thus, using classic loss functions such as the Mean Squared Error (MSE) is not ideal. As such, training the model using the geodesic loss function expressed in [24] is more appropriate.

Given two sequences unitary quaternions $(q_k)_k, (\hat{q_k})_k$ such that $\forall k$, $\hat{q_k}$ is an estimate of q_k using the MLP ; the geodesic loss can be expressed as follows:

$$\mathcal{L}\left((\hat{q}_k)_k, (q_k)_k\right) := \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_G(\hat{q}_i, q_i) + \mathcal{L}_N(\hat{q}_i) \tag{25}$$

With $\forall q, p \in \mathbb{H}$

$$\mathcal{L}_G(q,p) := 1 - \langle q,p\rangle^2$$
$$\mathcal{L}_N(q) := \beta\left(1 - ||q||\right)^2$$

Where N is the number of samples of the sequences, β is an arbitrary positive weight and $\langle q,p\rangle := \sum_{i=1}^{4} q_i p_i$. $\mathcal{L}_G(\hat{q},q) = \frac{1-cos(\theta)}{2}$ is the true geodesic loss, where θ is the geodesic distance between the rotations represented by the two quaternions as in eq. (5). This term tends to make the algorithm make θ converge towards 0 $[2\pi]$. $\mathcal{L}_N(\hat{q},q)$ is a normalization term, to make the model have (near) normalized outputs.

MLP results

The algorithm was first tested on 1000 sequences of 60 seconds, uniformly distributed on the orbit. This amount of data was not sufficient, as the algorithm didn't converge properly.

80,000 training data points were then generated (only on the shadow zone this time). The model architecture was chosen by comparing 1200 different models with 3 hidden layers, and choosing the one that minimise the test error. Ultimately, a better 5 layer was found by guessing and checking.

- hidden layer sizes : [126, 96, 72, 65, 56]
- activation function : ReLU
- learning rate : 0.001
- batch size : 500

[2] A rotation is represented by a unique unitary quaternion and its opposite

The algorithm still had a hard time learning, providing a validation error mean of 16.94° and a standard deviation of 11.49°.

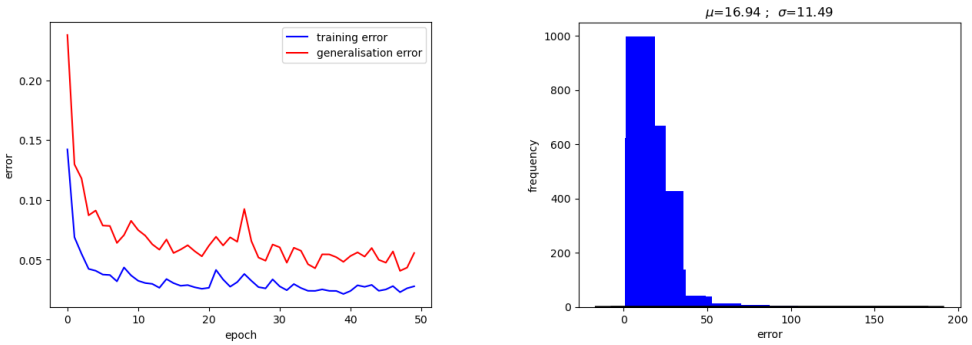


Figure 22: Training curve and error histogram

MLP-aided TRIAD

As seen in the previous section, the MLP fails to generalise the estimation of the attitude quaternion of the satellite. One way to simplify the task of the algorithm is to implement in into the TRIAD method. Rather than directly estimating the attitude quaternion, the MLP can focus on estimating for instance the nadir and the magnetic field directions. These two components can then be used to retrieve the attitude of the satellite using the TRIAD method.

