INSTITUTE OF CONTROL AND COMPUTATION ENGINEERING
FACULTY OF ELECTRONICS AND INFORMATION TECHNOLOGY
WARSAW UNIVERSITY OF TECHNOLOGY

MASTER OF SCIENCE THESIS

STAR-TRACKER PROGRAM FOR CUBESAT SATELLITES

Szymon MICHALSKI

Supervisor:
prof. dr hab. inż. Ryszard Romaniuk

Warszawa 2016

Abstract

Streszczenie

Contents

Nomenclature

$\boldsymbol{b}$	Known directional unit vector in the BODY frame
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{M}$	Least squares estimate of rotation matrix
$\boldsymbol{n}$	Unit vector
$\boldsymbol{Q}$	Quaternion matrix
$\boldsymbol{q}$	Unit quaternion
$\boldsymbol{q}_{vec}$	Vector part of unit quaternion
$\boldsymbol{r}$	Known directional unit vector in the NED frame
$\boldsymbol{R}(\cdot)$	Rotation matrix using Euler angles
$\boldsymbol{R}_n^b$	Rotation matrix representing a rotation from n to b
$\boldsymbol{S}(\cdot)$	Skew symmetric matrix
ϕ	Euler angle, roll
ψ	Euler angle, yaw
θ	Euler angle, pitch
q_0	Scalar part of unit quaternion
v	General Euler angle

[1] [2] [3] [4] [5] [6] [7] [8]

1 Introduction

1.1 Motivation

The goal of this work is to make fully operational star-tracker program, that could be used on Cubesat satellites. Such program could be used on space missions and could start Polish state-of-the-art technology in growing space technology sector.

1.2 Outline of thesis

This thesis consists of several chapters. Here they are shortly summarized:

Chapter 1 serves as introduction to this thesis and describest the motivation and goal of this work. It also describes the background of the topic.

Chapter 2 describes all the important foundations for the fully understanding given work.

Chapter 3 is the main part of this thesis. It describes how the star-tracker program works and goes through detailed comparison of different approaches.

Chapter 4 describes the created prototype of star-tracker in Python language.

Chapter 5 talks about the implementation of star-tracker on the existing prototype of on-board computer.

Chapter 6 describes how the finished program is performing.

Chapter 7 contains conclusons about this work and created star-tracker program.

1.3 Cubesat

Cubesat was designed on CalPoly in 1999[9]. Dimensions of satellite are measured in units. Each unit (often described simply as u) can be 10x10x10cm and can weight up to 1.33 kg. Satellites can be 1u, 2u, 3u, 6u or even 12u.

Such small satellites are suspectible to noise from densly packed electronics.

Zdjecie Cubesata

CubeSat missions, goals, what can they be and are used for? Why is it innovative and important?

1.4 Means of attitude estimation

There exist many different types of attitude estimation: sun sensors, star-trackers, magnetometers, etc. However star-tracker gives the best possible accuracy for nowadays and is not suspectible to electrical nor magnetic noise.

1.4.1 Megnetometers

1.4.2 Sun sensors

1.4.3 Earth sensors

1.4.4 GPS

1.4.5 Star trackers

[10] [12]

Sensor	**Accuracy**	**Characteristics and Applicability**
Magnetometers	1.0o (5000km alt) 5.0 (200 km alt)	Attitude measured relative to Earth's local magnetic field. Magnetic field uncertainties and variability dominate accuracy. Usable only below ≈6,000 km.
Earth sensors	0.05 (GEO) 0.1 (LEO)	Horizon uncertainties dominate accuracy. Highly accurate units use scanning.
Sun sensors	0.01	Typical field of view +-30
Star sensors	2 arc-sec	Typical field of view +-6
Gyroscopes	0.001 deg/hr	Normal use involves periodically resetting reference.
Directional antennas	0.01 to 0.5	Typically 1 of the antenna beamwidth

Table 1: Sensor Accuracy Ranges. Adapted from [11]

1.5 On-board computer

This section will describe the on-board computer which was done as part of other thesis.

