# Design, Implementation and Validation of an Attitude Determination Subsystem for Nanosatellites 

Mathieu Naslin<br>Embry-Riddle Aeronautical University - Daytona Beach

Follow this and additional works at: https://commons.erau.edu/db-theses
Part of the Aerospace Engineering Commons

## Scholarly Commons Citation

Naslin, Mathieu, "Design, Implementation and Validation of an Attitude Determination Subsystem for Nanosatellites" (2009). Theses - Daytona Beach. 156.
https://commons.erau.edu/db-theses/156

This thesis is brought to you for free and open access by Embry-Riddle Aeronautical University - Daytona Beach at ERAU Scholarly Commons. It has been accepted for inclusion in the Theses - Daytona Beach collection by an authorized administrator of ERAU Scholarly Commons. For more information, please contact commons@erau.edu.

# Embry Riddle Aeronautical University Aerospace Engineering Department 

Master of Science Thesis

# Design, Implementation and Validation of an Attitude Determination Subsystem for Nanosatellites 

## INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI Microform EP31994
Copyright 2011 by ProQuest LLC
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346

Ann Arbor, Ml 48106-1346

Design, Implementation and Validation of an Attitude Determination Subsystem for Nanosatellites
by

Mathieu Naslin

This thesis was prepared under the direction of the candidate's thesis committee chairman, Dr. Bogdan Udrea, Department of Aerospace Engineering, and has been approved by the members of his thesis committee. It was submitted to the Aerospace Engineering Department and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE:


Dr. Bogdan Udrea, Chairman


ABSTRACT<br>Author: Mathieu Naslin<br>Title: Design, Implementation and Validation of an Attitude Determination Subsystem for Nanosatellites<br>Institution: Embry Riddle Aeronautical University<br>Degree: Master of Science in Aerospace Engineering<br>Year: 2009

The purpose of this study is to design and analyze the accuracy of an attitude determination subsystem for a satellite of the CubeSat class by using low cost sensors. CubeSats are nanosatellites that complies a certain amount of layout criterions described by the California Polytechnic State University. A 3-axis attitude determination platform has been designed with emphasis on the use of low cost, off the shelf sensors. This platform features a sun sensor, a magnetometer and an earth sensor. The principles of observation and the description of the acquisition method are explained. The interfacing of the software package and real hardware is emphasized so as to obtain a practical platform for the nano-satellite's Guidance and Control (G\&C) tests. Computation of the attitude has been tested using the Three Axis Attitude Determination scheme (TRIAD). A test bench has been designed to be able to perform accurate rotation measurements. Easily feasible test procedures have been used to test the precision of each component of the acquisition and computing scheme. Tests have shown the relevance of the output of each sensing items. Results shown that a lens correction algorithm is needed to have a better accuracy on data computed from the camera used. The current design show that the sun sensor is accurate within 8 degree half cone and the earth sensor is accurate within 1 degree half cone. First attitude determination tests computed with the TRIAD algorithm showed that the overall accuracy of the computation scheme is within 5 degree half cone. This study has been especially focused on providing the general platform for the algorithm. Later studies will be necessary to make the subsystem more robust and accurate enough to be used in a real mission. The different ways to improve the system and its accuracy are discussed both for specific items and the entire sensing module.

## Content

1 INTRODUCTION .....  8
11 Purpose of this thesis ..... 8
12 Mission description ..... 9
13 MISSION ORIENTATION REQUIREMENTS AND ASSUMPTIONS ..... 11
2 ATTITUDE DETERMINATION ALGORITHMS ..... 14
21 INTRODUCTION TO ATTITUDE DETERMINATION ..... 14
22 TRIAD ALGORITHM ..... 17
221 Code structure ..... 19
23 QUEST ALGORITHM ..... 19
231 Computation Process ..... 20
3 VECTOR DETERMINATION AND SENSORS DESIGN ..... 23
31 SUN VECTOR DETERMINATION ..... 23
311 Sun vector in the Inertial Geocentric Frame ..... 23
312 Measured sun vector Sun sensor design ..... 26
3121 Hardware ..... 26
31211 Sensor ..... 26
31212 Lens design ..... 28
3122 Software ..... 32
32 MAGNETIC FIELD VECTOR DETERMINATION ..... 36
321 The reference magnetic field - World Magnetic Model ..... 36
3211 Overview ..... 36
32111 The IGRF WMM2005 ..... 38
3212 Magnetic field sensor magnetometer ..... 41
32121 The MemSense nIMU ..... 42
32122 Interfacing ..... 44
33 EARTH SENSOR DESIGN ..... 46
331 Introduction ..... 46
332 Computation scheme ..... 48
4 TEST PROCEDURE AND RESULTS. ..... 57
41 ACCURACY OF THE DESIGNED SUN SENSOR ..... 58
411 Test procedure ..... 58
412 Sun sensor test 1 Fixed dot test ..... 58
413 Sun sensor test 2 Homogeneous displacement on two directions ..... 60
42 ACCURACY OF THE MAGNETOMETER ..... 62
43 ACCURACY OF THE EARTH SENSOR ..... 64
431 Pitch accuracy of the Horizon sensor ..... 65
432 Roll accuracy of the Horizon sensor ..... 66
44 TRIAD ATTITUDE DETERMINATION ALGORITHM TEST ..... 67
5 CONCLUSIONS ..... 69
51 SOURCES OF ERROR \& ACCURACY OF EACH SENSOR ..... 69
52 ACCURACY OF THE ENTIRE ALGORITHM ..... 70
53 OVERALL CONCLUSION ..... 71
6 REFERENCE DOCUMENTS ..... 72
7 APPENDIX ..... 73

## List of figures

Figure 1: The Dip TE design ..... 9
Figure 2: Detailed design of the Dipte platform ..... 10
Figure 3: Artist view the the Dipte in orbit ..... 11
Figure 4: Current attitude determination algorithm pattern ..... 16
Figure 5: The TRIAD general setup ..... 17
Figure 6: Sketch of vectors for the Quest algorithm where the b frame represents the bF frame. ..... 20
Figure 7: Geometry of the earth-centered inertial frame and the Vsuni vector, [rD3] ..... 24
Figure 8: The Optina Imaging MT9M019 ..... 26
FIGURE 9: Picture acquisition process ..... 28
Figure 10: Different lens aperture shape ..... 29
Figure 11: Pinhole aperture geometry ..... 30
Figure 12: Geometric optics associated with the lens design ..... 31
Figure 13: Frame for the computation seen through the CCD sensor. Xccd and Ycdd define the CCD screen ..... 32
Figure 14: SUN frame description ..... 33
Figure 15: Graphic optics for sun vector determination ..... 35
Figure 16: Simple model of the Earth as a magnetic dipole for 2010 ..... 37
FIGURE 17: LOCATIONS OF THE NORTH DIP POLE (RED) AND THE GEOMAGNETIC NORTH POLE (blue) fOR THE YEARS 1900-2010, IGRF MODEL, source: British Geological Survey ..... 38
Figure 18: The CHAMP satellite, picture courtesy of GFC ..... 39
Figure 19: Sketch of a 3-axis magnetometer. ..... 42
Figure 20: Memsense nimu Functional Block diagram ..... 43
FIGURE 21: nIMU ACQUISITION CHAIN ..... 44
Figure 22: SAmple structure of data sent by the nimu ..... 44
FIGURE 23: HORIZON SENSORS PROVIDE A REFERENCE VECTOR USING THE DETECTED HORIZON LINE, SOURCE [RD3] ..... 46
Figure 24: Explanation of the nadir vector ..... 48
Figure 25: The earth sensor's field of view cone on a practical application ..... 49
FIGURE 26: HORIZON SENSOR OUTPUT PICTURE (SPACE SET TO BE WHITE FOR CLARITY) ..... 50
Figure 27: Angular radius geometry, AdAPTED From [RD15] ..... 51
Figure 28: Geometry of the roll angle $\eta$ ..... 52
Figure 29: Geometry of the nadir angle equation ..... 53
Figure 30: Configuration of negative pitch and tilted to the left compared to Figure 29 ..... 54
Figure 31: Configuration of positive pitch. ..... 56
Figure 32: SUN SENSOR SPATIAL ACCURACY TEST PROCEDURE ..... 58
FIGURE 33: FIXED DOT TEST OF THE SUN SENSOR WITH LASER POINTER, FOR A RANDOMLY CHOSEN POINT, SHOWN WITH CIRCLE FITTING DATA. ..... 59
Figure 34: nIMU reference frame ..... 62
Figure 35: Earth sensor's test bench ..... 64
FIGURE 36: Pure pitch test procedure for the horizon sensor. ..... 65
Figure 37: Pure roll test for the horizon sensor ..... 66
Figure 38: The Chain of sources of error. ..... 69
FIGURE 39: ILLUSTRATION OF DIPTE SPACECRAFT-BASED COORDINATE SYSTEMS PERTAINING TO THE SPACECRAFT CONFIGURATION AND SUBSYSTEMS 73
Figure 40: Sun sensor's structure frame ..... 76
Figure 41: Illustration of the 3-2-1 Euler angles rotation Sequence. ..... 78
Figure 42: Iluustration of the angle of attack and of the sideslip angle ..... 79
Figure 43: Main lens distortion types ..... 80
Figure 44: Relation between distances in the object plane and distances in the focal plane ..... 81
Figure 45: Attitude determination test bench, Catia model (left) and actual hardware (right) ..... 83
Figure 46: Bench frame ..... 84
FIGURE 47: InsTRUMENTS FRAME FOR TEST PLATFORM ..... 85
FIGURE 48: VECTOR TRANSFORMATION FOR THE ROTATION DEVICE MEASUREMENTS ..... 86
Figure 49: Magnetic field deviation due to the rotation device. ..... 87
Figure 50: Magnetic field direction in Daytona Beach,Fl ..... 88
Figure 51: Precision of the sun sensor by using a lighter ..... 89
Figure 52: Precision of a fixed dot tested with a flashlight ..... 90
Figure 53: Precision of fixed dot using a laser pointer. ..... 90
Figure 54: Displacement test between two dots separated by 18.4 degree in azimuth and elevation with the lighter ..... 91
FIGURE 55: DISplacement test between two points with flashlight ..... 91
Figure 56: Displacement test with a laser pointer ..... 92
Figure 57: Sun sensor test with three dots, laser pointer. ..... 92

## List of abbreviations

| BGS | British Geological Survey |
| :---: | :---: |
| DAB | Daytona Beach |
| DoD | United States Department Of Defense |
| ERAU | Embry-Riddle Aeronautical University |
| ESA | European Space Agency |
| G\&C | Guidance and Control |
| GMT | Greenwich Mean Time |
| GPS | Global Positioning System |
| GSFC | NASA Goddard Space Flight Center |
| GW | Gravity Wave |
| IGRF | International Geomagnetic Reference Field |
| MAG | The magnetometer's frame |
| NASA | National Aeronautics and Space Administration |
| nimu | Memsense Nano Inertial measurement Unit |
| NOAA | US National Oceanic and Atmospheric Administration |
| OBC | OnBoard Computer |
| RD | Reference document |
| SUN | The sun sensor's frame |
| WMM | World Magnetic Model |

## Chapter 1: Introduction

## 1 INTRODUCTION

### 1.1 PURPOSE OF THIS THESIS

The objective of this thesis is implementing and testing the first attitude determination algorithm of the Dipping Thermospheric Explorer spacecraft described in 1.2. By first algorithm, one means that it is intended to be a rough model of the attitude determination model in the way that all items are present but not developed in depth. This attitude determination module is unique because the design team wants to use low cost sensors and determine if the required accuracy can still be obtained. This is a first step in providing a cheap access to space. Therefore, this thesis provides a reference for the design team. It will help in setting up the future and more complex evolution of the algorithm.

It is going to be explained that the computation scheme requires finding the position of astronomical references like the sun or the earth. Sensors are needed to compute the position of these reference points. A sun sensor, a magnetometer and an earth sensor are used along with the "Tri-Axial Attitude Determination" or the "Quaternion Estimator" to determine the attitude of the spacecraft. Several assumptions have been made throughout the paper so as to design simple models.

The implementation of MATLAB algorithms to interface sensors and models is explained. These codes permit a first overview of the accuracy of the method described in this paper.

The results of this thesis show which aspects of the attitude scheme need to be enhanced to fulfill the attitude determination requirements stated in the CubeSat proposal [RD4], namely precision in the range of $1^{0}$ half cone angle at $3 \sigma$.

### 1.2 MISSION DESCRIPTION

A team of graduate masters students from Embry-Riddle Aeronautical University started the design of spacecraft under the direction of Dr Bogdan Udrea. The goal of the design process is to build a nanosatellite that flies through the thermosphere for a period of six months. This CubeSat is called "Dipping Thermospheric Explorer" or DipTE. The payload that will perform the measurements is a collection of miniaturized charged-particle spectrometer-based instruments developed by Fred Herrero at NASA Goddard Space Flight Center (GSFC). The DipTE platform is a $340 \times 100 \times 100 \mathrm{~mm}$ Cubesat that will carry spectrometers to measure perturbations in the neutral wind and temperature, in the ion velocity and temperature, and in the densities of the primary neutral and ionic species as the CubeSat orbit decays down to the lowest altitude of radio contact ( $\cong 150 \mathrm{~km}$ ). Figure 1 shows the general layout of the designed spacecraft.


Figure 1: The DipTE design

The resulting data set will have many applications to thermospheric and ionospheric science: principal among these, the team proposes to characterize the gravity wave (GW) spectrum as a function of altitude, latitude, longitude, and local time; to study GW sources by raytracing individual waves back to their sources; and to determine the spatial variability in the neutral winds.

It is assumed that the DipTE satellite will be released in a circular orbit above the altitudes of scientific interest for the mission. A propulsion system will be employed to make the orbit
elliptic. The apogee of the elliptic orbit will be at the altitude of the initial circular orbit. The 70 degree inclination of the orbit will stay the same. The total impulse capability of 313 Ns is sufficient to perform orbital maneuvers to bring the perigee 200 km lower than the apogee. The DipTE is aerodynamically stabilized by morphing a shape similar to that of a shuttlecock from deployable aero panels which produce stabilizing aerodynamic torques. Figure 2 is a view of the internal structure of the spacecraft where aeropanels, solar panels and the internal components can be seen.


Figure 2 Detailed design of the DipTE platform

As stated in the proposal (see [RD4]), aerodynamic stability analysis has shown that the configuration is stable and that passive attitude control can be achieved. An attitude determination system based on four sun sensors placed to have a field of view of more than $180^{\circ}$ and a magnetometer provides an attitude determination accuracy of about $1^{\circ}$ half-cone at $3 \sigma$. A three-axis digital magneto-inductive magnetometer will be employed to determine the direction and magnitude of the Earth's magnetic field vector. The "Tri-Axial Attitude Determination" (TRIAD) and quaternion estimation (QUEST) algorithms will be employed to combine the measurement of the sun-satellite vector and the magnetic field vector. The
accuracy of the attitude determination is required to be in the range of $1^{\circ}$ half cone at 30 . This requirement is derived from the science payload requirements.

A total of 13 thrusters are installed on DipTE: one orbital maneuver thruster of 1 N and 12 micro thrusters of 40 mN each for the reaction control system (RCS). The first design loop showed that the launch mass of DipTE should not exceed 4.5 kg .

### 1.3 MISSION ORIENTATION REQUIREMENTS AND ASSUMPTIONS

The DipTE mission, summarized in section 1.2, is described in detail in [RD4]. The payload position and orientation is the most important piece of knowledge required for the G\&C design.


Figure 3: Artist view the the DipTe in orbit

From this orientation, it becomes possible to determine how the spacecraft needs to be oriented with respect to the inertial geocentric frame throughout the orbit (see Figure 3). This thesis is not concerned with controlling the spacecraft but only with retrieving its current attitude with respect to the geocentric inertial frame. The following table summarizes the nominal spacecraft orbital parameters.

Table 1: Orbital plane characteristics

| Orbit apogee altude | 600 km |
| :--- | :--- |
| Orbit perigee altitude | 400 km |
| Eccentricity | 00145 |
| Orbital inclination | $72^{\circ}$ |

The attitude of the spacecraft needs to be determined with respect to the inertial frame of reference with an accuracy of $1^{\circ}$ half cone at $3 \sigma$ or better. This requirement can translate itself to each sensor and be considered as the design requirements. Off-the-shelf components can be found on the market with the following precisions:

Sun vectors are generally precise within 0.1 arc-second
Off the shelf magnetometers have a sensitivity of 2 nT
Earth sensors are accurate to $0.1^{\circ}$ half cone at $3 \sigma$
Therefore, it is intended in the final version of the attitude determination algorithm to use low cost sensors with high quality algorithms to obtain precisions as close as possible to what can be found in the market. This study is clearly a first step in providing a cheap access to space.