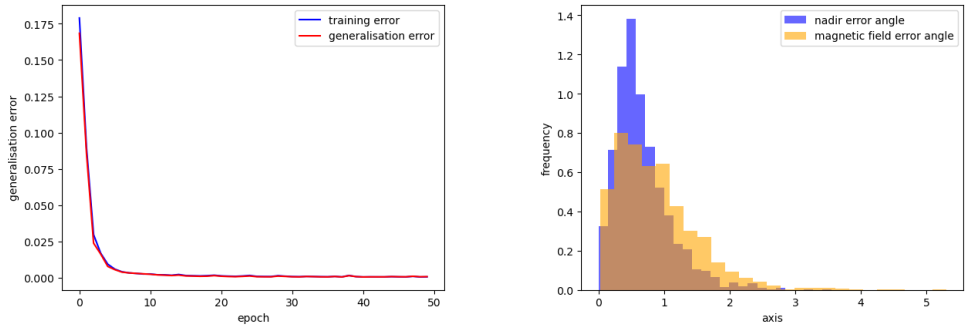


Figure 23: Training curve and error histogram for nadir/magnetic field estimation

Training the MLP using the MSE loss on the magnetic field and nadir directions in the body frame results in errors all below 5°.

The estimated errors are very low, and can thus yield to attitude estimates of better precision compared to the classic TRIAD method which had 6° of standard deviation around a mean of 8°.

	MEAN (deg)	STD (deg)
Nadir angle error	0.67	0.43
Magnetic angle error	0.87	0.62

Table 3: Nadir and magnetic direction angle estimation errors

The attitude geodesic angle error distribution can then be estimated using a Monte Carlo method. This simulation is done as follows:

Two unitary vectors $\mathbf{v}_{nadir}$ and $\mathbf{v}_{mag}$ are generated randomly. A random unitary attitude quaternion is also generated. The corresponding vectors are computed in the body frame of the satellite. These vectors are then perturbed by a small random rotation of an angle drawn in $\mathcal{N}(0.67, 0.43)$ and $\mathcal{N}(0.87, 0.62)$ respectively. These perturbed body frame vectors are simulate the MLP's output.

The TRIAD method can then be used to compute the estimated attitude of the satellite. This attitude can be compared to the true attitude of the satellite using the geodesic error of eq. (5).

This procedure has been iterated $N_{iter} = 100,000$ times.

MEAN (deg)	STD (deg)
1.29	2.24

Table 4: Monte Carlo error

The results, seen in table 4 are way better than the regular TRIAD method which had a 6° standard deviation around an 8° mean.

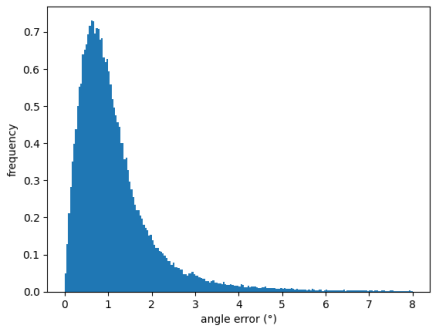


Figure 24: Monte carlo error distribution

Conclusions & Perspectives

The urgent need to comprehend and address the complex phenomenon of climate change has driven advancements in space technology, enabling more actors to send their own small satellites in orbit. Equipped with the right sensors, a constellation of cubesats could monitor and measure with high definition the ERB, one of the important variables of climate change.

The present report focused on the specific case of Inspire-sat 7, a cubesat aimed at monitoring the ERB not equipped with any active pointing system. The attitude determination of the satellite is critical for having accurate measurements. Traditional Attitude Determination (AD) algorithms have been examined in this context, revealing their strengths and limitations. However, this study went beyond the conventional methods and delved into the realm of Deep Learning (DL) techniques, particularly the Multilayer Perceptron algorithm for attitude retrieval.

First through a simplified laboratory test, this report showed the feasibility of the technique in a controlled environment, by retrieving the attitude of the satellite with an angular error of 4.07° at 1σ. Because of the small ammount of data, a simulation was implemented and enabled to generate a lot more datapoints. This simulation showed the limitations of the MLP to directly estimate the attitude quaternion of the satellite. It has then been demonstrated that including the intermediary step of estimating the nadir and magnetic field direction, paired with the TRIAD method could solve this problem. The MLP indeed retrieved the two vectors with high precision (error angles means close to 1° and with low standard deviation). Inputting these values in a Monte Carlo simulation showed that in the shadow zone, the MLP outperformed the classic TRIAD method, with an error of 2.24° standard deviation around a 1.29° mean, against the TRIAD's 6° standard deviation around 8° mean.