2 Preliminaries

2.1 Coordinate frames

2.1.1 ECI frame

The Earth Centered Inertial frame has its x-axis pointing towards the vernal equinox, and its z-axis pointing along the rotation axis of the Earth at some initial time. The y-axis completes a right handed orthogonal coordinate system. The frame's origin is at the center of the Earth. [10]

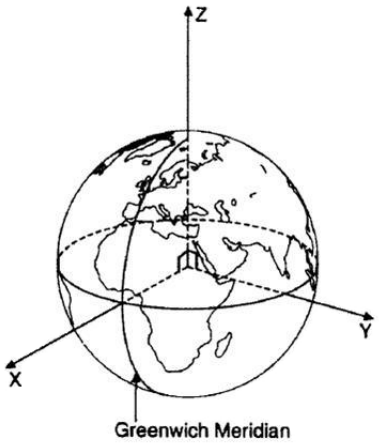

Figure 1: ECI frame, Image [10]

2.1.2 ECEF frame

This frame also has its origin at the center of the Earth, but the Earth Centered Earth Fixed frame has its x-axis pointing towards the point where the intersection between the longitude and latitude have zero value. It can also be described as the intersection between the Greenwich meridian and the Equator. The frame's z-axis is pointing along the Earth's rotation axis.

The y-axis completes the right handed orthogonal system. The ECEF frame is not an inertial frame, it rotates relative to the ECI frame along the Earth rotation.

2.1.3 NED frame

The North East Down frame has its z-axis pointing downwards, perpendicular to the tan- gent plane of the Earth's reference ellipsoid. The ellipsoid is mathematically defined and fitted for approximation of the Earth. The x-axis points towards true north and the y-axis points East. The NED frame is an inertial frame.

2.1.4 BODY frame

This frame is attached to the satellite, and is moving and rotating with it. The origin coincides with the origin of the NED frame. The axes coincide with the principle axes of inertia; the x-axis is pointing forwards, the y-axis is pointing to the right side and the z-axis is pointing downwards through the camera side of the satellite.

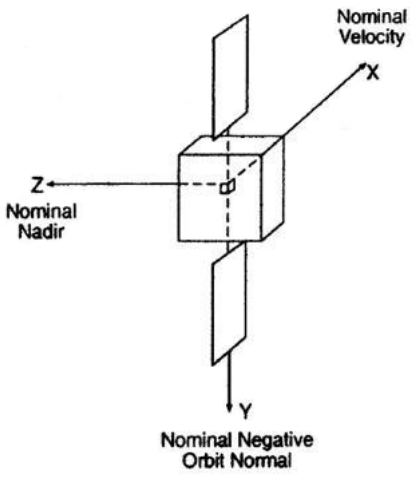

Figure 2: BODY frame, Image [10]

2.2 Space environment

2.3 Attitude representations

Several representations for describing attitude are available, the most common being Euler angles. More complicated attitude representations are quaternions. Quaternions are used for all the estimation methods presented in this thesis. They are singular-free, and are therefore well suited for attitude determination.

2.3.1 Euler angles

[13]

$$\boldsymbol{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{1}$$

$$\boldsymbol{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2}$$

$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

2.3.2 Quaternions

[14]

[15]

[16]

[17]

[18]

[19]

[20]

$$\boldsymbol{q} \coloneqq \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{4}$$

$$q_0 = \cos(v/2) \tag{5}$$

$$\boldsymbol{n} = \frac{\boldsymbol{n}}{||\boldsymbol{n}||} \tag{6}$$

$$\boldsymbol{q}_{vec} \coloneqq \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = [\boldsymbol{n} \sin(v/2)] \tag{7}$$

$$\boldsymbol{q} \coloneqq q_0 + \boldsymbol{q}_v ec = q_0 + q_1 i + q_2 j + q_3 k \tag{8}$$

$$\boldsymbol{Q} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{9}$$

$$\boldsymbol{q}^* \coloneqq q_0 - \boldsymbol{q}_{vec} = q_0 + q_1 i + q_2 j + q_3 k \tag{10}$$

$$\boldsymbol{q}^T \boldsymbol{q} = 1 \tag{11}$$

2.4 Quaternion properties

2.4.1 Advantages of quaternions

2.4.2 Multiplication of quaternions

2.4.3 Quaternions and rotations

2.5 Cholesky factorization

2.6 Lyapunov analysis

3 Star-tracker program

[22]

Generally star-tracker is divided into three main parts[23]:

- recogiting stars on the image and converting the data into list of star vectors by calculating star centroids;
- identyfing which star vector represents which real star in catalogue. This is done by comparing star vectors from the image with data in star catalogue, which is generated before space mission;
- estimating the attitude by calculating the displacement between two frames.