Some assumptions have been made to simplify the problem for this first design loop. The first assumption is that the self-stabilizing shuttlecock shape of the DipTE is such that the spacecraft does not require too many active control maneuvers. Consequently the spacecraft attitude is assumed to be close to the nominal attitude so that each of the astronomical reference points will be seen in the corresponding sensor's field of view without being disturbed (e.g. sun within the sun sensor field of view, earth in the horizon sensor field of view). It is also assumed that the current designed attitude determination will only be used in the day side of the earth, that is to say when the earth sphere is well lit up by the sun. This assumption makes sense because the sun sensor used works within the visible spectrum. Moreover, it is assumed that the designed algorithm is only intended to be used for nontumbling modes. The tumbling modes will be detected using the inertial measurement unit. A robust recovery control method based on inertial measurements would be used to recover from such an event.

The DipTE system layout can be consulted in appendix A along with the definition of the frames of each component and the rotation sequences used throughout this thesis. But a minor change has been made to the DipTE design here in that the algorithm is developed for a spacecraft which only features one sun sensor (and not four as it is for the DipTE). This simplifying assumption was made so that the combination of the four sun sensors' field of view of DipTE and the relative algorithm could be left out for the time being. It is assumed that the CubeSat has one magnetometer, one GPS, one sun sensor and one earth sensor.

Later adaptations will be necessary to fit the DipTE configuration, namely programming a sun sensor that is, in fact, the combination of four sensors.

## Chapter 2: Attitude Determination Algorithms

## 2 ATTITUDE DETERMINATION ALGORITHMS

This section describes all the different principles used for the attitude estimation. Attitude determination schemes are based on the use of vectors pointing to some astronomical reference points. Sensors are used to measure these vectors that can be related to both the spacecraft body-fixed frame and the earth centered inertial frame.

This section is first providing an introduction to the problem. In a second part, the TRIAD algorithm and the QUEST algorithm are explained since they represent the core of the attitude determination problem. The different ways to obtain reference vectors will be discussed in Chapter 3.

### 2.1 INTRODUCTION TO ATTITUDE DETERMINATION

To find the orientation (or attitude) of a body in space, information is needed on the movement of external reference points seen from the spacecraft. These reference points can be stars, the sun, or the location of the earth.

By using sensors, one is able to compute vectors from the spacecraft to the reference points. These sensors being inside de satellite, this first set of vectors is expressed in the spacecraft body frame. The description of the body-fixed frame of the DipTE satellite can be seen in Appendix A.1.ii. The same vectors can then be computed in the inertial frame by using astronomical or geophysical models.

By comparing the relative orientation of the body frame with respect to the inertial frame, one is able to compute the attitude of the spacecraft as pitch, roll and yaw angles. The mathematical methods used to compute this relative orientation are known as the "Three axis Attitude Determination" scheme (TRIAD) and the "Quaternion Estimator" schemes (QUEST).

The most common sensors used for attitude determination are:
a) Gyroscope: Senses the deviation of the spin axis of a rapidly spinning mass based on the conservation of angular momentum principle.
b) Accelerometer: Senses the linear acceleration of an object
c) Inertial measurement units (IMU): Provide accurate measure of the rotation of a spacecraft using rate gyros and translations of the spacecraft with accelerometers. IMUs are useful but not reliable for long missions due to the accumulation of computational errors.
d) Magnetometer: Senses the local magnetic field. For a non-magnetically disturbed environment (typically not a computer laboratory), the measurement corresponds to the magnetic field of the earth which has now been accurately measured and modeled.
e) Star trackers: Permit measuring the position of stars. These positions are then compared with a star catalogue thus permitting to find the attitude of a spacecraft. This is the most accurate type of sensing.
f) Horizon (or earth) sensor: Uses the Earth's albedo to compute the horizon of the earth (the limit between the cold space and the warm earth). This measurement permits finding the spacecraft nadir vector which is the vector going from the spacecraft to the center of the earth.
g) Sun sensor: Measures the angular position of the sun from the spacecraft. In this 1 arc minute accurate system, no measurement can be taken for satellites when they reach the night side of the earth on their orbit.
h) Global positioning system (GPS): the now well known GPS system is used on spacecraft and provides inertial coordinates of a spacecraft with respect to the geocentric frame.

Table 2: Potential accuracies of reference sensors at 3 sigma, [RD13] p. 309

| Reference object | Potential Accuracy |
| :---: | :---: |
| Star sensor | 1 arc second |
| Sun sensor | 1 arc minute |
| Earth sensor | 6 arc minutes |
| Magnetometer | 30 arc minutes |
| Narstar GPS | 6 arc minutes |

Each of these instruments provides an output vector expressed in the sensor's frame. Each sensor is mounted in a known orientation in the spacecraft body frame. By using the Euler definition of rotations, as described in appendix A.2, one is able to perform a transformation of vectors from the sensor frame to the body frame. The accuracy of potential accuracy of the different sensor is shown in Table 2.

At least two vectors are needed to compute the attitude of a body. For the current spacecraft design, the design team decided to use a magnetometer, a sun sensor and an earth sensor as attitude determination hardware for the first design loop. Several other sensors are planned for redundancy as the attitude determination algorithm becomes more robust and the satellite configuration becomes more completely defined.


Figure 4: Current attitude determination algorithm pattern

The actual attitude determination can be computed by two different schemes as stated earlier. The first one is called the "Three-Axis Attitude Determination" or TRIAD algorithm (see 2.2) which permits computing the attitude using two different sensor's measurements. The gross procedure that leads to obtain the final attitude is outlined in Figure 4 where it can be seen two path that retrieves vectors in the body and the inertial frame which are then blended together with the TRIAD. The "QUAternion ESTimator" or QUEST, a more powerful
algorithm, can also be used and is described in section 2.3. This latter algorithm has the advantage that it can compute the attitude using an "unlimited" number of measurements. These two methods are derived by Shuster in [RD1].

### 2.2 TRIAD ALGORITHM

The TRIAD algorithm is an attitude determination tool that permits to retrieve the attitude of a spacecraft using two observation vectors. To describe this method, one intends to use a sun sensor and a magnetometer respectively retrieving the position of the sun and dthe magnetic field seen from the spacecraft. Using these measurements, and by computing the sun position and the magnetic field direction in the inertial frame, one is able to retrieve the attitude in a deterministic way. This method is called deterministic because it does not feature any statistical analysis to optımize the computation and obtain a precise attitude matrix. It assumes vectors are retrieved simultaneously (which is not the case in real life).

This is a simple algorithm to understand, but it is powerful enough for the first iteration of the DipTE design loop. It has been derived by Shuster in [RD1].


Figure 5• The TRIAD general setup

As described in [RD1], the entire algorithm is based on 4 vectors:
Two reference vectors that describes the direction of the sun and the magnetic field in the inertial frame
Two measurement vectors from the sun sensor and the magnetometer that are expressed in the spacecraft body frame

Let $V_{\text {sun }}$ and $V_{\text {magneto }}$ be the reference vectors expressed in the inertial frame and $W_{s u n}$ and $W_{\text {magneto }}$ be the measurement vectors in the body frame. There exists an unique orthogonal matrix, the attitude matrix or direction cosine matrix $A$, that satisfies the relationship

$$
A \vec{r}_{L}=\vec{s}_{l}
$$

where it can be seen that the A matrix is a transformation matrix from the inertial frame to the spacecraft body frame. With the constructed vectors, one can construct two frames as:

$$
\begin{aligned}
& \left\{\begin{array}{c}
\overrightarrow{r_{2}}=\left(\overrightarrow{V_{\text {sun }}} \times \overrightarrow{V_{\text {magneto }}}\right) /\left|\overrightarrow{V_{\text {sun }}} \times \overrightarrow{V_{\text {sagneto }}}\right| \\
\overrightarrow{r_{3}}=\left(\overrightarrow{V_{\text {sun }}} \times\left(\overrightarrow{V_{\text {sun }}} \times \overrightarrow{V_{\text {magneto }}}\right)\right) /\left|\overrightarrow{V_{\text {sun }}} \times \overrightarrow{V_{\text {magneto }}}\right|
\end{array}\right. \\
& \left\{\begin{array}{c}
\overrightarrow{s_{2}}=\left(\overrightarrow{W_{\text {sun }}} \times \overrightarrow{W_{\text {magneto }}}\right) /\left|\overrightarrow{W_{\text {sun }}} \times \overrightarrow{W_{\text {magneto }}}\right| \\
\overrightarrow{s_{3}}=\left(\overrightarrow{W_{\text {sun }}} \times\left(\overline{W_{\text {sun }}} \times \overrightarrow{W_{\text {magneto }}}\right)\right) /\left|\overrightarrow{W_{\text {sun }}} \times \overrightarrow{W_{\text {magneto }}}\right|
\end{array}\right.
\end{aligned}
$$

Note that the first vector to be used is the sun vector. This is due to the fact that because the second and third vectors are built on the first one, this first vector needs to be the most precise one. The AeroAstro MediumSunSensor that has been selected and might replace the currently designed sensor has an accuracy of $1^{\circ}$ which makes it our most accurate sensor.

A simple expression for the attitude matrix is then

$$
A=M_{o b s} M_{r e f}^{T}
$$

with

$$
\begin{aligned}
& M_{r e f}=\left[\overrightarrow{r_{1}} \backslash \overrightarrow{r_{2}} \backslash \overrightarrow{r_{3}}\right] \\
& M_{o b s}=\left[\overrightarrow{s_{1}} \backslash \overrightarrow{s_{2}} \backslash \overrightarrow{s_{3}}\right]
\end{aligned}
$$

As it can be seen, for a computation stand point, this algorithm is easy to implement. But it does not take into account measurement errors so that a TRIAD covariance matrix analysis needs to be performed to get the most precise attitude matrix with this method. Analysis from Mike Jankowsky, ERAU master's student, showed that the covariance optimized version of the TRIAD does not increase the accuracy by a significant amount compared with the QUEST algorithm. Therefore it is intended to use the QUEST algorithm for the future design loops but this latter algorithm was not implemented for the current attitude estimation program.

### 2.2.1 Code structure

The general structure of the code computing the TRIAD algorithm is showed here.
The definition of the $V_{i}$ and $W_{i}$ vectors matches what is described earlier.

Calibrate the inertial vectors Vsun and Vmagneto (V1 and V2)
Get vectors Wsun and Wmagneto
Change Frame for Wsun from (SUN) to (BF)

Change Frame for Wmagneto from (MAG) to (BF)

Compute $\widehat{r_{l}}$ and $\widehat{\widehat{s}_{t}}$
$M_{r e f}=\left[\widehat{r_{1}}, \widehat{r_{2}}, \widehat{r_{3}}\right]$ and $M_{\text {obs }}=\left[\widehat{\hat{s}_{1}}, \widehat{s_{2}}, \widehat{\widehat{r}_{3}}\right]$

Compute the current attitude matrix $A=M_{o b s} M_{r e f}^{t}$

Compute the pitch, roll and yaw angle from A

### 2.3 QUEST ALGORITHM

The Quaternion Estimator method is more complex and more precise than the TRIAD. It has been derived by Davenport in 1968 and has two major advantages.

The first one is that it allows computing the attitude of a spacecraft using 2 or more sensors. For the DipTE design, we want to use 3 vectors: 1 sun vector, 1 magnetic field vector and 1 nadir vector. But this method offers an even better advantage in the sense that it uses statistical analysis to retrieve the attitude. Therefore, several measurements can be taken from each sensor and based on these measurements the best attutude matrix is found.


Figure 6. Sketch of vectors for the QUEST algorithm where the $b$ frame represents the BF frame

This algorithm is not implemented in the current attitude estimation program. But it will need to be featured in future versions.

### 2.3.1 Computation Process

The purpose of this method is to be able to choose an optimal attitude matrix using several measurements Let $\vec{w}_{i}^{B F}$ be the i-th measurement of a vector in the body frame and $\vec{v}_{l}^{\text {inertiat }}$ be the $i$-th vectors in the inertial frame This method can be seen In a mathematical way as mean to find a direction cosine matrix $A$ that minimizes the loss function

$$
L(A)=\frac{1}{2} \sum_{i=1}^{n} b_{i}\left|\vec{w}_{i}^{B F}-A \vec{v}_{i}^{\text {inertial }}\right|^{2}
$$

where the $b_{i}$ are nonnegative weights which represents the relative accuracy of each sensor with respect to each other so that the most accurate sensor is more "trusted" than the others. Let us define the unnormalized vector $W_{i}=\sqrt{b_{i}} \vec{w}_{i}^{B F}$ for measured vectors and $V_{i}=\sqrt{b_{i}} \vec{v}_{i}^{B F}$ for inertial vectors.
It is shown in [RD1] that this problem can be solved using a quaternion analysis and then going from the quaternion representation of the attitude to a direct cosine matrix A. This method is not derived here. Only the implementable part of the algorithm is showed. For more detail on the derivation, refer to [RD1].
Let

$$
\begin{aligned}
W & =\left[\overrightarrow{W_{l}}\left\langle\ldots<\overrightarrow{W_{n}}\right]\right. \\
V & =\left[\overrightarrow{V_{l}} \ell \ldots \backslash \overrightarrow{V_{n}}\right]
\end{aligned}
$$

By using the following expression and the method of Lagrange multiplier, one is able to rewrite the entire problem.

$$
\begin{gathered}
B=W V^{T} \\
S=B^{T}+B \\
Z=\left(B_{23}-B_{32}, B_{31}-B_{13}, B_{12}-B_{21}\right)^{T} \\
\sigma=\operatorname{tr}(B) \\
K=\left(\begin{array}{cc}
S-1 \sigma & Z \\
Z^{T} & \sigma
\end{array}\right)
\end{gathered}
$$

With the help of the variables, one is able to solve the optimal attitude matrix problem by solving the Eigen value problem.

$$
K q_{o p t}=\lambda_{\max } q_{\text {opt }}
$$

where $\lambda_{\max }$ is the largest Eigen value for this problem and $q_{\text {opt }}$ is the eigenvector which corresponds to this Eigen value. When $q_{\text {opt }}$ is found, one can compute the attitude matrix and be sure that it minimizes the loss function using

$$
\begin{aligned}
& q_{o p t}=\left(\begin{array}{l}
q_{1} \\
q_{2} \\
q_{3} \\
q_{4}
\end{array}\right) ; q=\left(\begin{array}{l}
q_{1} \\
q_{2} \\
q_{3}
\end{array}\right) ; Q=\left[\begin{array}{ccc}
0 & -q_{3} & q_{2} \\
q_{3} & 0 & -q_{1} \\
-q_{2} & q_{1} & 0
\end{array}\right] \\
& A(q)=\left(q_{4}^{2}-q \cdot q\right) \operatorname{Id}(3 * 3)+2 q q^{T}-2 q_{4} Q
\end{aligned}
$$

## Chapter 3: Sensors design and Inertial vectors computation

## 3 VECTOR DETERMINATION AND SENSORS DESIGN

The previous chapter described the method used to compute the attitude of a spacecraft using sets of vectors. Now that it is clear why these vectors are needed, this chapter describes how to compute them. For every reference point, vectors in the inertial frame and in the body frame are required. The computation of such vectors is going to be discussed for the sun sensor first, then the magnetometer and finally the earth sensor. All the methods described in this section have been implemented into MATLAB code.

### 3.1 SUN VECTOR DETERMINATION

A good reference point for earth satellites is the sun. It has been used as a reference point for a long time in spaceflight because it is a big and steadily emitting body. This section describes how this astronomical reference can be used for computing vectors that will then be used for attitude determination.

### 3.1.1 Sun vector in the Inertial Geocentric Frame

The sun pointing vector is almost the same if its origin is at the center of the earth than if it is at the center of mass (CoM) of the spacecraft, when computed in the inertial geocentric frame. Therefore, defining vector $\overrightarrow{V_{\text {sun }_{I}}}$ as the unit vector from the earth's center to the sun and $\overrightarrow{\text { SatelliteSun being the unit pointing vector from the }}$ satellite's CoM to the sun, on can write the approximation:

$$
\overrightarrow{V_{\text {sun }_{I}}} \approx \overrightarrow{\text { SatelliteSun }}_{I}
$$

Consequently the inertial sun vector can be computed at the center of the earth in a first approximation. To compute the $\overrightarrow{\text { SatelliteSun }}_{I}$ vector, one needs to apply a seven steps procedure as described in [RD3] which needs as input the current GMT time.

Figure 7 shows the considered configuration where $\overrightarrow{r_{\odot}}=\overrightarrow{V_{\text {sun }}}$ is the sun vector in the earth inertial frame $(\hat{l}, \hat{J}, \widehat{K})$.