The perspective of this study are numerous. This work has first shown the potential of ML-based techniques for attitude estimation. With the development of significantly more complete test benches, one could imagine training a ML algorithm specifically for a satellite on ground before using it in orbit. The implemented simulation could also be used to train and test other Machine Learning algorithms, say recurrent ones such as Long Short Term Memory (LSTM), or Gated Recurrent Units (GRU).

With the very short amount of time needed to conduct a cubesat mission and their potential to measure climate change with high precision, cubesats have become the perfect tools to monitor our planet from Space.

A Quaternions [3]

A.1 Definition

Quaternions are a mathematical concept invented by Sir William Rowan Hamilton in 1874. In the same way complex numbers are used to represent rotations in the complex plane, quaternions can be used to represent rotations in 3d space. They are especially useful when dealing with attitude representation problems, since they don't inherit from the singularity issue euler angles have.

Figure 25: The invention of quaternions by Sir William R. Hamilton – image taken from https://www.quora.com/What-made-the-quaternions-a-great-discovery

Quaternions are a set of hyper-complex numbers (written $\mathbb{H}$), generated by the basis $\{i, j, k, 1\}$ and following Hamilton's non commutative product rule :

$$\begin{cases} i^2 = j^2 = k^2 = ijk = -1 \\ ij = -ji = k \\ jk = -kj = i \\ ki = -ik = j \end{cases}$$

A.2 Properties and operations

A quaternion $p \in \mathbb{H}$ can thus be written as follows : $q = q_1 i + q_2 j + q_3 k + q_4$

Where $q_1 i + q_2 j + q_3 k$ is the imaginary part of the quaternion and q_4 is its real part.

Two main quaternion notations exist. One where the real element is the first element, and one where it is the last. The latter will be used, as in [3].

The different operations used with quaternions can now be listed :

Let $q, p \in \mathbb{H}$

- Summation
  $q + p = (q_1 + p_1)i + (q_2 + p_2)j + (q_3 + p_3)k + (q_4 + p_4)$
- Product [3]
  The quaternionic product is done using the distributivity of the multiplication combined with Hamilton's product rule :

$$\begin{aligned} q \otimes p =& (q_1 i + q_2 j + q_3 k + q_4)(p_1 i + p_2 j + p_3 k + p_4) \\ =& q_1 p_1 i^2 + q_1 p_2 ij + q_1 p_3 ik + q_1 p_4 i \\ &+ q_2 p_1 ji + q_2 p_2 j^2 + q_2 p_3 jk + q_2 p_4 j \\ &+ q_3 p_1 ki + q_3 p_2 kj + q_3 p_3 k^2 + q_3 p_4 k \\ &+ q_4 p_1 i + q_4 p_2 j + q_4 p_3 k + q_4 p_4 \end{aligned}$$

$$\begin{aligned} q \otimes p =& (q_1 p_4 + q_2 p_3 - q_3 p_2 + q_4 p_1)\, i \\ &+ (-q_1 p_3 + q_2 p_4 + q_3 p_1 + q_4 p_2)\, j \\ &+ (q_1 p_2 - q_2 p_1 + q_3 p_4 + q_4 p_3)\, k \\ &+ (-q_1 p_1 - q_2 p_2 - q_3 p_3 + q_4 p_4) \end{aligned} \tag{26}$$

- Conjugation
  $\bar{q} := -q_1 i - q_2 j - q_3 k + q_4$

A.3 Matrix representation

Quaternions can also be represented by matrices. This matricial paradigm greatly simplifies the previous operations by using linear algebra.