3.1 Centroid - start recognition

[24]

Due to limitations of camera there exists necessity of calculating star centroids. Each camera converts image into photo divided by pixels. As it is necessary to have high precision of star coordinates, the pixel accuracy is not enough. Subpixel accuracy is needed. Typically it is done by defocusing the lens of the camera and calculating the lumosity of all pixels around the lightest ones. The idea of how to calculate such centroids is adapted from[23].

If FOV is too small, one star will be considered by program as few stars, and if FOV is too large, few stars placed near each other will be considered as one star. Calculating star centroids is tradeoff between counting few stars as one and counting one star as a few. It seems however that it is worse to count one star as few than few stars as one.

$$x_{start} = x - \frac{a_{ROI} - 1}{2} \tag{12}$$

$$y_{start} = y - \frac{a_{ROI} - 1}{2} \tag{13}$$

$$x_{end} = x_{start} + a_{ROI} \tag{14}$$

$$y_{end} = y_{start} + a_{ROI} \tag{15}$$

$$I_{bottom} = \sum_{i=1}^{x_{end}-1} I(i, y_{start}) \tag{16}$$

$$I_{top} = \sum_{i=2}^{x_{end}} I(i, y_{end}) \tag{17}$$

$$I_{left} = \sum_{j=1}^{y_{end}-1} I(x_{start}, j) \tag{18}$$

$$I_{right} = \sum_{j=2}^{y_{end}} I(x_{start}, j) \tag{19}$$

$$I_{border} = \frac{I_{top} + I_{bottom} + I_{left} + I_{right}}{4(a_{ROI} - 1)} \tag{20}$$

$$\tilde{I}(x, y) = I(x, y) - I_{border} \tag{21}$$

$$B = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \tilde{I}(i, j) \tag{22}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{i \times \tilde{I}(i, j)}{B} \tag{23}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{j \times \tilde{I}(i, j)}{B} \tag{24}$$

$$u = \frac{\begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix}^T}{|| \begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix} ||} \tag{25}$$

3.2 Star identification

all [25]

Brightness Independent 4-Star Matching Algorithm for Lost-in-Space 3-Axis Attitude Acquisition[26]

SP-Search: A New Algorithm for Star Pattern Recognition [27]

Star Identification using Neural networks [28] [29]

Star pattern recognition using neural networks [30]

3.2.1 Angle Matching

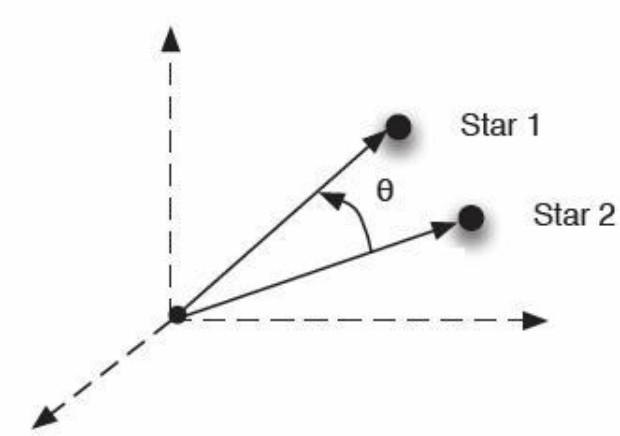

Figure 3: Vector angle method, Image [31]?

[31]

$$\theta = \cos^{-1}(\boldsymbol{r}_1 \cdot \boldsymbol{r}_2) \tag{26}$$

$$\boldsymbol{b}_i = A\boldsymbol{r}_i \tag{27}$$

$$\tilde{\boldsymbol{b}}_i = A\boldsymbol{r}_i + \boldsymbol{v}_i, \quad \boldsymbol{v}_i^T A\boldsymbol{r}_i = 0 \tag{28}$$

$$E\{\boldsymbol{v}_i\} = 0$$
$$E\left\{\boldsymbol{v}_i\boldsymbol{v}_i^T\right\} = \sigma_i^2[\boldsymbol{I} - (A\boldsymbol{r}_i)(A\boldsymbol{r}_i)^T]$$

$$\boldsymbol{b}_i^T\boldsymbol{b}_j = \boldsymbol{r}_i^T A^T A\boldsymbol{r}_j = \boldsymbol{r}_i^T\boldsymbol{r}_j \tag{30}$$