Figure 7 shows the considered configuration where $\overrightarrow{r_{\odot}}=\overrightarrow{V_{\text {sun }_{I}}}$ is the sun vector in the earth inertial frame $(\hat{I}, \hat{J}, \widehat{K})$.


Figure 7 shows the considered configuration where $\overrightarrow{r_{\odot}}=\overline{V_{s u n_{I}}}$ is the sun vector in the earth inertial frame $(\hat{I}, \hat{J}, \widehat{K})$.


Figure 7 shows the onsidered configuration where $; \pi_{i=2}$ is the sun vector in the

$$
T_{U T 1}=\frac{J D_{U T 1}-2451545.0}{36525}
$$

One computes the mean longitude of the sun in the mean equator of date frame using

$$
\lambda_{M \odot}=280.460^{\circ}+36000.771 T_{U T 1}
$$

The mean anomaly for the sun can be computed by

$$
M_{\odot}=357.5277233^{\circ}+35999.050 T_{U T 1}
$$

To avoid possible numerical problems, one reduces $M_{\odot}$ and $\lambda_{M \odot}$ to be modulo 360 degrees. Thereafter, the ecliptic longitude can be computed by applying the equation of center

$$
\begin{gathered}
v_{\odot}=M_{\odot}+\left(2 e-\frac{e^{3}}{4}+5 \frac{e^{5}}{96}\right) \sin \left(M_{\odot}\right)+\left(5 \frac{e^{2}}{4}-11 \frac{e^{4}}{24}\right) \sin \left(2 M_{\odot}\right) \\
+\left(13 \frac{e^{3}}{12}-43 \frac{e^{5}}{64}\right) \sin \left(3 M_{\odot}\right)
\end{gathered}
$$

Where $e$ is the eccentricity of the earth's orbit around the sun $e=0.016708617$
Let the longitude and latitude of the ecliptic be

$$
\left\{\begin{array}{c}
\lambda_{\text {ecliptic }}=\lambda_{M \odot}+1.914666471^{\circ} \sin \left(M_{\odot}\right)+0.019994643 \sin \left(2 M_{\odot}\right) \\
\phi_{\text {ecliptic }}=0^{\circ}
\end{array}\right.
$$

Obliquity can be approximated using $\epsilon=23.439291^{\circ}-0.0130042 T_{U T 1}$
Finally the sun vector position magnitude can be computed using

$$
r_{\odot}=1.000140612-0.016708617 \cos \left(M_{\odot}\right)-0.000139589 \cos \left(2 M_{\odot}\right)
$$

Eventually, the sun vector can be computed by

$$
\overrightarrow{r_{\odot}}=\overrightarrow{V_{\text {sun }}^{I}}= \begin{cases}r_{\odot} \cos \left(\lambda_{\text {ecliptic }}\right) & \hat{I} \\ r_{\odot} \cos (\epsilon) \sin \left(\lambda_{\text {ecliptic }}\right) & \hat{J} \\ r_{\odot} \sin (\epsilon) \sin \left(\lambda_{\text {ecliptic }}\right) & \widehat{K}\end{cases}
$$

This sun vector expressed in the inertial frame has been coded and tested with MATLAB (see Appendix F). In the code, this sun vector is then normalized to have $\overrightarrow{V_{\text {sun }}}$ as a unit vector

### 3.1.2 Measured sun vector: Sun sensor design

A sun sensor retrieves a vector showing the sun direction. This vector is expressed in the frame attached to the sensor called SUN frame. Because the position and orientation of the sensor is known in the spacecraft body frame, a transformation can then be applied to the SUN frame to express the vector in the body frame.

### 3.1.2.1 Hardware

### 3.1.2.1.1 Sensor

The hardware used is a Webcam from Feiya Technology corp. This is a low-cost webcam using a MI-1320 CCD cell from Micron Technology Inc. The chip is made by APTINA Imaging, a new company created by Micron, and as product reference MT9M019


Figure 8 The Optina Imaging MT9M019
The following specifications were provided:
Optical format: $\quad 1 / 5$ inch
Image area: $\quad 2.83 \mathrm{~mm} \times 2.27 \mathrm{~mm}$
Active format (array): $1288 \mathrm{H} \times 1032 \mathrm{~V}$
Pixel size: $2.2 \mu \mathrm{~m} \times 2.2 \mu \mathrm{~m}$
Color filter array: RGB bayer pattern
Shutter: Electronic rolling shutter
Maximum Data rate: $64 \mathrm{Mp} / \mathrm{s}$
Master Clock: 64Mhz
Frame rate : $640 \times 480$ at 60 fps or $1280 \times 1024$ at 30 fps
This video camera is an affordable device (\$8). It was designed for home use, and is used in cellular phones, PC cameras and PDAs. Consequently the quality of the manufacturing does not comply with the quality required for real spacerated components. However, it is adequate for the purpose of this thesis.
The sensor is essentially a digital camera consisting of an array of pixels each containing a photodetector and a signal amplifier. The photodetectors are "Complementary Metal-Oxyde-Semiconductor" (CMOS). This CMOS is placed into a plastic case that is assumed not to distort the picture.


Figure 9: Picture acquisition process

The sensing process is shown in Figure 9. The CCD array consists of a homogeneous matrix of pixels, each of whose location within the array is accurately known. Based on the intensity of light on one or more pixels, a picture can be computed after a data conversion process done by the camera's electronic circuit.

### 3.1.2.1.2 Lens design

The CCD cell is a sensitive component operating over a precise range of wavelengths. It also has a range of detectable intensity with a maximum allowable intensity (not specified by the manufacturer). First tests with this device showed that the sun is too bright for this camera to be used directly with the manufacturer's lens. Therefore a new lens was designed.

The two design drivers for this lens are:

- Providing a Field Of View (FoV) $>90$ degrees

Reducing the intensity of the light seen by the CCD cell
Several different lenses were tested. They are the pinhole shape aperture, the cross shape and the dual-slot shapes as shown in Figure 10.


Figure 10: Different lens aperture shape

Each of the three lens shapes were manufactured and tested. The corss shape was designed to have a large field of view, but failed to meet the intensity criteria because the manufacturing process could not be as precise as required.

The dual slot aperture was tried to test an algorithm based on the computation of an axis using the intensity of two measured pixel lines. Two slots in one direction complemented with two slots in a $90^{\circ}$ rotated direction were supposed to help us find the sun position by finding the intersection between two intensity lines. Such a lens required more precision than the ERAU manufacturing lab was able to provide.

The pinhole shape seemed to be the best fit for the problem. Adding a "Black Polymer" filter to the lens permitted decreasing the light intensity. Therefore an increase in the diameter of the pinhole was possible resulting in an increase of the field of view.


Figure 11: Pinhole aperture geometry

As shown in Figure 11, the optics of the lens is quite simple. Consequently geometric optics is sufficient for calculations. Two parameters are important: the diameter of the pinhole and the length of the lens as described by the letter "a" in Figure 12. The thickness of the top part of the lens is a blocking parameter since it has to be as thin as possible but could not be too low due to manufacturing constraints. On the current lens, it is about 1.016 mm .


Figure 12: Geometric optics associated with the lens design

Let " $a$ " be the lens' length, $O$ be the focal point, $t$ the thickness of the top part of the lens, $\phi$ be the diameter of the aperture and $l_{c c d}$ be the length of the ccd cell (namely $l_{c c d}=2.83 \mathrm{~mm}$ ). The relationship between the length and the pinhole diameter are:

$$
\begin{gathered}
\phi=2 * a * \tan \left(\frac{F o V}{2}\right)-l_{c c d} \\
a=\frac{\phi+l_{c c d}}{2 * \tan \left(\frac{F O V}{2}\right)}
\end{gathered}
$$

This calculations shows that if one wants to have a $\mathrm{FoV}=90^{\circ}$ with a lens length of $a=3 \mathrm{~mm}$ the pinhole needs to be

$$
\phi=2 * 3 * \tan \left(\frac{90}{2}\right)-2.83=3.17 \mathrm{~mm}
$$

And for a FoV $=100^{\circ}$ with a lens of $a=2.5 \mathrm{~mm}$ one can achieve $\phi=3.13 \mathrm{~mm}$

A pinhole lens has also been designed for accuracy tests with a field of view of 65 degrees resulting in a 1 mm pinhole diameter. Two pinhole apertures have been used for tests as $\phi_{1} \approx 1 \mathrm{~mm}$ and $\phi_{2} \approx 3.17 \mathrm{~mm}$.

### 3.1.2.2 Software

The vector measured can be expressed in two different frames, the CCD and the SUN frames. The CCD frame allows us to express vectors in the picture frame using pixels as described in Figure 13. Therefore the CCD frame helps to describe whatever appears on the sensor's picture. Because the size of the cod cell is the same than the size of the retrieved picture, one can assume the CCD frame is attached to the CCD cell itself.

The SUN frame is attached to the camera casing and permits to describe the orientation of the structure holding the sun sensor in the body frame. Therefore, a bias exists between the CCD and SUN frame that could be found by accurate measurement on the placement of the cell in its casing.


Figure 13: Frame for the computation seen through the $C C D$ sensor. Xccd and Ycdd define the $C C D$ screen.

In addition, [RD5] shows the sensor's electric center can be shifted from the geometric center, therefore shifting all pixel measurements. More accurate tests would need to be performed to find such a misalignment. The biggest expected error resulting from such a misalignment is that the entire problem is coupled because the azimuth and elevation of a retrieved pixel cannot be expressed in pure coordinates ( $x, y$ ) in the defined frame.

Therefore it is assumed the two frames (SUN and CCD) are aligned.
The SUN frame is described here as:

Origin: center of the pinhole of the lens

Ox: passing through the two lens mount fixation holes
Oy: completes the right hand system
Oz: goes from the lens to the ccd cell


Figure 14: SUN frame description

The computation process requires 3 steps:

1) Detect pixels on the sensor that are above a certain intensity threshold. Obtain a binary picture with black and white pixels. The sun seen by the sensor looks like a disk.
2) Find the centroid of the white disk in the CCD frame (i.e. in pixels)
3) Using the camera field of view, the sun's diameter and the distance between the sun and earth, define a vector $\overrightarrow{C C D t o S U N}$ the sun vector expressed in the body frame

As stated earlier, a couple of assumptions had to be made. The attitude determination algorithm is started after a first stabilization of the spacecraft using the inertial measurements from an IMU. Therefore one can assume the sun would always be in the field of view of the camera when the spacecraft is on the day side of earth. This means that no partial-sun detections would occur (the
situation when the entire sun disk is not seen by the sensor). Also due to the high brightness of the sun, stars and other celestial objects_would not disturb the acquisition and pixel errors would not have much influence on the accuracy. The latter assumption is made because the conversion of the gray scale picture to a binary picture removes most of the noise on the picture as tests showed.

The purpose of the algorithm is to retrieve a 3-D vector pointing toward the sun. This is done by first computing the azimuth $\phi$ and the elevation $\theta$ of the sun position and then computing the wanted vector (see Figure 15).

Before computing the two parametric angles of the sun position $(\phi, \theta)$, the picture generated by the sensor needs to be analyzed to find the coordinates of the sun disk. This center computation can be done by many different ways. The Canny edge detector algorithm in [RD6] and the Hough transform in [RD7] have been successfully tried. However, these methods are very CPU intensive. The performances of the on board computer cannot permit such computation techniques. Therefore, a simpler method is used which finds the center of the shape of an image as the barycenter of this shape. This is not the most robust method, but it works well keeping in mind the assumptions we made earlier.

$$
C=\frac{\sum I_{j} r_{j}}{\sum I_{j}}
$$

Where $C=(x, y)$ coordinates of the centroid of the shape
$r_{j}=\left(x_{j}, y_{j}\right)$ coordinates of the pixel j
$I_{j}$ intensity of the pixel j on the picture ( recall on the final picture $I_{j}=\left\{\begin{array}{l}0 \\ 1\end{array}\right)$

This algorithm can't permit finding the centroid of a shape of an image where there is more than one body present. In this application, only pixels of intensity 1 are counted in the computation.


Figure 15: Graphic optics for sun vector determination

The relationship between the physics and the picture can now be derived. As it can be understood from Figure 15, the two parametric angles of the sun position can be found by the following relationships:

$$
\begin{gathered}
\phi=\arctan \left(\frac{x}{a}\right) \\
\theta=\arctan \left(\frac{y}{a}\right) \cos (\phi)
\end{gathered}
$$

where $x$ and $y$ are the coordinates of the barycenter of the disk, and " $a$ " the distance between the aperture and the CCD cell also called focal length.

This relation is true because pixels are equally separated in the sensor matrix as the characteristics of the CCD cell show and as discussed in [RD5] (equal spatial displacement). This relationship also implies that precision in the distance between the CCD array and the pinhole aperture affects the entire precision of the algorithm. For accuracy, the focal length of the camera was determined using the "Matlab Calibration Toolbox" as explained in appendix B.

When these azimuth and elevation angles are found, the following equations are used to obtain the unit sun vector:

$$
\left[\begin{array}{l}
x_{\text {sunvector }} \\
y_{\text {sunvector }} \\
z_{\text {sunvector }}
\end{array}\right]=\left[\begin{array}{c}
-r * \cos (\theta) * \sin (\phi) \\
r * \cos (\theta) * \cos (\phi) \\
-r * \cos (\phi)
\end{array}\right]
$$

where $r$ is the magnitude of the vector $(r=1)$. This algorithm have been implemented and tested. Results from these tests are shown in Chapter 4.

### 3.2 MAGNETIC FIELD VECTOR DETERMINATION

It has now been more than 400 years since the existence of a magnetic force around the earth was modeled by the English physician William Gilbert. But it was Carl Friedrich Gauss who first measured the strength and direction of this field.

The earth magnetic field is made of three different contributions: the main field generated by the earth core, the crustal field from earth's magnetized crustal rocks and the field coming from the currents flowing in the magnetosphere and the ionosphere.

Today the earth's magnetic field is well known both theoretically and experimentally. This field can be used as a reference for an autonomous spacecraft and it has been used for many years in space programs.

The designed attitude determination algorithm features a magnetometer able to measure the strength and direction of the surrounding magnetic field. This section explains how such a measurement can be helpful in providing a measurement vector for attitude determination.

### 3.2.1 The reference magnetic field - World Magnetic Model

### 3.2.1.1 Overview

Even though the magnetic field of earth is time and position dependent, it is possible to develop a simple model of it. The magnetic field of the earth can be modeled as a magnetic dipole (see Figure 16). This magnetic dipole is not aligned with the geographic north and south poles of the earth, but is tilted by an angle of approximately 11.5 degree.


Figure 16: Simple model of the Earth as a magnetic dipole for 2010

There are two ways of describing the location of the poles of the magnetic field. The first set of poles is called "dip poles" and is defined as points where the geomagnetic field of the earth is vertical. The uniqueness of this definition is in the fact that the two poles do not have to be antipodal with respect to the center of the earth. The dip poles are experimentally determined by looking for the points where the magnetic field has null horizontal components.

The second way to describe the poles of the magnetic dipole is the geomagnetic definition which comes from the different scientific models that have been derived. Therefore, the geomagnetic poles cannot be experimentally located like the dip poles. This model describes the earth as a magnetic dipole rotated $\sim 11.5$ degrees from the geographic north (for 2010). Figure 17 shows the past position of the north dip pole and north geomagnetic pole.


Figure 17 Locations of the north dip pole (red) and the geomagnetic north pole (blue) for the years 1900-2010, IGRF model, source British Geological Survey

The intensity of the magnetic field of earth has a maximum value of 67000 nanoTeslas ( nT ) and a minimum of 22000 nT as measured at the earth's surface Thus the magnetic field perceived in orbit can be lower than 22000 nT since the strength of the field evolves with $\frac{1}{\text { distance }^{3}}$

Predictions of the future changes in the magnetic field are now possible thanks to the long-term observations of the evolution of the variation in direction and intensity of this field

### 3.2.1.1.1 The IGRF WMM2005

The US National Oceanic and Atmospheric Admimistration (NOAA) and the British Geological Survey Geomagnetism Group (BGS) Joned efforts to develop a common model This model is now the standard model for the US DOD, the UK Ministry of Defense and the North Atlantic Treaty organization (NATO)


Figure 18: The CHAMP satellite, picture courtesy of GFC

This International Geomagnetic Reference Field (IGRF) World Magnetic Model 2005 (WMM2005) is only accounting for the main geomagnetic field $B_{\text {core }}$ so ṭhat the output of the model is $\vec{B} \approx B_{\text {core }}$. It does not model the influence of the atmosphere and the crustal component of the real field. It has been designed using the Danish Ørsted and the German CHAMP satellite (Figure 18) data combined with ground observation data.