Using this notation, it will be writen from now on :

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \in \mathbb{H}$$

[3] $\otimes$ is the quaternionic product

Looking at the extended formula written in eq. (26), and using the matrix notation, one can be simultaneously write:

$$p \otimes q = \underbrace{\begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix}}_{\mathcal{L}_q} p \qquad p \otimes q = \underbrace{\begin{bmatrix} p_4 & p_3 & -p_2 & p_1 \\ -p_3 & p_4 & p_1 & p_2 \\ p_2 & -p_1 & p_4 & p_3 \\ -p_1 & -p_2 & -p_3 & p_4 \end{bmatrix}}_{\mathcal{R}_p} q \quad (27)$$

A.4 Spatial rotations

A.4.1 Transformation matrix

Rotations in space can be described by unitary quaternions.

Let us choose an arbitrary unitary axis $\mathbf{u} \in \mathbb{R}^3$, an angle $\theta \in [0, 2\pi]$ and two frames a and b such that frame b is the frame resulting from the rotation of frame a of an angle of θ along $\mathbf{u}$. The transformation from frame a to frame b can be described using the following quaternion.

$$q_{\mathbf{u},\theta} := \begin{bmatrix} \sin(\frac{\theta}{2})\mathbf{u} \\ \cos(\frac{\theta}{2}) \end{bmatrix} \text{ [4]}$$

Given any vector $\mathbf{v}|_b \in \mathbb{R}^3$ in frame b, its equivalent vector $\mathbf{v}|_a$ expressed in frame a can be computed using quaternions.

$$\begin{bmatrix} \mathbf{v}|_a \\ 0 \end{bmatrix} = q_{\mathbf{u},\theta} \otimes \begin{bmatrix} \mathbf{v}|_b \\ 0 \end{bmatrix} \otimes \bar{q}_{\mathbf{u},\theta}$$

Which, using eq. (27), can be re-written as :

$$\begin{bmatrix} \mathbf{v}|_a \\ 0 \end{bmatrix} = \mathcal{L}_{q_{\mathbf{u},\theta}} \mathcal{R}_{\bar{q}_{\mathbf{u},\theta}} \begin{bmatrix} \mathbf{v}|_b \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{v}|_a \\ 0 \end{bmatrix} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_3q_4) & 2(q_1q_3+q_2q_4) & 0 \\ 2(q_1q_2+q_3q_4) & 1-2(q_1^2q_3^2) & 2(q_2q_3-q_1q_4) & 0 \\ 2(q_1q_3-q_2q_4) & 2(q_1q_4+q_2q_3) & 1-2(q_1^2+q_2^2) & 0 \\ 0 & 0 & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}|_b \\ 0 \end{bmatrix} \quad (28)$$

[4] Note that given the unitarity of $\mathbf{u}$, $q_{\mathbf{u},\theta}$ is also unitary

Taking only the first 3 coordinates of equation eq. (28), one has: $\mathbf{v}|_a = \mathcal{P}_a^b \mathbf{v}|_b$, with

$$\boxed{\mathcal{P}_a^b := \left[\mathcal{L}_{q_{\mathbf{u},\theta}} \mathcal{R}_{\bar{q}_{\mathbf{u},\theta}}\right]_{[1:3,1:3]} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_3q_4) & 2(q_1q_3+q_2q_4) \\ 2(q_1q_2+q_3q_4) & 1-2(q_1^2q_3^2) & 2(q_2q_3-q_1q_4) \\ 2(q_1q_3-q_2q_4) & 2(q_1q_4+q_2q_3) & 1-2(q_1^2+q_2^2) \end{bmatrix}} \tag{29}$$

From this equation, the attitude matrix of $q_{\mathbf{u},\theta}$ can be defined as $A(q_{\mathbf{u},\theta}) := \mathcal{P}_a^b$[5]

This attitude matrix naturally respects the following property:

$$A(q \otimes p) = A(q)A(p) \tag{30}$$

for q and p some arbitrary rotation quaternions.

A.4.2 From transformation matrix to quaternion

It was just seen that it was possible to compute rotation matrices given some rotation quaternion. It can be shown that it is also possible to compute a quaternion from the rotation matrix.

Shepperd introduced in 1978 [25] a method of extracting the unit rotation quaternion associated to a rotation matrix.