$$\tilde{\boldsymbol{b}}_i = A\boldsymbol{r}_i + \boldsymbol{v}_i$$
$$\tilde{\boldsymbol{b}}_j = A\boldsymbol{r}_j + \boldsymbol{v}_j$$

$$z \equiv \tilde{\boldsymbol{b}}_i^T \tilde{\boldsymbol{b}}_j = \boldsymbol{r}_i^T \boldsymbol{r}_j + \boldsymbol{r}_i^T A^T \boldsymbol{v}_J + \boldsymbol{r}_j^T A^T \boldsymbol{v}_i + \boldsymbol{v}_i^T \boldsymbol{v}_j \tag{32}$$

$$E\{z\} = \boldsymbol{r}_i^T \boldsymbol{r}_j \tag{33}$$

$$p \equiv z - E\{z\} = \boldsymbol{r}_i^T A^T \boldsymbol{v}_J + \boldsymbol{r}_j^T A^T \boldsymbol{v}_i + \boldsymbol{v}_i^T \boldsymbol{v}_j \tag{34}$$

$$\sigma_p^2 \equiv E\{p\} = \boldsymbol{r}_1^T A^T R_2 A \boldsymbol{r}_1 + \boldsymbol{r}_2^T A^T R_a A \boldsymbol{r}_2 + Trace(R_1 R_2) = Trace(A\boldsymbol{r}_1 \boldsymbol{r}_1^T R_2) + Trace(A\boldsymbol{r}_2 \boldsymbol{r}_2^T R_1) + Trace(R_1 R_2) \tag{35}$$

3.2.2 Spherical Triangle Matching

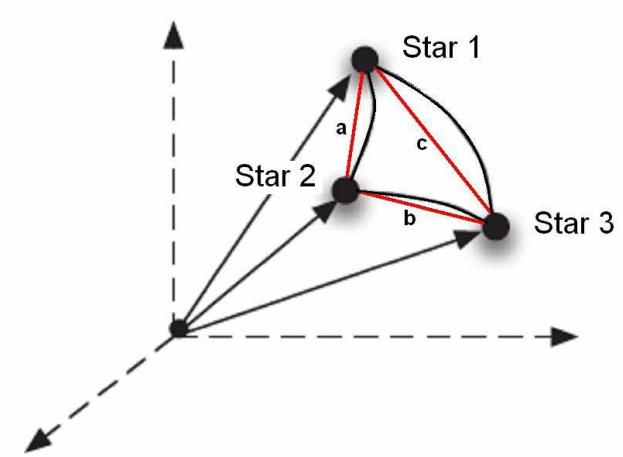

Figure 4: Spherical Triangle Method, Image [32]?

[32]

$$A = 4\tan^{-1}\sqrt{\tan\frac{s}{2}\tan\frac{s-a}{2}\tan\frac{s-b}{2}\tan\frac{s-c}{2}} \tag{36}$$

$$s = \frac{1}{2}(a+b+c)$$
$$a = \cos^{-1}\left(\frac{b_1 \cdot b_2}{|b_1||b_2|}\right)$$
$$b = \cos^{-1}\left(\frac{b_2 \cdot b_3}{|b_2||b_3|}\right)$$
$$c = \cos^{-1}\left(\frac{b_3 \cdot b_1}{|b_3||b_1|}\right)$$
$$I_p = \sum \theta^2 dA \tag{38}$$

3.2.3 Planar Triangle

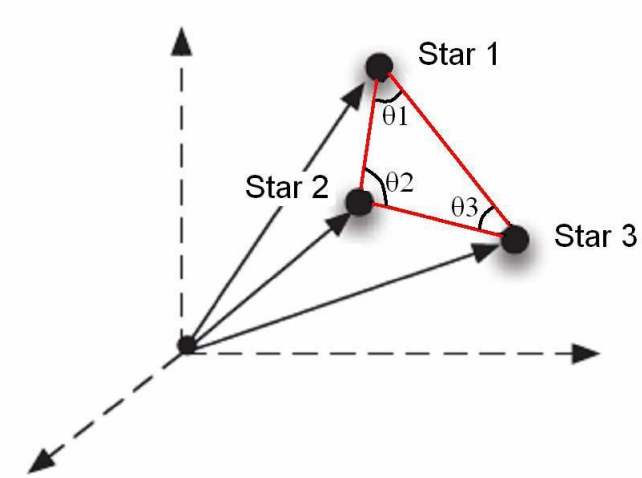

Figure 5: Planar Triangle Method, Image [33]?