The geomagnetic vector is described by the following parameters:

- The $X$ component or northerly intensity
- The $Y$ component or easterly intensity
- The $Z$ component as the vertical positive downward intensity
- F the total intensity as $F=\sqrt{X^{2}+Y^{2}+Z^{2}}$
- $H$ the horizontal intensity as $H=\sqrt{X^{2}+Y^{2}}$
- I the inclination between the horizontal plane and the field vector measured positive downwards as $I=\operatorname{atan}(Z, H)$
- $D$ the declination which is the angle between geographic true north and the field vector a $D=\operatorname{atan}(Y, X)$

The mathematic model accounting for the sources internal to the earth expressed in the geocentric frame is described as:

$$
\text { (Model) }\left\{\begin{aligned}
X & =-\sum_{n=1}^{N}\left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n}\left(g_{n}^{m}(t) \cos (m \lambda)+h_{n}^{m}(t) \sin (m \lambda)\right) \frac{d \breve{P}_{n}^{m}\left(\sin \varphi^{\prime}\right)}{d \varphi^{\prime}} \\
Y & =-\frac{1}{\cos \varphi^{\prime}} \sum_{n=1}^{N}\left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} m\left(g_{n}^{m}(t) \sin (m \lambda)-h_{n}^{m}(t) \cos (m \lambda)\right) \breve{P}_{n}^{m}\left(\sin \varphi^{\prime}\right) \\
Z & =-\sum_{n=1}^{N}(n+1)\left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n}\left(g_{n}^{m}(t) \cos (m \lambda)+h_{n}^{m}(t) \sin (m \lambda)\right) \breve{P}_{n}^{m}\left(\sin \varphi^{\prime}\right)
\end{aligned}\right.
$$

Where $\varphi^{\prime}$ is the latitude, $\lambda$ is the longitude and $r$ is the radius in a geocentric reference frame, $a$ is the standard earth's magnetic reference radius, $\breve{P}_{n}^{m}\left(\sin \varphi^{\prime}\right)$ are the Schmidt semi-normalized associated Legendre functions. Last but not least the $g_{n}^{m}(t)$ and $h_{n}^{m}(t)$ coefficients are the main output of this World Magnetic Model. In fact these values are the core of the algorithm and are updated by the NOAA/BGS every couple years (the next generation of the IGRF will be available in December 2009). We can therefore see the magnetic field as a sum of harmonics with varying coefficients. These coefficients (Gauss coefficients) are provided in tables. A sample of such a table is shown in Table 3. N in (Model) is the number of sets of coefficients provided in the tables.

Table 3: Example of the provided table. WMM2005 table

| $n$ | $m$ | $g_{n}^{m}$ | $h_{n}^{m}$ | $\dot{g}_{n}^{m}$ | $\dot{h}_{n}^{m}$ |
| :---: | :--- | :--- | :--- | :--- | :--- |
| 1 | 0 | -29556.8 |  | 8.0 |  |
| 1 | 1 | -1671.7 | 5079.8 | 10.6 | -20.9 |
| 2 | 0 | -2340.6 |  | -15.1 |  |
| 2 | 1 | 3046.9 | -2594.7 | -7.8 | -23.2 |
| 2 | 2 | 1657.0 | -516.7 | -0.8 | -14.6 |
| 3 | 0 | 1335.4 |  | 0.4 |  |
| 3 | 1 | -2305.1 | -199.9 | -2.6 | 5.0 |
| 3 | 2 | 1246.7 | 269.3 | -1.2 | -7.0 |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\cdots$ | $\ldots$ |

As can be seen in the table, the $h_{n}^{0}$ coefficients are blank because as it can be seen from the (Model), for $m=0$ the coefficients are on the $X$ and $Z$ axis multiplied with a $\sin (m)$ and the $Y$ component is multiplied by $m$. To compute the magnetic field at a location ( $\varphi^{\prime}, \lambda, r$ ), one just needs to compute the proper Gauss coefficients with the variation change provided as

$$
\begin{aligned}
& g_{n}^{m}(t)=g_{n}^{m}+\dot{g}_{n}^{m}\left(t-t_{0}\right) \\
& h_{n}^{m}(t)=h_{n}^{m}+\dot{h}_{n}^{m}\left(t-t_{0}\right)
\end{aligned}
$$

Where $t_{0}$ is the reference date of the model, here 2005.0 for the WMM2005. The magnetic field expressed in the geocentric frame can then be computed from (S).

Finally, one feature of this model is to provide the evolution of the field. The secular variation can thereby be computed as

$$
\left(S_{v a r i}\right)\left\{\begin{array}{c}
\dot{X}^{\prime}=-\sum_{n=1}^{N}\left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n}\left(\dot{g}_{n}^{m}(t) \cos (m \lambda)+\dot{h}_{n}^{m}(t) \sin (m \lambda)\right) \frac{d \breve{P}_{n}^{m}\left(\sin \varphi^{\prime}\right)}{d \varphi^{\prime}} \\
\dot{Y}^{\prime}=-\frac{1}{\cos \varphi^{\prime}} \sum_{n=1}^{N}\left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} m\left(\dot{g}_{n}^{m}(t) \sin (m \lambda)-\dot{h}_{n}^{m}(t) \cos (m \lambda)\right) \breve{P}_{n}^{m}\left(\overline{s i n} \varphi^{\prime}\right) \\
\dot{Z}^{\prime}=-\sum_{n=1}^{N}(n+1)\left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n}\left(\dot{g}_{n}^{m}(t) \cos (m \lambda)+\dot{h}_{n}^{m}(t) \sin (m \lambda)\right) \breve{P}_{n}^{m}\left(\sin \varphi^{\prime}\right)
\end{array}\right.
$$

This model has been tested by integrating the IGRF95 code to the designed algorithm.

### 3.2.1.2 Magnetic field sensor: magnetometer

Magnetometers are the devices especially designed to measure the magnetic field of a surrounding in 3 directions as it can be seen in Figure 19. While they may not be very accurate, they are used a great deal in the aerospace industry. These instruments are not particularly reliable attitude sensors in the sense that they measure the close magnetic field which can be disturbed by many unknown sources and therefore they provide data that one does not generally know how to correct. However they are used because they are accurate if one knows the surrounding environment, they can provide both the direction and the intensity of the field, have low power consumption and do not have moving parts.


Figure 19: Sketch of a 3-axis magnetometer

Since the earth's magnetic field strength decreases with the distance as $\frac{1}{r^{3}}$, satellites above 1000 km cannot use magnetometers as a main attitude sensor. There are two main categories of magnetometers:

- Quantum magnetometers, which use fundamental atomic properties
- Induction magnetometers, which use Faraday's law of magnetic inductance

In both types, the output of the measurements is converted by an electronics unit to provide numerical data.

### 3.2.1.2.1 The MemSense nIMU

The ERAU Aerospace Engineering department had for a previous project a complete Inertial Measurement Unit which is the Memsense Nano IMU. This IMU features

- 3-axis Gyroscopic measurement
- A 3-axis accelerometer
- A 3-axis magnetometer
- A thermometer unit for data correction


Figure 20: Memsense nIMU Functional Block diagram

The magnetometer included in this IMU is the only part used here (In later versions of the currently designed algorithm, the other data available through the nIMU will be used)

The nIMU's magnetometer has the following specifications:
Dynamic range: $\pm 1.9$ gauss
Drift: $\quad 2700 \mathrm{ppm} /$ degrees Celsius

Nonlinearity: $\quad 0.5 \%$ of the best fit straight line
Typical Noise: $\quad 0.00056$ gauss

Maximum noise: $\quad 0.0015$ gauss at $1 \sigma$
Bandwidth: $\quad 50 \mathrm{~Hz}$ at -3 dB point

### 3.2.1.2.2 Interfacing

The nIMU uses the $I^{2} C$ protocol (see [RD14]) or $R S 422$ protocol for sending data back. This device only outputs data; it is not capable of receiving commands. The manufacturer's package includes a USB adapter and a computer driver to convert data received from the USB port to serial RS422 data.


Figure 21: nIMU acquisition chain

This acquisition of data was done using the Matlab environment to get and convert the data into a usable form.


Figure 22: Sample structure of data sent by the nIMU

Data from the nIMU are formatted in a 38 byte package, also called sample, with a 13 byte header, 14 bytes of carried data and one checksum error byte. The
structure of this sample is well explained by Figure 22. Samples are retrieved at a baud rate of 115200 . The magnetic field data are included in the sample structure from byte \#25 to byte \#30. Each magnetic field component is represented by a set of 2 signed 1 -byte short integers that must be combined and converted to its corresponding numerical value before use. Therefore the Least Significant Byte (LSB) of a value and the Most Significant Bit (MSB) needs to be combined as:

$$
r a w_{v a l u e}=M S B \ll 8+L S B
$$

where the symbol $M S B \ll 8$ describes the bit shifting to the left by 8 bits of the value of MSB. The last step in the computation is to convert the raw value computed form the equation above into a usable engineering value by.the following equation (provided by Memsense).

$$
\text { Value }=\text { raw }_{\text {value }} * \text { digital sensibility }
$$

Where the digital sensibility of the magnetometer has the value 8.6975* $10^{-5}$ gauss/bit. Therefore one can easily follow the pseudo-code for the magnetic field acquisition:

Read a sample
Synchronize the sample

Read bytes 25 to 30

Combine the magnetic field MSB and LSB bytes

Convert the data with equation incorporating digital sensitivity
Output data are said to be temperature compensated but as the nIMU was used, it has been witnessed that the accuracy of data varies with the temperature. Also, a difference as high as $20 \%$ has been found between measurements from the written MATLAB code and measurement done with the provided Memsense program. These anomalies are rare and direction dependent. In most cases the error difference between the outputs is closer to $1 \%$ than $20 \%$. The manufacturer has been contacted but could not explain this phenomenon. The written MATLAB code has been reviewed by Memsense engineers and was said
to be correct. Therefore, one can only explain this difference by the magnetically noisy environment such as found in the ERAU computer laboratories.

### 3.3 EARTH SENSOR DESIGN

### 3.3.1 Introduction

Earth sensors, also called horizon sensors, detect the edges of the earth (commonly called the horizon). An illustration is provided by Figure 23 showing a spacecraft in orbit looking at a portion of the earth. This technology has been improved throughout the years. Consequently the variety of earth sensors is large. This principle has been often used on spinning spacecraft, but algorithm for non spinning spacecraft also exist. To avoid the problem of no measurement when in the dark side of the earth, most horizon sensors do not use the visible spectrum. The Infrared spectrum is preferred because with $\mathbb{R}$ sensors, sun interferences are reduced and they are also able to work on the night side of the earth. $\mathbb{R}$ sensors detect the earth by sensing the radiating heat. The horizon is the limit between the "cold" space environment and the "hot" earth radiation. But even though IR is preferred, visible light sensors, also called albedo sensors, have advantages too. This second type of sensor is often low-cost, has a faster response and provides a higher signal-to-noise ratio due to the radiation intensity in this spectrum


Figure 23. Horizon sensors provide a reference vector using the detected horizon line, source [RD3]

Most horizon sensors feature a scanning mechanism because the detection process requires finding two points on the horizon by scanning the biggest region possible (maximizing the sensor field of view cone in Figure 23).

The scanning mechanism of horizon sensors can be of different types. The simplest is the rigidly body-mounted low field of view sensor which can be installed on spinning spacecraft. A second type is the wheel mounted sensor where a spinning wheel provides the rotation to scan. Some sensors also feature a rotating prism that provides the scanning ability without actually rotating the sensor. For the current configuration, a fixed, non spinning earth sensor has been designed.

The earth sensor is here used in determining the nadir vector of the spacecraft. This vector is defined as the vector from the CoM of the spacecraft to the center of the earth. As described in Figure 24, a horizon vector is defined as a vector form the spacecraft tangent to the earth circle. The nadir vector makes an angle $\rho$ with the horizon vector.


Figure 24: Explanation of the nadir vector

This nadir vector describes the relative "down" direction with respect to the earth and is intended to be used in the QUEST algorithm in addition to the sun vector and the magnetic field vector. By definition, one can easily understand that the nadir vector expressed in the earth inertial frame can be found by using the GPS coordinate of the spacecraft which are in the inertial frame as

$$
\overrightarrow{V_{\text {nadir }}^{l}}=\left(\begin{array}{l}
-X_{S / C}^{g p s} \\
-Y_{S / C}^{g p s} \\
-Z_{S / C}^{g p s}
\end{array}\right)_{(\hat{1}, \hat{J}, \hat{K})}
$$

### 3.3.2 Computation scheme

The goal of the computation scheme is to determine the nadir vector by processing partial images of the Earth. The designed algorithm implements the computation scheme of scanning earth sensors found in the literature (see [RD11] section 5.4.3
p.261) to a fixed sensor retrieving a square image. The sensor used is the same as for the sun sensor. Its characteristics can be found in 3.1.2.1.1.

The DipTE body frame (see Figure 25) shows that the science payload is aligned with the velocity vector. Therefore, there is no room left on the front of the nanosatellite to fit a sensor. Consequently the horizon sensor needs to be at a different location. It has been chosen to put the horizon sensor optical axis aligned with the $Y$ vector of the body frame and located so that the optical axis passes through the CoM for convenience in the computation. If the optical axis is not crossing the CoM, which might be the case for the spacecraft since the CoM is moving with respect to time, a transformation will be necessary.


Figure 25: The earth sensor's field of view cone on a practical application

Considering Figure 26, assume that an image has already been properly corrected for the lens distortion and pixel errors. Further assume no shadows on the earth and a proper illumination of the globe.

The edge of the earth can be found by scanning the intensity of the pixels of the image on a circle. This circle has radius $d$ and its origin is located at the center of the picture O . This circle represents the projections, on the camera image plane, of
the base of a cone defining the field of view. This circle of origin O and radius $d$ defines the field of view of the sensor so that for maximizing the field of view, one needs to select

$$
d=\min (\text { width picture }, \text { height picture })
$$

This field of view circle is represented on the figure. The earth's edge is quite special because of sun reflection. It is a variation of luminosity which goes from dark space to bright horizon to average intensity earth. Therefore the most accurate way to find the edge is to find the point of highest intensity gradient.


Figure 26: Horizon sensor output picture (space set to be white for clarity)

Let $A_{1}$ and $A_{2}$ be the points found on the earth's horizon. These points are determined as the intersection points between the image circle and the horizon of the earth E is the angle between $\overrightarrow{O A_{1}}$ and $\overrightarrow{O A_{2}}$. Let $\vec{N}$ be the nadir vector (of magnitude $N$ ) in the inertial frame expressed in GPS coordinates. Let the horizon
plane be define as a cut of the earth globe by the points $A_{1}$ and $A_{2}$ and parallel to the $Y_{B F}$ axis (see Figure 27).


Figure 27: Angular radius geometry, adapted from [RD15]

Let $\rho$ be the angular radius of the earth defined as the angle between the nadir vector and an horizon vector (eg $\mathbf{V}_{\mathbf{1}}$ or $\mathbf{V}_{\mathbf{2}}$ which respectively goes from the CoM to $A_{1}$ and $A_{2}$ as described on Figure 27) seen from the current spacecraft position. This angle expresses the fact that the size of the earth seen from the spacecraft changes with the altitude h . The angular radius $\rho$ can be computed with the following equation

$$
\cos (\rho)=\frac{\sqrt{\|\vec{N}\|_{1}^{2}-R_{e a r t h}^{2}(l a t)}}{\|\vec{N}\|_{1}}
$$

recall $\|\vec{N}\|_{I}=N$ is known thanks to the GPS system and the earth radius is a function of the latitude by the simplified model for visible light relationship:

$$
R_{\text {earth }}(\text { lat })=R_{\text {equator }}\left(1-\frac{R_{\text {equator }}-R_{\text {polar }}}{R_{\text {equator }}} \sin ^{2}(\text { lat })\right)
$$

Let $f$ be the focal length of the sensor's lens. Let $\gamma$ be the half cone of the scan defined as

$$
\gamma=\frac{F o V}{2}
$$



Figure 28: Geometry of the roll angle $\eta$

Therefore $\rho, \gamma$ and $E$ are known. The scanner roll angle also called nadir angle is denoted by the letter $\eta$. It is the angle between the optical axis of the camera (which is aligned with the $Y$ axis of the body frame) and the nadir vector (see Figure 28). This nadir angle is expressed as the solution to the following spherical law of cosine equation

$$
\cos (\rho)=\cos (\gamma) \cos (\eta)+\sin (\gamma) \sin (\eta) \cos \left(\frac{E}{2}\right)
$$

The geometry of this equation is shown in Figure 29 where can be recognized the different angles already defined.