Given a rotation matrix $R = (R_{ij})_{i,j\in[1,3]} \in \mathbb{R}^{3\times3}$ which can be written as:

$$R = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_3q_4) & 2(q_1q_3+q_2q_4) \\ 2(q_1q_2+q_3q_4) & 1-2(q_1^2q_3^2) & 2(q_2q_3-q_1q_4) \\ 2(q_1q_3-q_2q_4) & 2(q_1q_4+q_2q_3) & 1-2(q_1^2+q_2^2) \end{bmatrix}$$

[5] $\mathcal{P}_a^b$ being a rotation matrix of SO(3), its inverse is its transpose

One can find that :

$$
\begin{aligned}
q_1^2 &= \frac{1}{4}(1 + R_{11} - R_{22} - R_{33}) \\
q_2^2 &= \frac{1}{4}(1 - R_{11} + R_{22} - R_{33}) \\
q_3^2 &= \frac{1}{4}(1 - R_{11} - R_{22} + R_{33}) \\
q_4^2 &= \frac{1}{4}(1 + R_{11} + R_{22} + R_{33})
\end{aligned}
$$

Given that the quaternion is unitary, at least one of its components is non null. One of the following equivalent formulae can thus be used, given it is properly defined, to compute our quaternion :

$$
\begin{aligned}
q^{(1)} &= \frac{1}{4q_1} \begin{bmatrix} 4q_1^2 \\ R_{12} + R_{21} \\ R_{13} + R_{31} \\ R_{32} - R_{23} \end{bmatrix} \\
q^{(2)} &= \frac{1}{4q_2} \begin{bmatrix} R_{12} + R_{21} \\ 4q_2^2 \\ R_{23} + R_{32} \\ R_{13} - R_{31} \end{bmatrix} \\
q^{(3)} &= \frac{1}{4q_3} \begin{bmatrix} R_{13} + R_{31} \\ R_{23} + R_{32} \\ 4q_3^2 \\ R_{21} - R_{12} \end{bmatrix} \\
q^{(4)} &= \frac{1}{4q_4} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \\ 4q_4^2 \end{bmatrix}
\end{aligned}
$$

Although these calculations are mathematically correct, dividing by the square root of a small number can make poorly numerically stable calculations. Shepperd provided a way to choose the most stable equation to compute the quaternion.

Indeed the most stable computation is $q^{(i)}$, where:

$$i := \text{argmax} \begin{bmatrix} \text{Tr}(R) \\ R_{11} \\ R_{22} \\ R_{33} \end{bmatrix}$$

This algorithm still results in singularities when the rotation axis is one of the main frame's axis. Mike Day has adapted an improved version of this algorithm in [26], which eliminates these issues.

A.5 Rotational dynamics

One can now compute the derivative of a rotation quaternion following the steps developed in [27], and adapted to our representation.

Given an arbitrarily small time step Δt, $q(t+\Delta t)$ can be written by using eq. (30), as $\Delta q \otimes q(t)$, where Δq represents an infinitesimal rotation of q.

By defining the rotation rate ω in the body frame, subtended by the imaginary basis and associated to Δq, both the small angle of rotation can be expressed as $\Delta\theta = ||w||\Delta t$ and the unitary axis of rotation as $\frac{\omega}{||\omega||}$. Thus,

$$\Delta q = \sin\left(\frac{||w||\Delta t}{2}\right)\frac{\omega}{||\omega||} + \cos\left(\frac{||w||\Delta t}{2}\right) \tag{31}$$

From this the rate of change of q can be computed:

$$\begin{aligned}
\frac{q(t+\Delta t)-q(t)}{\Delta t} &= \frac{\Delta q \otimes q(t) - q(t)}{\Delta t} \\
&= \frac{\left[\sin\left(\frac{||w||\Delta t}{2}\right)\frac{\omega}{||\omega||} + \cos\left(\frac{||w||\Delta t}{2}\right)\right] \otimes q(t) - q(t)}{\Delta t} \\
&= \frac{\left[\sin\left(\frac{||w||\Delta t}{2}\right)\frac{\omega}{||\omega||} + \cos\left(\frac{||w||\Delta t}{2}\right) - 1\right] \otimes q(t)}{\Delta t} \\
&= \frac{\left[\frac{||\omega||\Delta t}{2}\frac{\omega}{||\omega||} + 1 - 1 + o(\Delta t)\right] \otimes q(t)}{\Delta t} \\
\frac{q(t+\Delta t)-q(t)}{\Delta t} &= \frac{1}{2}\omega \otimes q + o(1) \otimes q(t)
\end{aligned}$$