Figure 29: Geometry of the nadir angle equation

The earth is seen with a minimum angular radius $\rho \cong 66^{\circ}$ for the highest altitude of the DipTE orbit which is 600 km . The camera field of view is approximately such that $\gamma=16.5^{\circ}$. Therefore $\rho>\gamma$ which tells us the previous equation has a unique solution.

Solving the above equation implies that one now knows the nadir angle $\eta$.


Figure 30: Configuration of negative pitch and tilted to the left compared to Figure 31

The spacecraft pitch and yaw angle are coupled. On an output image, the relative position of the earth with respect to the mid-line of the image is providing both information on the pitch and the roll inclination of the spacecraft. One can understand that, for a null yaw angle, the displacement of the earth from the right to the left of the image corresponds to a pitch up motion. But, for a fixed pitch, one can see that the same displacement describes a rotation around the +Yaw axis clockwise. Therefore, one can define $p$ as an angle which is the combination between the spacecraft pitch angle and the spacecraft yaw angle.

This $p$ angle can be found from the output image. Let the vertical axis cutting the image plane through the middle (called mid-line in Figure 30) be our reference. Angles can be computed counterclockwise from this axis. Let $\theta_{\text {pickoff }}=180^{\circ}$. Let $\theta_{1}^{\text {hort }}$ be the angle between the mid-line and the vector $\overline{O A_{1}}$ described earlier. Let
$\theta_{2}^{\text {hori }}$ be the angle between the mid-line and the vector $\overrightarrow{O A_{2}}$. With this definition of angles, one can relate to the definition of $E$ as $E=\theta_{2}^{\text {hori }}-\theta_{1}^{\text {hori }}$.
The $p$ angle can be expressed as

$$
p=\frac{\theta_{2}^{\text {hori }}+\theta_{1}^{\text {hori }}}{2}-\theta_{p i c k o f f}
$$

which describes an angle from the mid-line to the median of the $E$ angle as can be seen on Figure 31.

To conclude, we can say that:
Let $E_{1}$ and $E_{2}$ be the values of the angle $E$ such that if $E_{1}>E_{2}$. The configurations corresponding to $E_{1}$ and $E_{2}$ are configuration (1) and configuration $\langle 2\rangle$, respectively. Configuration $\langle 1\rangle$ represents the situation where the spacecraft is tilted about the +roll axis in a counterclockwise direction. Configuration $\langle 2\rangle$ is such that the roll angle is less than for configuration $\langle 1\rangle$.

If the yaw angle is null : $\quad p>0$ means negative pitch
$p<0$ means positive pitch
This can also be illustrated by comparing Figure 31 and Figure 30.


Figure 31: Configuration of positive pitch.

From the two attitude parameters and one can now compute the nadir vector expressed in the earth sensor's frame as

$$
\vec{N}=\left[\begin{array}{c}
\sin (\eta) \cos (p) \\
\sin (\eta) \sin (p) \\
\cos (\eta)
\end{array}\right]
$$

Again, by knowing the location and orientation of the earth sensor in the body frame, one is able to compute the nadir vector in the spacecraft body-fixed frame.

## Chapter 4: Tests

## 4 TEST PROCEDURE AND RESULTS

Chapter 2 and Chapter 3 describe the method for computing the attitude of the spacecraft. To check the soundness of this method, the entire algorithm has been implemented into MATLAB code. Chapter 4 is intended to describe tests that have been performed to validate the written code. Simple and easy to setup test benches have been designed. The precision of the results obtained is a rough estimate of the algorithm precision because most of the tests are based on an eyeballing method.

After the first tests on the sun sensor, it has been noticed that an angle in the real world is not retrieved as an angle by the camera. This effect is mostly due to distorted image. This distortion is coming from the lens which modifies the image. A lens correction algorithm had to be designed to correct this effect. It is described in appendix B . This correction algorithm has not improved the data by a sufficient amount so that for future design, this algorithm would need to be reworked.

A rotation test platform has also been designed to permit rotating the sensors around the pitch, roll and yaw axis. This device was designed to allow accurate rotation of the sensors to then test the accuracy of the TRIAD and QUEST algorithm. This device has not been used for the current thesis tests, but will be useful later, providing an accurate test platform to simulate the orientation of the DipTE spacecraft. This design can be found in appendix C .

A remark has to be made on the terms "accuracy" and "precision" used in this section. The accuracy of a measurement describes how far off a value is compared to its true (or expected) value. The precision of the measurement is the degree to which repeated measurements under unchanged conditions show the same results. It is important to note that no statistical analysis have been made in this section besides the sun sensor analysis. The accuracy is approximated by looking at the difference between angles inputted in the system (and eyeballed) and angles retrieved by the program. Even though many tries have been performed to have the shown values, the precision of the algorithm was not computed due to the poor accuracy of the sensors.

### 4.1 ACCURACY OF THE DESIGNED SUN SENSOR

### 4.1.1 Test procedure

The tests performed were done using a board located at 3 meters from the sensor. On this board, dots of light were created by using a lighter, a flashlight or a laser pointer. Three tests have been performed for two reasons:

- Sense the consistency of the vector corresponding to a non moving dot projected on the board
- Sense the spatial accuracy of the sensor with the manufacturer lens to answer the question: does a degree in real life correspond to a degree processed by the algorithm?

Throughout the research, the designed lens was not available due to the busy schedule of the ERAU manufacturing lab. Therefore the manufacturer's lens has been used for most of the tests. Tests have shown that the lens adds distortion to the picture as discussed in the previous introduction.

### 4.1.2 Sun sensor test 1: Fixed dot test



Figure 32: Sun sensor spatial accuracy test procedure

As stated before, this test is intended to show how consistent are the data processed by the sensor and the code. The azimuth $\phi$ and elevation $\theta$ of a dot of light, as
computed in 3.1.2, retrieved with a precision of $\sigma_{A Z I}=0.067$ and $\sigma_{E L E}=0.036$ for a non moving dot projected on the board. Figure 33 shows data used to compute these standard deviations from the fixed laser dot projected on the board. The values are circumscribed by a circle to show the maximum distance between dots. This test shows that the maximum angle between dots in azimuth is $\phi_{\max }^{e r r o r}=0.21^{\circ}$ and the maximum distance in elevation is $\theta_{\max }^{\text {error }}=0.12^{\circ}$.


Figure 33: Fixed dot test of the sun sensor with laser pointer, for a randomly chosen point, shown with circle fitting data.

The results from this test are different depending on the source of light used. This test is more precise if a punctual light dot is detected than if a flashlight, which provides a spread light dot, is used. One test has been performed per type of source. The accuracy on the fixed dot can be summarized in the following Table 4. The charts corresponding to these tests can be seen in Figure 51,Figure 52 and Figure 53 from appendix $E$.

Table 4: Sensitivity of the sun sensor

| Type | $\phi_{\max }^{\text {error }}$ | $\theta_{\max }^{\text {error }}$ |
| :---: | :--- | :--- |
| Flashlight | 045 deg | 060 deg |
| Lighter | 030 deg | 010 deg |
| Laser Pointer | 021 deg | 012 deg |

Based on this test, one can consider the accuracy of the sun sensor to be within 0.6 degrees for rendering the position of the sun, when using the manufacturer's lens which has a 33 degrees field of view. This stated precision, as defined earlier, is the best that can be obtain with the current camera. Any value output by this sensor would need to be considered at $\pm 0.2^{\circ}$ (considering the laser pointer as the simulation light source).

### 4.1.3 Sun sensor test 2: Homogeneous displacement on two directions

This test is intended to compute the position of dots on the board and compute their relative distance. Here, one is trying to see if a degree in real life corresponds to a degree computed with the algorithm. This displacement test is therefore the most important because it shows the mapping of the lens. The test was performed with different light sources: a lighter, a flashlight and a laser pointer. An illustration of the method is shown in Figure 32 where the wall with the ruler and the lighter can be seen. Due to time imitations, a more precise procedure could not be accomplished. [RD5] shows a test bench that would permit knowing all the parameters of the camera such as the position of the electric center of the CCD cell. Such a set up and procedures are beyond the scope of this study.

The sensor with manufacturer's lens is used. Two rulers of 1 meter are perpendicularly placed so that the angle seen by the camera between point $\mathrm{P} 1(0,0)$ (in the ( $\mathrm{X}, \mathrm{Y}$ ) plane) and point $\mathrm{P} 2(-1 \mathrm{~m}, 1 \mathrm{~m})$ is 18.44 degrees in magnitude for both azimuth and elevation as shown in Figure 32.

Graphical results are shown in Figure 54, Figure 55, Figure 56 from Appendix E where the coordinates of point P1 and P2 are not important, only the difference
between them is important. The expected result of this test is to find that the code is retrieving the same value of 18.4 degrees for both elevation and azimuth. The accuracy on this test can be discussed from the following table.

Table 5: Spatial mapping of the sun sensor

| Type | Measured Azımuth angle | Accuracy in Elevation |
| :---: | :---: | :---: |
| Flashlight | -1307 deg | 1056 deg |
| Lighter | -13346 deg | 1155 deg |
| Laser Pointer | -1088 deg | 1022 deg |

As seen in Table 5, the expected values are not close to the measured values Therefore the sun sensor's code had to be modified to include a lens correction algorithm as explained in appendix B This correction algorithm does improve the accuracy of about 0.5 degree which is not significant. More work will be needed on this lens correction scheme.

To be consistent on the method developed, tests with the designed lens had to be done. Recall this lens only features a pinhole, and no lens deflection is expected. Let P0 be a dot located at $P 0=(0,0)$ (in the $(\phi, \theta)$ plane). Let $P 18=\left(-18.4^{\circ}, 9.2^{\circ}\right)$ and finally $P 33=\left(-33.7^{\circ}, 18.4^{\circ}\right)$.

Dots of light have been placed with a laser pointer on the board at these point coordinates. The results of this test are shown in Table 6 and Figure 57. It can be seen from this table that, here again, the retrieved dot positions are different than expected.

Table 6: Test of spatial homogeneity with the designed lens

| Point type | Azimuth | Elevation |
| :---: | :---: | :---: |
| P18 | -11788 deg | -693 deg |
| P33 | -2265 deg | -1160 deg |

The precision of the test with the pinhole aperture can be summarized in Table 7. It shows that data are shifted about $35 \%$ from the value they are suppose to have.

Table 7: Accuracy of the sun sensor by using a pinhole lens

| Type | Error in Azimuth | Error in Elevation |
| :--- | :---: | :---: |
| P18 | $35.93 \%$ | $24.67 \%$ |
| P33 | $32.79 \%$ | $36.96 \%$ |

One conclusion from these tests is that the homogeneity assumption (same spatial scaling on both axes) appears to be true for the current CMOS/lens combination since pinhole lens shows roughly the same error in both directions. But the strong deflection of the picture leads to the idea that the ccd cell plastic protection case might deflect the light ravs so that, even if no lens is used, the picture is deflected.

### 4.2 ACCURACY OF THE MAGNETOMETER

To check the device, magnetometer measurements were made at two different physical locations. The magnetometer frame or MAG is described by the manufacturer in Figure 34.


Figure 34: nIMU reference frame

The reference WMM2005 model from NOAA gives the output so that:
Horizontal north intensity is along the $+X$ value
Horizontal east intensity is along the $+Y$ value
Vertical intensity is along the $Z$ (completes the right hand system)
Therefore, to be able to compare the data from the model and measurements, the Xmag of the sensor is aligned with the local magnetic north.

Important note: Measurements always include errors. Besides the fact that the magnetometer is imprecisely oriented so that its Xmag axis point toward the local magnetic north, one needs to remember that the nIMU is connected to a computer with a limited wire length. Therefore measurements taken with the nIMU in a laboratory, full of computers and other electrical components, are in a disturbed electromagnetic environment. To compensate for this fact, the measurements have been taken with a laptop far from any electronic device (within the limit of the power cord).

The following tables are showing the difference in the measurement of the magnetic field in two cities of Florida: Ormond Beach and Daytona Beach.

Table 8: Magnetic field of Daytona Beach

| Magnetic field in DAYTONA | X (gauss) | Y (gauss) | Z (gauss) |
| :---: | :--- | :--- | :--- |
| Reference (WMM) | 0.24229 | -0.025056 | 0.403196 |
| Measurement | 0.21987 | -0.02548 | 0.355728 |

Table 9: Magnetic field in Ormond Beach

| Magnetic field in Ormond Beach | X (gauss) | Y (gauss) | Z (gauss) |
| :---: | :--- | :--- | :--- |
| Reference | 0.242443 | -0.024913 | 0.403234 |
| Measurement (best) | 0.229527 | -0.0250488 | 0.45488 |

The following table shows the percentage difference between what was expected and what is computed. One cannot talk about "error" here since the WMM is just a model and the
magnetometer is measuring the real magnetic field. Table 10 shows that the vertical component of the magnetic field has a deviation from the model of more than $10 \%$. This fact needs to be taken into account to achieve the required attitude determination precision later on.

Table 10: Measured accuracy of the nIMU magnetometer

| Magnetic field | Bias X | Bias Y | Bias Z |
| :---: | :--- | :--- | :--- |
| Daytona | $9.2 \%$ | $1.6 \%$ | $11.77 \%$ |
| Ormond Beach | $5.33 \%$ | $0.5 \%$ | $11.35 \%$ |

### 4.3 ACCURACY OF THE EARTH SENSOR

The hardest part in designing an earth sensor is actually to check the accuracy of the method. One cannot directly check the accuracy of the nadir vector. But the attitude angles (pitch and roll angles) can be measured and compared to the actual physical camera rotation. Therefore, the horizon sensor test is based on how accurately this algorithm can retrieve the current attitude described by the rotation device namely pitch and roll angles.


Figure 35. Earth sensor's test bench

A picture of a disk of radius 48.9 cm (so as to represents the earth) have been printed on a board located at a distance $D=21.8 \mathrm{~cm}$ from the sensor. This picture corresponds to a white circle printed on a black board so that the scaling of this problem represents an orbit at altitude of $\mathrm{h}=600 \mathrm{~km}$ which perfectly matches our problem ( $\rho \cong 66^{\circ}$ ). For convenience, the problem is decoupled into pitch and roll. Figure 35 shows the setup of such a test

### 4.3.1 Pitch accuracy of the Horizon sensor

The pitch accuracy of the program needs to be determined. The pitch accuracy is checked by rotating the board in front of the camera. The camera optical axis passes through the picture rotation point. This procedure is best illustrated by the following Figure 36 which shows the protractor where physical rotations angles are read from (with an eyeballing accuracy of 1 degree) and that the rotation point of the picture is aligned with the optical axis of the camera. Measurements showed a constant roll angle during this test.


Figure 36 Pure pitch test procedure for the horizon sensor

The pure pitch test summarized in Table 11 shows that an error of about $1.5^{\circ}$ is made on the first measurement test 2 resulting in a $14.6 \%$ error difference. This first measurement inaccuracy is due to the eyeballing process taking place for the first measurement. But for the other pitch angles, the error is about $1.5 \%$. Therefore one roughly considers the earth sensor to be precise within $1^{\circ}$ in pitch. This test also shows the consistency of the output from negative to positive pitch position.

Table 11: Pure pitch test, lens correction applied

|  | Physical pitch | Computed pitch | Delta pitch | Error |
| :--- | :---: | :---: | :---: | :---: |
| Test 1 | $0^{\circ}$ | $1.8087^{\circ}$ | - |  |
| Test 2 | $10^{\circ}$ | $10.345^{\circ}$ | $8.536^{\circ}$ | $14.6 \%$ |
| Test 3 | $20^{\circ}$ | $20.651^{\circ}$ | $20.306^{\circ}$ | $1.5 \%$ |
| Test 4 | $0^{\circ}$ | $1.636^{\circ}$ | $0.985^{\circ}$ | $1.5 \%$ |
| Test 5 | $-10^{\circ}$ | $-8.508^{\circ}$ | $-10.145^{\circ}$ | $1.45 \%$ |
| Test 6 | $-20^{\circ}$ | $-18.974^{\circ}$ | $-20.466^{\circ}$ | $2.32 \%$ |

### 4.3.2 Roll accuracy of the Horizon sensor

The pure roll procedure is done using the camera pod rotation point as can be seen in Figure 37. The hard part of this measurement is to have an accurate reading of the mechanical rotation of the sensor.