By passing to the limit as $\Delta t \to 0$, one finally has:

$$\boxed{\dot{q}(t) = \frac{1}{2}\omega(t) \otimes q(t)} \tag{32}$$

The matricial form of this equation, by defining $\Omega := \mathcal{L}_{w(t)}$ is as follows:

$$\dot{q}(t) = \frac{1}{2}\Omega(t)q(t)$$

Which can be integrated as follows, for t sufficiently close to any $t_k \in \mathbb{R}^+$, allowing us to consider Ω constant on $[t_k, t]$:

$$q(t) = e^{(t-t_k)\Omega(t_k)}q(t_k) \tag{33}$$

Discretising equation eq. (33), by using $t_k := k\Delta t$ and $t = t_{k+1}$ and only keeping the first order terms gives us the expression of the next q iterate at each time-step :

$$\boxed{q_{k+1} = q_k + \frac{1}{2}dt\ \Omega_k q_k} \tag{34}$$

B Kalman Filters [3]

B.1 Definition

The traditional Kalman Filter (KF) is an iterative algorithm designed to estimate the state of a linear system. It uses an iterative approach, first predicting the next step's system state using the currently known informations and the dynamics. It then adjusts this prediction using the current sensor measurements of different quantities.

The KF is used on systems of the following form:

$$\begin{cases} x_{k+1} = \Phi_k x_k + G_k u_k + \omega_k \\ y_k = H_k x_k + \nu_k \end{cases} \tag{35}$$

Where $x_k \in \mathbb{R}^n$ is the state of the system, $u_k \in \mathbb{R}^p$ is the command of the user on the system, $y_k \in \mathbb{R}^q$ is the sensor's measurements term. In addition, $\Phi_k \in M_{n,n}(\mathbb{R})$, $G_k \in M_{n,p}(\mathbb{R})$ and $H_k \in M_{q,n}(\mathbb{R})$ are the different linear maps defining the dynamics and observability of the system. Finally, ω_k, ν_k are centered noise terms such that

$$E[\nu_k \nu_k^T] := R_k \succ 0 \quad , \quad E[w_k w_k^T] := Qd_k \succ 0$$

In a Kalman Filter, it is supposed that the system's different linear maps are known.

B.2 KF algorithm

At each iteration, the filter starts by estimating the system state, given the known dynamics and the previous state estimations:

$$\begin{aligned} \hat{x}_{k+1}^- &= \Phi_k x_k^+ + G_k u_k \\ y_{k+1}^- &= H_{k+1} \hat{x}_{k+1}^- \end{aligned}$$

Where the $-$ superscript indicates the prediction estimate and the $+$ superscript indicates the corrected estimate.

This first estimate is then adjusted given the error measured between the measurement estimation and the true sensor's measurements:

$$\hat{x}_{k+1}^+ = \hat{x}_{k+1}^- + L_{k+1}(y_{k+1} - y_{k+1}^-)$$

Where L_k is the chosen gain matrix. The Kalman filter's objective is to find the "best" gain matrix.

By defining:

$$\begin{aligned}\delta x_k &:= x_k - \hat{x}_k \\ \delta y_k &:= y_k - \hat{y}_k \\ P_k &:= E\left[(\delta x_k - E[\delta x_k])(\delta x_k - E[\delta x_k])^T\right]\end{aligned}$$

Computations show that:

$$\begin{aligned}\delta x_{k+1}^- &= \Phi_k \delta x_k^+ + w_k \\ \delta y_k^- &= H_k \delta x_k^- + \nu_k \\ \delta x_k^+ &= (I - L_k H_k)\delta x_k^- - L_k \nu_k \\ P_{k+1}^- &= \Phi_k P_k^+ \Phi_k^T + Q d_k \\ P_k^+ &= (I - L_k H_k) P_k^- (I - L_k H_k)^T + L_k P_k L_k^T\end{aligned}$$

The goal of the Kalman filter is to find the gain matrix L_k that minimizes the variance in the estimation result, with the posterior estimator written as:

$$\hat{x}_k^+ := \hat{x}_k^- + L_k \delta y_k^-$$

Thus, the following optimisation problem is derived:

$$\underset{L_k}{\operatorname{argmin}}\ J(L_k) = Tr(P_k^+) \tag{36}$$

This cost function can be then developped as:

$$\begin{aligned}J(L_k) &= Tr(P_k^- - P_k^- H_k^T L_k^T - L_k H_k P_k^- + L_k(H_k P_k^- H_k^T + R_k)L_k^T) \\ &= Tr(P_k^-) - 2Tr(L_k H_k P_k^-) + Tr(L_k(H_k P_k^- H_k^T))\end{aligned}$$

The Necessary Optimality Condition finally gives:

$$\begin{aligned}&\nabla_{L_k} J(L_k) = 0 \iff L_k = (H_k P_k^-)^T (H_k P_k^- H_k^T + R_k)^{-1} \\ &\nabla_{L_k}^2 J(L_k) \succ 0\end{aligned}$$

This leads to the Kalman gain:

$$\boxed{K_k := (H_k P_k^-)^T (H_k P_k^- H_k^T + R_k)^{-1}} \tag{37}$$

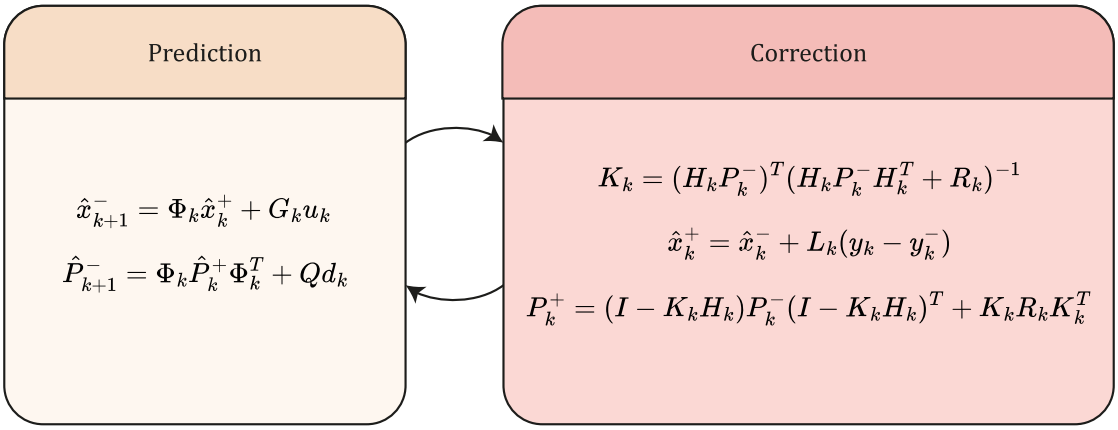


Figure 26: Kalman Filter

References

[1] Michael A. Nielsen, Jan 1970.

[2] Adrien Finance et al. A new method based on a multilayer perceptron network to determine in-orbit satellite attitude for spacecrafts without active adcs like uvsq-sat. *Remote Sensing*, 13(6), 2021.

[3] F. LANDIS MARKLEY. *Fundamentals of spacecraft attitude determination and control*. SPRINGER, 2016.

[4] HAROLD D. BLACK. A passive system for determining the attitude of a satellite. *AIAA Journal*, 2(7):1350–1351, 1964.

[5] J. L. Farrell, J. C. Stuelpnagel, R. H. Wessner, J. R. Velman, and J. E. Brook. A least squares estimate of satellite attitude (grace wahba). *SIAM Review*, 8(3), 1966.

[6] Landis Markley. Attitude determination from vector observations: A fast optimal matrix algorithm. *Journal of the Astronautical Sciences*, 41, 07 1993.