Figure 37: Pure roll test for the horizon sensor

Table 12: Pure roll rotation test

|  | Physical roll angle | Computed roll (deg) | Delta roll (deg) | error |
| :--- | :---: | :---: | :---: | :---: |
| Test 1 | 0 deg | 51,9624 | - | - |
| Test 2 | 10 deg | 60,9109 | 8,9485 | $10.5 \%$ |
| Test 3 | 20 deg | 70,1212 | 19,2103 | $4 \%$ |
| Test 4 | 25 deg | 75,0479 | 24,9267 | $0.3 \%$ |

Previous roll tests showed the sensitivity of the sensing on this axis. The low field of view of the camera ( $\mathrm{FoV} \cong 33^{\circ}$ ) makes it difficult to take many measurements. Results from this test are shown in Table 12 where the delta roll is computed assuming the previous value is the new reference. It showed that the code seem to be accurate (Test 3). This test would need to be done again with smaller angular steps to be able to really have a good overview of the accuracy of the roll computed by the program.
The precision can only be determined by running multiple tests and studying the statistical error retrieved by the code. But for this first design loop, the results are considered sufficient in the sense that it proves that the written code is performing its duty. Moreover, the percentage error stated in this section does not only come from the code but also from the inaccuracy in the reading of physical angles set. But as a general conclusion one can state that this set of test validates the method and the implementation of the horizon sensor simulator that can be looked up in Appendix F.

### 4.4 TRIAD ATTITUDE DETERMINATION ALGORITHM TEST

By using the TRIAD algorithm with a sun sensor and a magnetometer placed on the rotation device, a first preview of the attitude determination scheme's efficiency can be computed. No significant results were found from the few tests performed.

These results show two things. First of all the system is found not to be uncoupled as for a fixed yaw angle, the output shows an evolving yaw angle. Rotation on the pitch axis has mainly an effect on the retrieved pitch but the simple frame transformation performed does not seem to be enough to correct the output.

This test also shows that the worst retrieved pitch accuracy has a $30 \%$ error, which is good since, as discussed before, the TRIAD algorithm cannot retrieved accurate data unless each
of the items involved are accurate. More effort would need to be put in the correction of the program to then have better test results.

## Chapter 3: Conclusions of this Thesis

## 5 CONCLUSIONS

### 5.1 SOURCES OF ERROR \& ACCURACY OF EACH SENSOR

Everybody who has ever worked with actual hardware on a scientific project knows how complex problem can become when all the sources of errors are taken into account. The designed attitude determination scheme features many devices therefore increasing the number of sources of error. Figure 38 shows the different sources of errors in the procedure.


Figure 38: The chain of sources of error

No statistical analysis has thus far been done but it is necessary for a future version of the code to feature such a procedure to have a better estimate of the precision of each item. For a future version of the program, more work would need to be provided in setting up different test benches to check the accuracy of each item.

Tests show that sun sensor measurements are accurate within $8^{\circ}$ (as described in 4.1.3). The error of the measurement is mainly due to a difference between actual angles and retrieved angles due to image deflection. This can be corrected by developing a more robust lens correction algorithm. This correction would also have a good effect on the accuracy of the earth sensor designed base on the same camera.

The magnetometer seems to have a steady deviation from the model of $11 \%$ on its $Z$ axis. The inaccuracy of this sensor is mostly due to the test environment which can be greatly perturbed by the magnetic field of surrounding computers and lights. The accuracy is expected to be increased after the sensor has been implemented in the DipTE platform which features a less noisy electrical environment compared to computer laboratories where tests were performed.

The earth sensor is the most experimental part of the code since it was hard to come up with the proper algorithm and because the test procedure requires an extremely high accuracy. Nevertheless, the error in accuracy is about $1.5 \%$ in pitch and $4 \%$ in roll. More tests would need to be performed for characterizing this sensor behavior mainly by setting up more precise test procedures than the white circle on black board test. It also needs to be tested on the real DipTE platform to check how efficient the algorithm is in correcting data from non pure rotations.

### 5.2 ACCURACY OF THE ENTIRE ALGORITHM

An insufficient amount of time has been spent on trying to increase the precision of each part of the program due to the amount of work to be done. In fact, each sensor in this thesis could have been the object of a thesis research. To increase the overall precision, one would need to start working on a more complex and accurate algorithm for each sensor. More accurately manufactured lenses would also need to be done or a proper selection of on-shelf components like the AeroAstro MediumSunSensor.

It is projected to add electrical motors to the rotation device. This will permit having a greater accuracy on physical rotation angles for tests. First tests of the TRIAD algorithm showed an error discrepancy ranging from $1 \%$ to $29 \%$ in the worst case. This suggests that there might be an error in the code which needs to be found and corrected.

### 5.3 OVERALL CONCLUSION

The methods to retrieve the position or orientation of reference celestial phenomenon have been studied and implemented so that they could be used for attitude determination. A sun sensor and an earth sensor have been designed, implemented and tested. A magnetometer has been interfaced with the algorithm to have a measurement of the spacecraft-surrounding magnetic field. The TRIAD algorithm has been implemented and tested with theoretical values. Tests have been performed to have a sense of the first accuracy of the attitude determination method. All this represents a significant amount of data that are explained in the present report. Besides the specific points mentioned in the above sections, the goal of this thesis is reached. Recall it is to setup a first rough platform for the DipTE attitude determination algorithm. Now, more time can be spent in increasing the complexity of the models of each item. After a couple of design interations the overall algorithm will be able to fulfill the accuracy requirement of $1^{\circ}$ half cone at $3 \sigma$.

## 6 REFERENCE DOCUMENTS

[RD1] "Three-Axis attitude determination from vector observations", M.D. Shuster and S.D Oh, Journal of Guidance and Control, Vol. 4, No.1, Jan-Feb 1981, pp. 70-77
[RD2] Spacecraft attitude determination and control, Part III on attitude determination, ed. by James R.Wertz, ,1978
[RD3] "Fundamental of astrodynamics and applications", ed. 3, ed. by David A. Vallado, Space Technology Library, 2007
[RD4] Dipping Thermospheric Explorer (DipTE) proposal, Bogdan Udrea, Embry-Riddle Department of Engineering, 2009
[RD5] "Design and characterization of a sun sensor for the SSETI-ESEO project", P. Rodrigues \& P Ramos, XVII IMEKO WORLD CONFERENCE - Metrology for sustainable development, 2006
[RD6] "A computational approach to edge detection", John Canny, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 8, Issue 6, pp. 679-698, published in 1986
[RD7] "Use of the Hough transform to detect lines and curves in pictures", Hart/Duda, Communications of the ACM, Volume 15, issue 1, pp. 11-15, published in 1972
[RD8] "The US/UK World Magnetic Model for 2005-2010", McLean/Maus/Dater/Macmillan/Lesur/Thomson for NOAA \& BGS, NOAA Technical Report NESDIS/NGDC-1
[RD9] "Memsense Nano IMU product specification user's guide", No. PSD-0822, Rev. A, memsense.com
[RD10] "Contributions of spherical harmonics to magnetic and gravitational fields", Carlos Roithmayr of NASA Langley Research Center, NASA/TM-2004-213007, 2004
[RD11] Fundamentals of Space Systems, ed. by Vincent L. Pisacane, ed. 2, JHU/APL series in science and engineering, Oxford University press, 2005
[RD12] "Automatic Correction of lens distortion by using Digital Image Processing", Harri Ojanen, pre-print, 1999
[RD13] Spacecraft systems engineering, ed. By Fortescue/Stark/Swinerd, Wiley $3^{\text {rd }}$ edition
[RD14] "The 12C-bus specification version 2.1 of January 2000", Philips semiconductors, http://www.semiconductors.philips.com/i2c
[RD15] Spacecraft Dynamics \& control - A practical engineering approach, Marcel J. Sidi, Cambridge University Press, 2002
[RD16] "Rational Radial Distortion Models with Analytical Undistortion Formulae", Lili Ma \& YangQuan Chen \& Kevin L. Moore, IEEE CSOIS, Utah State University, 2003

## A. SPACECRAFT <br> LAYOUT <br> AND <br> ROTATION CONVENTIONS

Prior to describing the spacecraft attitude determination method, one needs to define the conventions used throughout this thesis. This section shows which components are currently set to be part of the DipTE layout and defines their reference frames. The basic rotations conventions are also stated for clarity.

## A.1. Frames \& hardware



Figure 39: Illustration of DipTE spacecraft-based coordinate systems pertaining to the spacecraft configuration and subsystems

Figure 39 shows an overall drawing of the layout from a device stand point. The spacecraft-based coordinate systems of each item are defined in the next sections.

## A.1.i. Geometric Fixed Coordinate System (GF)

The GF is the coordinate system with respect to which all the origins of the other spacecraft-based coordinate systems are defined and measured. The origin of the GF is placed at the reference (fiduciary) retro-reflector cube.

Origin: At the spacecraft reference marker (which will be located at one corner of the spacecraft).
Ox: Parallel to the (geometric) centerline of the 3 U cubesat. Positive direction points towards the direction of flight of DipTE (opposite from the aero panels.)
Oy: $\quad$ Normal to the Ox axis. Positive direction points in the same direction than the reference velocity vector.
Oz: Completes the RH coordinate system.

## A.1.ii. Spacecraft Body Fixed Coordinate System (BF)

The origin of the $B F$ is at the center of mass (CoM) of the spacecraft. As a consequence the origin of the BF will move during the mission as propellant is deleted from the propellant tank.

Origin: At the spacecraft center of mass (CoM)
Ox: Parallel to the Ox axis of the GF system.
Oy: Parallel to the Oy axis of the GF system.
Oz: Completes the RH coordinate system.

## A.1.iii. Payload Coordinate System (PL)

The payload is the wind and temperature spectrometer (WATS). The direction and magnitude of the wind relative to the spacecraft is measured with respect to the PL coordinate system.

Origin: As defined by the instrument maker and as determined after integration of the payload in the cubesat.

Ox: As defined by the instrument maker.
Oy: As defined by the instrument maker.
Oz: Completes the RH coordinate system.

## A.1.iv. Reaction Control System Cluster Coordinate System (RCSC)

The thrusters of the reaction control system are grouped by threes in reaction control clusters (RCSC). The thrusters provide forces and used in opposite pairs to generate attitude control torques.

RCSC\#1 is installed at the $\left(+x_{G F}, y_{G F},-z_{G F}\right)$ corner of the spacecraft.
RCSC\#2 is installed at the ( $+x_{G F},-y_{G F}+z_{G F}$ ) corner of the spacecraft.
RCSC\#3 is installed at the ( $-\mathrm{X}_{\mathrm{GF}},-\mathrm{y}_{\mathrm{GF}},-\mathrm{Z}_{\mathrm{GF}}$ ) corner of the spacecraft.
RCSC\#4 is installed at the $\left(-x_{G F},+y_{G F}, \mathbf{z}_{G F}\right)$ corner of the spacecraft.

Origin: Specific to each cluster - at the body corners
Ox: In the positive $\mathrm{Ox}_{\mathrm{GF}}$ direction.
Oy: In the positive $\mathrm{Oy}_{\mathrm{GF}}$ direction.
Oz: Completes the RH coordinate system.

## A.1.v. Orbital Maneuvering Thruster Coordinate System (OMT)

The orbital maneuvering thruster provides the thrust required for orbital maneuvers.

Origin: Geometric middle of the $-x$ (back) panel of the spacecraft.
Ox: In the direction of applied thrust, hence positive in the $+x_{G F}$ direction.
Oy: towards the aeropanel which has the GPS antenna
Oz: $\quad$ Completes the RH coordinate system.

## A.1.vi. Sun sensors SUNi



Figure 40: Sun sensor's structure frame
One attitude determination unit (ADU) is a sun tracker that is made of four cameras (not shown in Figure 39). Therefore the sun can be seen in four different fixed directions. By combining the different cameras, one can achieve a field of view greater than 180 degrees. The sun sensor retrieves a vector direction with respect to its own frame.

Origin: Center of the Sun sensors structure
Ox Completes the RH coordinate system
Oy: From origin to the middle of sensor 4 lens
Oz: Along the median of the (SUN2,SUN3) angle, from sensors to center of structure.

## A.1.vii. Magnetometer

One attitude determination unit (ADU) is a magnetometer that provides the direction of the earth magnetic field

Origin As defined by the instrument maker and as determined after integration of the payload in the cubesat

Ox: As defined by the instrument maker (see Figure 34).
Oy: As defined by the instrument maker.
Oz : Completes the RH coordinate system.

## A.1.viii. Earth sensor

The last attitude determination unit (ADU) is an earth sensor that retrieves a vector going through the origin of the spacecraft to the center of the earth: the nadir vector of the cubesat. It is not shown in Figure 39 but will roughfly be located at the same position than the SUN structure.

Origin: Center of the optical lens.
Ox: Same direction than the $\mathrm{x}+$ of the retrieved picture
Oy: Completes the RH coordinate system
$\mathrm{Oz}: \quad$ Optical axis of the sensor pointing behind the camera- IDEALLY passes through spacecraft CoM.

## A.2. Orientation and mathematical model

All measurement instruments are not located at the spacecraft's CoM. Therefore, a way to express the sensors' data in the body frame is needed. This section states the different conventions used to describe the orientation of one device with respect to another using the Euler angles and rotation sequences.

One needs to note that the different relative position would have to be physically measured at the end of the design so the algorithms can account for any misalignment due to the manufacturing process.

## A.2.i. Rotations and the Euler Angle Sequence

The Euler angles are denoted by the Greek symbol $\emptyset$. The rotation sequence for the DipTE mission is the 3-2-1 sequence. The rotation sequence transforms the original coordinate system Oxyz in the coordinate system OXYZ.

1. The first rotation with angle $\emptyset_{3}$, is about the Oz axis. The resulting coordinate system is denoted $O x^{\prime} y^{\prime} z^{\prime}$.
2. The second rotation, with angle $\emptyset_{2}$, is about the Oy' axis. The resulting coordinate system is denoted $O x^{\prime \prime} y^{\prime \prime} z^{\prime \prime}$.
3. The third rotation, with angle $\emptyset_{1}$, is about the $O x$ " axis. The resulting coordinate system is denoted OXYZ.

The 3-2-1 Euler angles rotation sequence is presented in Figure 41.


Figure 41: Illustration of the 3-2-1 Euler angles rotation sequence.

## A.2.ii. Angle of Attack and Sideslip Angle

The angle of attack ( $\alpha$ ) is defined as the angle between the projection of the velocity vector on the Oxz plane of the body fixed coordinate system (BF) and the Ox axis of the BF coordinate system. The positive direction is given by the right hand rule ( Oz over $O x$ ). The sideslip angle ( $\beta$ ) is defined as the angle between the projection of the velocity vector on the Oxy plane of the BF coordinate system and the velocity vector. The positive direction is given by the right hand rule (Ox over Oy). The angle of attack and sideslip angles are illustrated in Figure 42.


Figure 42: Illustration of the angle of attack and of the sideslip angle.

## A.2.iii. Setup

With the Euler rotations defined, the rotation matrices are

$$
\begin{aligned}
& {\left[R_{z}\right]=\left[\begin{array}{ccc}
\cos (\psi) & \sin (\psi) & 0 \\
-\sin (\psi) & \cos (\psi) & 0 \\
0 & 0 & 1
\end{array}\right]} \\
& {\left[R_{y}\right]=\left[\begin{array}{ccc}
\cos (\theta) & 0 & -\sin (\theta) \\
0 & 1 & 0 \\
\sin (\theta) & 0 & \cos (\theta)
\end{array}\right]} \\
& {\left[R_{x}\right]=\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & \cos (\phi) & \sin (\phi) \\
0 & -\sin (\phi) & \cos (\phi)
\end{array}\right]} \\
& \phi=\text { roll angle } \\
& \theta=\text { pitch angle } \\
& \psi=\text { yaw angle }
\end{aligned}
$$

Which leads to the general Euler $(3,2,1)$ rotation matrix to go from any item frame as described in A. 1 to the body frame

$$
R=\left[R_{x}^{\phi}\right] \times\left[R_{y}^{\theta}\right] \times\left[R_{z}^{\psi}\right]
$$

Thus the frame transformation is described by the following relationship

$$
\left[\begin{array}{c}
X \\
Y \\
Z
\end{array}\right]_{\text {bodyframe }}=[R] \times\left(\begin{array}{l}
x \\
y \\
z
\end{array}\right)_{\text {utemframe }}
$$

The R matrix is the direction cosine matrix (also called attitude matrix) that will be referred to as A from now on.

## B. LENS CORRECTION ALGORITHM

The sensor used throughout this thesis is a webcam described in 3.1.2.1. By using this sensor with the manufacturer's lens (which as a focal length $f=3.85 \mathrm{~mm}$ ) it as been noticed that the picture is deflected so that measures in the real world are not properly retrieved by the program.


Figure 43: Main lens distortion types

Lens deflections are of two main types: the barrel distortion and the pincushion distortion. The combination of the two is non-linear. These can be seen on Figure 43. More exotic lenses can have a spatially dependent combination of the two main shapes which can be hard to correct.

A picture is formed on a focal plane as a copy of visible part of an object in the object plane as seen in Figure 44. Image distortion can be mathematically expressed with the lens mapping $L(r)$. This lens mapping models the relationship between distances in the object plane and distances in the focal plane.