[7] Paul B. Davenport. A vector approach to the algebra of rotations with applications. *NASA Technical Note D-4696*, 1968.

[8] Yuhei KIKUYA, Yohei IWASAKI, Yoichi Yatsu, and Saburo MATUNAGA. Attitude determination algorithm using earth sensor images and image recognition. *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES*, 64:82–90, 01 2021.

[9] John A. Christian. A tutorial on horizon-based optical navigation and attitude determination with space imaging systems. *IEEE Access*, 9:19819–19853, 2021.

[10] Xiucong Sun, Xuejian Mao, and Pei Chen. High-precision attitude determination using spaceborne gravity gradiometer and gyroscope. *Acta Astronautica*, 200:213–225, 2022.

[11] Habib Ghanbarpourasl. Spacecraft attitude determination using electrostatically suspended gyroscope with rotating rings. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 235:095441002199615, 03 2021.

[12] Sung-Hoon Mok, Soo Yung Byeon, Hyochoong Bang, and Yoonhyuk Choi. Performance comparison of gyro-based and gyroless attitude estimation for cubesats. *International Journal of Control, Automation and Systems*, 18(5):1150–1160, 2020.

[13] Leandro Baroni. Attitude determination by unscented kalman filter and solar panels as sun sensor. *The European Physical Journal Special Topics*, 229(8):1501–1506, 2020.

[14] Kesaobaka Mmopelwa, Teddy Tumisang Ramodimo, Oduetse Matsebe, and Bokamoso Basutli. Attitude determination system for a cubesat experiencing eclipse. *Preprints*, May 2023.

[15] Alex Erlank. Arcminute attitude estimation for cubesats with a novel nano star tracker. volume 47, pages 9679–9684, 08 2014.

[16] Patrik Reizinger and Vajda Ferenc. Cubesat attitude determination with decomposed kalman filters. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 45, 02 2023.

[17] Zhenbing Qiu and Lei Guo. Improved cubature kalman filter for spacecraft attitude estimation. *IEEE Transactions on Instrumentation and Measurement*, PP:1–1, 11 2020.

[18] M. N. Filipski and R. Varatharajoo. Evaluation of a spacecraft attitude and rate estimation algorithm. *Aircraft Engineering and Aerospace Technology*, 82(3):184–193, Jan 2010.

[19] Hasan Kinatas and Chingiz Hajiyev. Quest aided ekf for attitude and rate estimation using modified rodrigues parameters. *WSEAS Transactions on Systems and Control*, 17:250–261, 06 2022.

[20] Itzhack Bar-Itzhack and Richard Harman. Optimized triad algorithm for attitude determination. *Flight Mechanics/Estimation Theory Symposium*, 1996.

[21] Adrien Finance. *Observing and quantifying Earth's radiation budget from UVSQ-SAT satellite*. PhD thesis, 2023. Thèse de doctorat dirigée par Meftah, Mustapha Astronomie et Astrophysique université Paris-Saclay 2023.

[22] NASA image by Jesse Allen. Earth's albedo in decline: Image of the day. https://earthobservatory.nasa.gov/images/5484/earths-albedo-in-decline, 2005. [Online; accessed July 24, 2023].

[23] E. G. HAUPTMANN. Angle factors between a small flat plate and a diffusely radiating sphere. *AIAA Journal*, 6(5):938–939, 1968.

[24] Julien Langlois et al. 3d orientation estimation of industrial parts from 2d images using neural networks. pages 409–416, 01 2018.

[25] S. W. Shepperd. Quaternion from rotation matrix. *Journal of Guidance and Control, Vol. 1, No.3*, pages 223–224, 1978.

[26] Mike Day. Converting a rotation matrix to a quaternion. https://d3cw3dd2w32x2b.cloudfront.net/wp-content/uploads/2015/01/matrix-to-quat.pdf. [Online; accessed July 21, 2023].

[27] Yan-Bin Jia. Quaternions. https://faculty.sites.iastate.edu/jia/files/inline-files/quaternion.pdf, Dec 2022. [Online; accessed July 21, 2023].