Let the focal plane and the object plane be parallel to each other and let the centers of these planes intersect the opticaj axis. For a dot, located in the object plane at a
distance " r " from the center, the corresponding image in the focal plane is a dot located at a distance $L(r)$ from the center of this plane. Therefore, a lens with no distortion would have $L(r)=r$. As described, in a 2-dimensional space, $r=\sqrt{x^{2}+y^{2}}$.


Figure 44: Relation between distances in the object plane and distances in the focal plane

Let the transverse magnification be described as $M=\frac{d L}{d r}$. This quantity permits expressing the main distortions as:

- Barrel distortion: happens for $\frac{d M}{d r}<0$. The magnification decreases the further away the dot goes from the center
- Pincushion distortion: happens for $\frac{d M}{d r}>0$, which is opposite of the barrel distortion

To correct for a specific distortion, we introduce the polynomial radial undistortion function $F^{-1}=L$ defined as

$$
F(r)=r * p(r) \text { where } p(r)=1+\sum_{i} k_{i} r^{2 i}
$$

and the $k_{l}$ are the distortion coefficients for a specific lens. It has been found in the literature (see [RD16]) that the distortion is dominated by the first term of $F(r)$ and
using too many high terms may result in numerical instability. Having $p(r)$ of order 4 provides good accuracy. If $X_{d}=\left[\begin{array}{l}x_{d} \\ y_{d} \\ z_{d}\end{array}\right]$ is the distorted coordinates of a point in the focal plane and $X=\left[\begin{array}{l}x \\ y \\ z\end{array}\right]$ the coordinates of the corrected point, we obtain

$$
X_{d}-X_{0}=p(r) *\left(X-X_{0}\right)
$$

Therefore, solving for the corrected coordinates of the point requires finding the inverse of the function $p(r)$. It is hard to do it analytically but easily done by a numerically iterative process.

The mathematical model described above is not the actual model used in the sensors' design. A lot more time would be needed to develop such a tool. Therefore, the actual correction process used is the "Matlab Calibration Toolbox" was written by Dr Jean Yves Bouguet from the California Institute of technology. This toolbox includes both a calibration code to compute the intrinsic parameters of the lens and the codes to correct the data.
The mathematical model of this toolbox is explained in details on the following webpage http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html The calibration process that was used is explained step by step at the following internet address http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html.
The author strongly recommend to visit these webpages that show precisely the correction method

## C. GNC ATTITUDE DETERMINATION TEST ROTATION PLATFORM

The GNC algorithms of the spacecraft could not be tested on a full scale model. Therefore the attitude determination part of the GNC package could be tested on ground by building a test bench. This bench has been called gimbal device or rotation platform in this thesis .


Figure 45: Attitude determination test bench, Catia model (left) and actual hardware (right)

This test bench is a tool designed so we can rotate the sun sensor and the magnetometer on the $x, y$ and $z$ axis accurately. The design of this tool had to permit the user to set a rotation angle on each axis precisely. It has been manufactured by the ERAU manufacturing lab under the direction of William Russo.

## C.1. Test bench frames

To be able to check if the algorithms are reliable, on needs to test them with a new reference. For the test performed, the gimbaled platform represents the spacecraft where:

Origin: At the geometric center of the compass support
Ox:Along the axis of rotation of the instrument support toward the worm gear

Oy. Along the main rod, toward the worm gear
Oz : Completes the RH coordinate system.


Figure 46. Bench frame

The top rotation platform has its own frame defined set as frame 2 such that:
Origin- At the geometric center of the top rotation plate
Ox2: Along the axis of rotation of the instrument support toward the worm
gear
Oy2 Completes the RH coordınate system
Oz 2 Goes toward the bottom part of the plate

The $U$ shape part of the rotation platform has its own frame defined set as frame 1 as:
Origin: At the geometric center of the top rotation plate
Ox1: Along the axis of rotation of the instrument support toward the worm gear

Oy1: Completes the RH coordinate system.
Oz1: Goes toward the main rod

Recall the instruments frames need to be aligned the following way.


Figure 47: Instruments frame for test platform

Due to the location of the instruments, the rotation provided by this device is not pure. Therefore corrections are needed. Let $X s u n=\left(x_{s}, y_{s}, z_{s}\right)$ be the coordinate of a vector expressed in the SUN frame. One is interested in expressing this vector in the GIM frame. Let Frame 2 be attached to the instrument, Frame 1 be attached to the instrument support $U$ shape and Frame $0 \equiv G I M$ be attached to the rotating plate as it can be seen on Figure 46 and Figure 48.

Using these assumptions, a point $S$ from the SUN frame complies

$$
\begin{gathered}
S_{C C D C e n t e r}^{S U N} \equiv\left(\begin{array}{lll}
x_{s} & y_{s} & z_{s}
\end{array}\right)_{X s u n, Y s u n, Z s u n} \\
S_{B}^{G I M} \equiv\left(\begin{array}{lll}
x_{0} & y_{0} & z_{0}
\end{array}\right)_{X g i m, Y g i m, Z g i m}
\end{gathered}
$$

$$
\begin{aligned}
& \left(\begin{array}{l}
x_{0} \\
y_{0} \\
z_{0}
\end{array}\right)_{\text {body }} \\
& \equiv\left[\begin{array}{ccc}
\cos (\psi) & \sin (\psi) & 0 \\
-\sin (\psi) & \cos (\psi) & 0 \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{ccc}
-\cos \left(\phi^{\prime}\right) & 0 & -\sin \left(\phi^{\prime}\right) \\
0 & 1 & 0 \\
\sin \left(\phi^{\prime}\right) & 0 & -\cos \left(\phi^{\prime}\right)
\end{array}\right]\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & \cos \left(\theta^{\prime}\right) & \sin \left(\theta^{\prime}\right) \\
0 & -\sin \left(\theta^{\prime}\right) & \cos \left(\theta^{\prime}\right)
\end{array}\right]\left(\begin{array}{l}
x_{s} \\
y_{s} \\
z_{s}
\end{array}\right)_{\operatorname{SUN}}
\end{aligned}
$$

Where $\theta^{\prime}=-\operatorname{atan}\left(\frac{d}{L}\right)$ the roll angle and $\phi^{\prime}=0$ for initial pitch angle

For the magnetometer, $L=1.5 \mathrm{~cm}$ and $d=2 \mathrm{~cm}$. For the sun sensor $L=2 \mathrm{~cm} d=-1.5 \mathrm{~cm}$.


Figure 48: Vector transformation for the rotation device measurements

## C.2. Test bench magnetic field

The gimbal design has its own magnetic field. Measurements have been taken showing that the magnetic field has a displacement of 6 degrees with respect to the environment field. This was computed after putting the magnetometer on the device.


Figure 49: Magnetic field deviation due to the rotation device.

A demagnetizing process has been performed on the steel parts of the device but did not seem to decrease the shifting effect. Aluminum is the main material. Magnetized parts are still shifting the magnetic field.

## D. TEST PROCEDURE OF ATTITUDE DETERMINATION ALGORITHM AT ERAU DAYTONA BEACH CAMPUS

## D.1. Setup

The device needs to be aligned with local magnetic field. This is done using the compass located in the middle.

Due to the expect $100^{\circ}$ FOV of the sun sensor with the designed lens, the sun sensor would not be able to detect the sun for rotation of more than $50^{\circ}$ around Xgim and Ygim , if the sun is assumed to be at the zenith with respect to the device.

The magnetometer sensor and the sun sensor frames are rotated compared to the gimbal frame. The parameters are the following

$$
\begin{gathered}
\varphi=\text { yaw }=0 \\
\theta=\text { roll }=-180 \mathrm{deg} \\
\phi=\text { pitch }=0
\end{gathered}
$$

By plugging them into the transformation equation $\left[\begin{array}{c}X \\ Y \\ Z\end{array}\right]_{\text {bodyrame }}=[R] \times\left(\begin{array}{c}x \\ y \\ z\end{array}\right)_{\text {nemframe }}$,
they can permit to express a vector in a sensor's frame to the body frame.

## D.2. Natural references

Two references are needed: the sun position and the magnetic field direction.
In Daytona Beach, the magnetic field has a declination of $5^{\circ} 53^{\prime} \mathrm{W}$ changing by $0^{\circ}$ $4^{\prime}$ W/year. This mean the magnetic field Xvector makes a $5^{\circ} 53^{\prime} \mathrm{W}$ angle with the geometric north.


Figure 50: Magnetic field direction in Daytona Beach,FI

## E. SUN SENSOR TESTS

Tests for non moving dot :

## Accuracy on a non moving dot tested with lighter



Figure 51: Precision of the sun sensor by using a lighter

$$
\sigma_{E L F}=0.15439
$$

Figure 52 Precision of a fixed dot tested with a flashlight

## Non moving dot Laser Pointer

135

$$
\begin{gathered}
\sigma_{A Z I}=0.067 \\
\sigma_{E L E}=0.0368
\end{gathered}
$$

1)     * 

$$
\sigma_{A 7 t}=016212
$$



Figure 53 Precision of fixed, dot using a laser pointer

## Displacement of lighter $\mathbf{1 m}$ in $X$ and $Y$ on board located at 3 meters



Figure 54: Displacement test between two dots separated by 18.4 degree in azimuth and elevatior with the lighter

## Flashlight displacement

$\mathrm{P} 1(4 ; 3.183)$

$$
\sigma_{A Z I}=0.1621
$$

$$
\sigma_{E L E}=0.154
$$

$$
0 \cdots
$$

$$
\$
$$

$-2$

P2(-9.069;-7.377)

$$
\sigma_{A Z I}=0.021
$$

$$
\sigma_{E L E}=0.023
$$

Aximuthin degrees ${ }^{-8}$

Figure 55: Displacement test between two points with flashlight

## Laser Pointer displacement test



Figure 56: Displacement test with a laser pointer Sun sensor Spatial distribution with designed lens
 - PO

AZIMUTH in dearees
Figure 57: Sun sensor test with three dots, laser pointer

## F. ATTITUDE DETERMINATION MATLAB CODE

The code included in this section represents the first attitude determination code of the DipTE spacecraft. The general structure of this code can be seen on the following picture.


```
###################################################################
        TRIAD Algorithm main file
        DipTE spacecraft project
    -- Notes --
Ref. Three-axis attitude determination from vector observations, Shuster
    Spacecraft attitude determination and control, Wertz
    A good brann, Udrea
##################################################################
                Mathreu Naslin
                        Summer 2009
################################################################
clear all
close all
clc
                        mntialisation
%
```



```
%defines the Centerofmass according to the Geometric fixed coordinate system (GF)
x_COM = 1;
y_COM = 1;
z_COM = 1;
%defines the Sun sensor center according to the Geometric fixed coordinate system
(GF)
x_SUN = 1;
Y_SUN = 1;
z_SUN = 1;
D_COMtOSUN = [x_SUN - x_COM;Y_SUN - y_COM;z_SUN - z_COM]; % distance from the COM
(vector ComSun)
Sdefines the MAG sensor center according to the Geometric fixed coordinate system
(GF)
x_MAG = 1;
Y_MAG = 1;
z_MAG = 1;
D_COMTOMAG = [x_MAG - x_COM;Y_MAG - y_COM;z_MAG - z_COM]; 咅 distance from the COM
# go from sensors frame to body frame for the measured vectors
%-..- -..- .-.. -.......... SUN
theta3_SUN = 0 * pi/I80; %sets the deviation between the suN frame and BF in
euler's definition
theta2_SUN = -180 * pi/180; % " "
thetal_SUN = 0 * pI/180; " " "
R3=[[cos(theta3_SUN),sin(theta3_SUN),0]; [-
sin(theta3_SUN),cos(theta3_suN),0];[0,0,1]];
R2=[[cos(theta2_SUN),0,-
sin(theta2_SUN)];[0,1,0];[sin(theta2_SuN),0,\operatorname{cos(theta2_SUN)]];}
R1=[[1,0,0];[0,cos(theta1_SUN),sin(theta1_SUN)];[0.-
sin(thetal_suN), cos(theta\__suN)]);
R_SUN_to_BF}=R1*R2*R3; - %express rotation matrix to go frome the SUn sensor
frame to the body frame
```



```
theta3_MAG = 0 * pi/180; %sets the deviation between the suN frame and BF in euler's
definition
theta2_MAG = -180 * pi/180; % "
thetal_MAG = 0 * pi/180; 多 "
R3=[[cos(theta3_MAG),sin(theta3_MAG),0]; [-
sin(theta3_MAG),cos(theta3_MAG),0];[0,0,1]];
R2= ([cos(theta2_MAG),0,-
sin(theta2_MAG)];[0,1,0];[sin(theta2_MAG),0,cos(theta2_mAG)]];
R1=[[1,0,0]; [0, cos(theta1_MAG),sin(thetal_MAG)],[0,-
sin(thetal_MAG),\operatorname{cos(thetal_MAG)]];}
R_MAG_to_BF = R1*R2*R3; %express rotation matrix to go frome the MAG sensor
frame to the body frame
%%% Orbit data
inclination_orbit = 70; %orbit inclination in degrees
while(1)
    %%
    % the observation unmt vectors are (Wsun,Wmagneto) & the reference unat
vectors are (V1, V2)
    [longitude_SC,latitude_SC,altatude_SC,X_ECF,Y_ECF,Z_ECF] =
location_from_GPS %get the current position of the spacecraft
expressed in ECF (angles are in deg)
    orientation = IMU_current_orientation
%get the current orientation of the spacecraft pitch roll yaw using the IMU
    [V1_1nert,V2_inert] =
reference_vectors_initialisation(altitude_SC,Inclination_orbit,latitude_SC)
%computes the reference vectors VI_inert and V2_ inert in the inertial frame
    %% computation of the measured vectors
    Wsun_measured = sun_sensor_main' Fcomputes the current
UNIT sun direction detected wath the sun sensor IN SUN FRAME
    Wmagneto_measured = 10e-4 * magnetometer %computes the current
magnetic field direction detected with the magnetometer IN MAG FRAME in TESLA
\%\% computation of the vectors in the spacecraft body frame
    %..-. -..-...-...-...-...- transformations
    Wsun_BF = R_SUN_to_BF * Wsun_measured; %sun vector measured by
the sensor expressed in the RF
    Wmagneto_BF= R_MAG_to_BF * Wmagneto_measured; %Magnetic field vector
measured by the sensor expressed in the RF
    *Test if the measurement vectors are colinear
if cross(Wsun_measured,Wmagneto measured) == [0;0;0]
    fprintf('Measurement vectors are collnear ! -- ERROR');
end
```

```
    %% TRIAD ALGORITHM
```



```
    % Computation of the attitude matrix
    % prior computations
    #|||||||||MAKE SURE THAT THE VECTORS RETRIEVED ARE GOING TO BE
OF MAGNITUDE 1 |'|!|!|!1!1!"!
    elev_SuN = atan(Wsun_BF(3)), %elevation of the SUN
vector in radians
    azIm_SUN = atan(Wsun_BF(2)/Wsun_BF(1)); %azimut of the SUN vector
in radians
    elev_MAG = atan(Wmagneto_BF(3)), %elevation of the MAG
vector in radians
    azım_MAG = atan(Wmagneto_BF(2)/Wmagneto_BF(1)), %azımut of the MAG vector
in radians
    % Computation of the attitude matrix A
    rl = V1_inert;
    r2 = cross(V1_1nert,V2_1nert)/norm(cross(V1_1nert,V2_1nert));
    r3 =
cross(V1_1nert,cross(V1_1nert,V2_1nert))/norm(cross(V1_1nert,V2_1nert)),
    sI = Wsun BF, 产first vector defined as the sun vector (CHECK IF IT
IS REALLY THE MOST PRECISE VECTOR)
    s2 = cross(Wsun_BF,Wmagneto_BF)/norm(cross(Wsun_BF,Wmagneto_BF));
    s3 =
cross(Wsun_BF,cross(Wsun_BF,Wmagneto_BF))/norm(cross(Wsun_BF,Wmagneto_BF)),
    Mref = [r1,r2,r3],
    Mobs = [s1,s2,s3].
    A = Mobs * Mref' 言 attitude matrix discarding error on the
measurements
    %Outputs the attitude of the spacecraft in pitch, roll and yaw angles
    grecall we are ASSUMING A is also using a 3-2-1 Euler sequence TO BE
CHECKED TO BE CHECKEDTO BE CHECKEDTO BE CHECKEDTO BE CHECKEDTO BE CHECKEDTO BE
CHECKEDTO BE CHECKED
    % as described in "Elements of spacecraft design" By Charles D Brown
Page 276
    yaw = atan( A(1,2)/A(1,1) );
    pltch = atan( -A(1,3)/sqrt(1-A(1,3)^2)),
    roll = atan(A(2,3)/A(3,3) ),
    fprintf('\n\n\tusing a 3-2-1 euler sequence, the attitude of the
spacecraft is found to be (n').
            fprintf('\t\t\t pitch angle (deg) \t%5 5f\n',pitch*180/pi).
            fprintf('\t\tlt Roll angle (deg) \t⿱⿻⿴囗丨丷日丿十⿱⿰㇒一大口
            fprintf('\t\t\t Yaw angle (deg) (t%5 5f\n\n\n\n',yaw*180/p1);
                            fprintf('\n\n\tUsing a 1-3-1 euler sequence, the attitude of the
spacecraft is found to be (n');
                            fprintf('\t\t\t Patch angle 131 (deg) \t:5 5f\n', atan2( A(1,3),A(1, 2)
)*180/p1);
            fprintf('\t\t\t Roll angle 131 (deg) \t%5 5f\n',acos( A(1,1) )*180/p1),
                            fprintf('\t\t\t Yaw angle 131 (deg) \t%5 5f\n\n\n\n',atan2( A(3,1), -
A(2,1) )*180/p1),
    #SC_orzentation (A,V1_Inert, V2_Inert, Wsun_measured, Wmagneto measured)
    pause(5*60), %aigorithm i run every 5 minutes
```

end

```
function [longitude, latitude,altitude,X_ecf, Y_ecf, Z_ecf] = location_from_gPS
% ##################################################################
% Retrieves the coordinates of the spacecraft given by the GPS
%
%In : None
%Out : Longitude (deg), Latitude (deg), X position, Y position and Z position of
% spacecraft in the ECF
%
% -- Notes --
& Ref. - http://ocw.mit.edu/NR/rdonlyres/Earth--Atmospheric--and-Planetary-
Sciences/12-540Spring-2008/LectureNotes/12_540_lec04 pdf
% Slide 21
% - also page 24 of Navigation: principles of positioning and guidance By
Bernhard Hofmann-Wellemhof
%
% ###############################################################
% Mathieu Naslin
% Fall 2009
##################################################################
```

```
Gdefinution of the WGS84 Ellapsoid
```

Gdefinution of the WGS84 Ellapsoid
a = 6378137; %in meters
a = 6378137; %in meters
b}=6356752.314;%1n meters
b}=6356752.314;%1n meters
f=(a-b)/a;
f=(a-b)/a;
?data from the hubble telescope position
?data from the hubble telescope position
% latitude = 26.6,}\quad\mathrm{ %Latitude in
% latitude = 26.6,}\quad\mathrm{ %Latitude in
degrees + 1s north - is south
degrees + 1s north - is south
% longitude =-123.7, 咅Longitude in
% longitude =-123.7, 咅Longitude in
degrees + is east - Is west
degrees + is east - Is west

* altitude = 561000, %altitude in
* altitude = 561000, %altitude in
meters
meters
latitude = 29.284924; %Latitude in
latitude = 29.284924; %Latitude in
degrees + is north - is south
degrees + is north - is south
longitude = -81.102968; 告ongitude in
longitude = -81.102968; 告ongitude in
degrees + 1s east - Is west
degrees + 1s east - Is west
altitude = 3; %altitude in
altitude = 3; %altitude in
meters
latitude = latitude * pi/180; longitude = longitude * pi/180; * conversion to
radians
N = a^2 / sqre(a^2 * cos(latitude)* 2 + b^2 * sin(latitude)^2);
%geodetic coordinates of the spacecraft
X ecf = (N + altitude) * cos(latitude) * cos(longitude);
Y_ecf = (N + altitude) * cos(latitude) * sin(longitude);
z_ecf = (b^2/a^2 * N + altitude) * sin(latitude);
altitude_sC = sqrt (X_ecf^2+Y_ecf^2+Z_ecf^2}2)/1000-6378.137

```
```

latıtude = latıtude * 180/p1; longıtude = longıtude * 180/pi; % conversion to
degrees

```
```

function [V1_inert,V2_inert] =
reference_vectors_initialisation(altitude,inclination_orbit,latitude,longitude)

```

```

% Initialization of the reference vectors in the inertial geocentric frame
% DipTE spacecraft project
%In none
soutput
V V1_inert 曾 reference direction of the sun in the inertial frame
Inertial frame in TESLA

# 

? - Notes --
The computation is based on the current time
\#\#\#\#\#\#\#\#\#\#\#\#揞\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#
Mathıeu Naslın
Summer 2009
\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#

```

읗웅
```

% Sun vector computation vi_inert (as explained in RD3)

```

```

ctime = clock, % is made of lyear month day hour minute
seconds]
JD = 367 * ctime(1) - Eloor((7*(ctime(1) +floor((ctime(2)+9)/12)))/4) + floor
(275*ctime(2)/9) + ctime(3) + 1721013 5 + ((ctime(6)/60+ctime(5))/60+ctime(4))/24;
%computes the number of Julian day
Tut = (JD-2451545 0)/36525, %computes the Julian century
LamdaM = mod(280 460 + 36000 771*Tut,360), fmean longitude of the sun
M = mod(357 5277233 + 35999 05034*Tut,360), %mean anomaly of the sun
Lambda_eclıptic = mod(LamdaM + 1 914666471 * sin(M*pı/180) + 0.019994643 *
sin(2*M*p1/180),180) % PROBLEM THERE - maybe correected now How can I check this
value *
epsilon = 23 439291-0.0130042 * Tut, %obliquity of the ecliptic
norm_R = 1000140612 - 0.016708617 * cos(M*p1/180) - 0 000139589* cos(2*M*p1/180);
% fprintf('Julian day = %10 10d\n',JD)
% fprintf('Julzan Century = %10 10d\n',Tut)
% fprintf('Lambda M= %10 10d\n',LamdaM)
% fprintf('M= %10 10d\n',M)
% fprintf('Lambda ecliptic= %10 100\n',Lambda_ecliptic)
% fprintf('epsilon = %10 10d\n',epsilon)
% fprintf('norm of vector r = %10 10d AU\n',normR)

```
```

V1_in_AU = norm_R * (cos(Lambda_ecliptic*pi/180) ;
cos(epsilon*pi/180)*sin(Lambda_ecliptic*pi/180) ;
sin(epsilon*pi/180)*sin(Lambda_ecliptic*pi/180)]; %sun vecto in AU
V1_inert(1) = V1_in_AU(1)/sqrt(V1_in_AU(1)^2 + V1_in_AU(2)^2 + V1_in_AU(3)*2);
V1_inert(2) = V1_in_AU(2)/sqrt(V1_in_AU(1)^2 + V1_in_AU(2)^2 + V1_in_AU(3)^2);
V1_inert(3) = V1_in_AU(3)/sqrt(V1_in_AU(1)^2 + V1_in_AU(2)^2 + V1_in_AU(3)^2);

```

\%\%

\% Magnetometer vector computation v2_inert

\({ }^{*} B=\) igrf_DipTE (altitude, inclination_orbit, latitude) \% get the magnetic field that
we are suppose to see in the current position
\([X Y Z, H, D E C, D I P, F]=\) wrldmagm(altitude* \(10^{*} 3\), latitude, longitude, decyear(clock));

TESLA
```

function pointing_vector = sun_sensor_main;
| \#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#

* sun sensor simulator
%
This codes permits to simulate from A to Z the sun sensing process.
It:
% 1) Creates a picture of the satellite field of view
% 2) Run an algorithm to find the sun on the snapshot and retrieve
* its characteristics
% 3) Compute the vector pointing on the sun expressend in the
% sensor's frame
%
% -. Notes --
Ref . Three-axis attitude determination from vector observations, Shuster
Spacecraft attitude determination and control, Wertz
A good brain, B. Udrea
\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#
Mathieu Naslin
Summer 2009
\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#
initialisation

```
vid \(=\) videoinput ('winvideo', 1); Fopen webcam preview
\%preview(vid) ;
date_of_test = date;
```

dist_CCD_pinhole = 3.85;
is 2.83mm -> }1288\mathrm{ pix)
dist_CCD_pinhole = dist_CCD_pinhole *1288 /2.83; %distance expressed in pixels
dist_CCD_pinhole = 1258.34466;

```

```

% acquisition
%
Cloud = getsnapshot(vid); %get a snapshot from the webcam
Cloud = rgb2gray(Cloud); %transform the picture for computation from RGB to gray
colors
%im_for_Hough_correction = Cloud;
%circles = houghcircles(im_for_Hough_correction, 1,4,0.5,4);
[rows,cols] = size(Cloud); 咅size of the picture
reso_width = cols;
reso_height = rows;
% --- black and white
for i = 1:rows
for j = 1:cols
if Cloud(i,j) <= 125
Cloud(i,j) = 0; % black dotA
else
Cloud(i,j) = 255;
end
end
end

```


```

%

```
%
computation
```

computation

```


```

% ***************** Detect the center of the sun disk ******************

```
% ***************** Detect the center of the sun disk ******************
```

% ***************** Detect the center of the sun disk ******************
x = ones(rows,1)*[1:cols]; 举 Matrix with each pixel set to its }
x = ones(rows,1)*[1:cols]; 举 Matrix with each pixel set to its }
x = ones(rows,1)*[1:cols]; 举 Matrix with each pixel set to its }
coordinate
coordinate
coordinate
y = [1:rows]:*ones(1,cols); 产 Matrix with each pixel set to its y
y = [1:rows]:*ones(1,cols); 产 Matrix with each pixel set to its y
y = [1:rows]:*ones(1,cols); 产 Matrix with each pixel set to its y
coordinate
coordinate
coordinate
area = sum(sum(Cloud));
area = sum(sum(Cloud));
area = sum(sum(Cloud));
X_sun_center = sum(sum(double(Cloud).*x))/area; % x coordinate of the center of the
X_sun_center = sum(sum(double(Cloud).*x))/area; % x coordinate of the center of the
X_sun_center = sum(sum(double(Cloud).*x))/area; % x coordinate of the center of the
disc
disc
disc
Y_sun_center = sum(sum(double(Cloud).*y))/area; 类 y coordinate of the center of the
Y_sun_center = sum(sum(double(Cloud).*y))/area; 类 y coordinate of the center of the
Y_sun_center = sum(sum(double(Cloud).*y))/area; 类 y coordinate of the center of the
disc
disc
disc
%Correction of the horizon point for lens deflection
edges = [X_sun_center;Y_sun_center];
[Xedges_corrected,Yedges_corrected] = distortion_correction_1point(edges);
x_coord = (Xedges_corrected - reso_width /2);
y_coord = (Yedges_corrected - reso_height /2);

```
```

azi= atan(x_coord/dist CCD pinhole); %azimuth in radians
elev= atan(y_coord/dist_CCD_pınhole)* cos(azi); %elevation in radians
%Computation of the sun vector using elevation and azamuth NORMALIZED
r = 1;
x = - r * cos(elev) * sin(azi);
y=r * cos(elev) * cos(azi);
z = - r * cos(azi);
pointing_vector = [x,y,z];
pointing_vector = pointing_vector/norm(polnting_vector); %output the unit pointing
vector
gclosing the device after each snapshot
品closepreview(vid);
stop(vid); delete(vid);
Eunction mag_vect = magnetometer;

```

```

%}\mathrm{ magnetometer aquisition subroutine
呂
Thys codes permits to get the magnetio Eleld from the MEMSense nIMU
device The magnetic field is in GAUSS
喽
%
% -- Notes --
Ref. MEMSense nIMU doc PSD-0822
%

```

```

% Mathzeu Naslin
% Summer 2009
\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#

```
```

\&
% In土tialisation

```

```

S =
serial('CoM4', 'BaudRate', 115200, 'Databits', 8, 'Parity', 'none','StopBits',1,'Timeout',5
00, 'Name', 'nIMU', 'BytesAvailableFcnMode','byte','InputBufferSize',38); 寽 Cxeate
serial port object
fopen(s): 丼 open the sexial port
%

* Aquisition

```

```

try
readasync(s); 若ask matlab to get data
asynchronously
pause(0.1);
%let the time for the nIMU to respond
output = fread(s,s.BytesAvallable,'int8'); Freads a complete data set THE
WORKING WAY

```

```

    nb var = size(output); \quad%get the saze of the packet (supposed
    to be 38 bytes)
catch

```
```

    fclose(s);
    %close the device for future use of
    the serial port
fprintf('ERROR reading the data from the nIMU - retry the operation !!\n\n')
return
end
we can work only if the synchronisation bytes are detected at the bigganing of the package
If ((output (1) == -1) \& (output (2) == -1) \& (output(3) == -1) \& (output (4) == -1))

```

```

                % Treatment of the data
                多 -------------------.---.------------------------------------
            % % data_from_sensor(1 to 4) synchronization bytes
            % 咅 data_from_sensor(13) last byte of the header
            % % data_from_sensor(26 to 30) Magnetometer data !!
                % combine the MSB and LSB values of each magnetic field component
                Mag_X_raw = bitshift(output(26),8) + output(27);
                Mag_Y_raw = bitshyft(output(28),8) + output (29);
                Mag_Z_raw = bitshift(output(30).8) + output(31);
                %using equation 1 to convert the sampled values
                Mag_X = Mag_X_raw * 8.6975*10* (-5);
                Mag_Y = Mag_Y_raw * 8.6975*10*(-5);
                Mag_Z = Mag_Z_raw * 8.6975*10*(-5);
    ```

```

                % Send back the magnetic field vector
                    % -------------------------------------------------------------
    mag_vect = [Mag_X;Mag_Y;Mag_Z]; %magnetic field in GAUSS
else %the data read don't have the proper format
fprintf('ERROR reading the data from the nIMU - retry the operation !!\n\n')
end

```

```

                                    Close the device
    %
delete(s);
clear s;

```
```

function [Xedges_corrected,Yedges_corrected] = distortion_correction(edges)

```
```

function [Xedges_corrected,Yedges_corrected] = distortion_correction(edges)

```


```

% Distortion correction

```
% Distortion correction
*Applying the theory as explained in
*Applying the theory as explained in
http.//www vision.caltech.edu/bouguety/calib_doc/htmls/parameters.html
http.//www vision.caltech.edu/bouguety/calib_doc/htmls/parameters.html
%
%
%
%
告 Thas codes permats to correct the distortion of a pacture taken with our weboam
告 Thas codes permats to correct the distortion of a pacture taken with our weboam
Made in chama
```

Made in chama

```
```

% It uses the CAMERA CALIBRATION TOOLBOX to carrect the camera distortion.
Theory and code by Jean-Yves Bouguet from CalTech
*
% -- Notes --
% Ref . http://www.vision.caltech.edu/bouguetj/calib_doc/
%
告 \#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#
Mathieu Naslin
% Fall 2009
% \#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#

```

```

% % Focal Length: fc = [ 1258.34466 1266.36745] \ [ 19.44581 20.00630 ]
% % Principal point: }\quadCC=[448.40357 359.22821] [ [ [ 25.24977 21.55412]
%% Skew: alpha_c = [0.00000] \ [0.00000 ] => angle of pixel axes =
90.00000 \pm0.00000 degrees
\#% Distortion: }\quad\textrm{kc}=[$$
\begin{array}{lllllll}{-0.36330}&{0.59953}&{0.00846}&{0.00326}&{0.00000}\end{array}
$$]

```

```

% % Pixel error: err = [ 1.36115 1.36984 ]
fc = [ 1258.34466 1266.36745 ]; cc = [ 448.40357 359.22821];
alpha_c = [0.00000 ];

```

```

];

```

\section*{}
```

distorded_edges = edges ;
奋The edges found before are not
corrected yet. They are distorded
% as [x1 x2 x3;y1 y2 y3]
$[x n]=$ normalize(distorded_edges, fc, cc,kc, alpha_c); Fnormalized distorded vector THIS ALREADY UNDO THE SKEN AND COMPENSATE FOR LENS DISTORTION

* as [x1 x2 x3;y1 y2 y3]

```
\([x, d]=\) comp_distortion_oulu (xn,kc);
\%Apply the distoxtion equations
to obtain \(x=(x d-\operatorname{delta} x)\)
\(K K=[\operatorname{fc}(1)\) alpha_c \(* \mathrm{fc}(1) \quad \mathrm{Cc}(1) ; \quad\) intrisic parameters of the
camera
\begin{tabular}{llc}
0 & \(\operatorname{fc}(2)\) & \(\operatorname{cc}(2) ;\) \\
0 & 0 & \(11 ;\)
\end{tabular}
\(X Y p 1=K K *\left[x_{-} d(1,1) ; x_{-} d(2,1) ; 1\right] ;\)
scorrect the position of the points using
the intrisic parameters of the camera

Xedges_corrected \(=[\) XYp1(1)];
Yedges_corrected \(=[x Y p 1(2)]\);```