[33]

$$s = \frac{1}{2}(a+b+c) \tag{39}$$

$$a = ||\boldsymbol{u_p} - \boldsymbol{u_q}|| \tag{40}$$
$$b = ||\boldsymbol{u_q} - \boldsymbol{u_r}|| \tag{41}$$
$$c = ||\boldsymbol{u_p} - \boldsymbol{u_r}|| \tag{42}$$

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{43}$$

$$J = A\frac{(a^2+b^2+c^2}{36} \tag{44}$$

Derivatives

$$H = \begin{bmatrix} \boldsymbol{h}_1^T & \boldsymbol{h}_2^T & \boldsymbol{h}_3^T \end{bmatrix} \tag{45}$$

$$\begin{aligned}
\boldsymbol{h}_1^T &\equiv \frac{\delta A}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_1} + \frac{\delta A}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_1} \\
\boldsymbol{h}_2^T &\equiv \frac{\delta A}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_2} + \frac{\delta A}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_2} \\
\boldsymbol{h}_3^T &\equiv \frac{\delta A}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_3} + \frac{\delta A}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_3}
\end{aligned}$$

$$\begin{aligned}
\frac{\delta A}{\delta a} &= \frac{u_1 - u_2 + u_3 + u_4}{4A} \\
\frac{\delta A}{\delta b} &= \frac{u_1 + u_2 - u_3 + u_4}{4A} \\
\frac{\delta A}{\delta c} &= \frac{u_1 + u_2 + u_3 - u_4}{4A}
\end{aligned}$$

$$\begin{aligned}
u_1 &= (s-a)(s-b)(s-c) \\
u_2 &= s(s-b)(s-c) \\
u_3 &= s(s-a)(s-c) \\
u_4 &= s(s-a)(s-b)
\end{aligned}$$

$$\begin{aligned}
\frac{\delta a}{\delta \boldsymbol{b}_1} &= (\boldsymbol{b}_1 - \boldsymbol{b}_2)^T/a, \quad & \frac{\delta a}{\delta \boldsymbol{b}_2} &= -\frac{\delta a}{\delta \boldsymbol{b}_1} \\
\frac{\delta b}{\delta \boldsymbol{b}_2} &= (\boldsymbol{b}_2 - \boldsymbol{b}_3)^T/b, \quad & \frac{\delta b}{\delta \boldsymbol{b}_3} &= -\frac{\delta b}{\delta \boldsymbol{b}_2} \\
\frac{\delta c}{\delta \boldsymbol{b}_1} &= (\boldsymbol{b}_1 - \boldsymbol{b}_3)^T/c, \quad & \frac{\delta c}{\delta \boldsymbol{b}_3} &= -\frac{\delta c}{\delta \boldsymbol{b}_1}
\end{aligned}$$

$$\sigma_A^2 = HRH^T \tag{50}$$

$$R \equiv \begin{bmatrix} R_1 & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & R_2 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & R_3 \end{bmatrix} \tag{51}$$

Polar Moment

$$\bar{H} = \begin{bmatrix} \bar{\boldsymbol{h}}_1^T & \bar{\boldsymbol{h}}_2^T & \bar{\boldsymbol{h}}_3^T \end{bmatrix} \tag{52}$$

$$\begin{aligned} \bar{\boldsymbol{h}}_1^T &\equiv \frac{\delta J}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_1} + \frac{\delta J}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_1} + \frac{\delta J}{\delta A}\boldsymbol{h}_1^T \\ \bar{\boldsymbol{h}}_2^T &\equiv \frac{\delta J}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_2} + \frac{\delta J}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_2} + \frac{\delta J}{\delta A}\boldsymbol{h}_2^T \\ \bar{\boldsymbol{h}}_3^T &\equiv \frac{\delta J}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_3} + \frac{\delta J}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_3} + \frac{\delta J}{\delta A}\boldsymbol{h}_3^T \end{aligned}$$

$$\frac{\delta J}{\delta a} = Aa/18, \quad \frac{\delta J}{\delta a} = Ab/18, \quad \frac{\delta J}{\delta a} = Ac/18$$
$$\frac{\delta J}{\delta A} = (a^2 + b^2 + c^2)/36$$

$$\sigma_J^2 = \bar{H}R\bar{H}^T \tag{55}$$

3.2.4 Pyramid

[34]

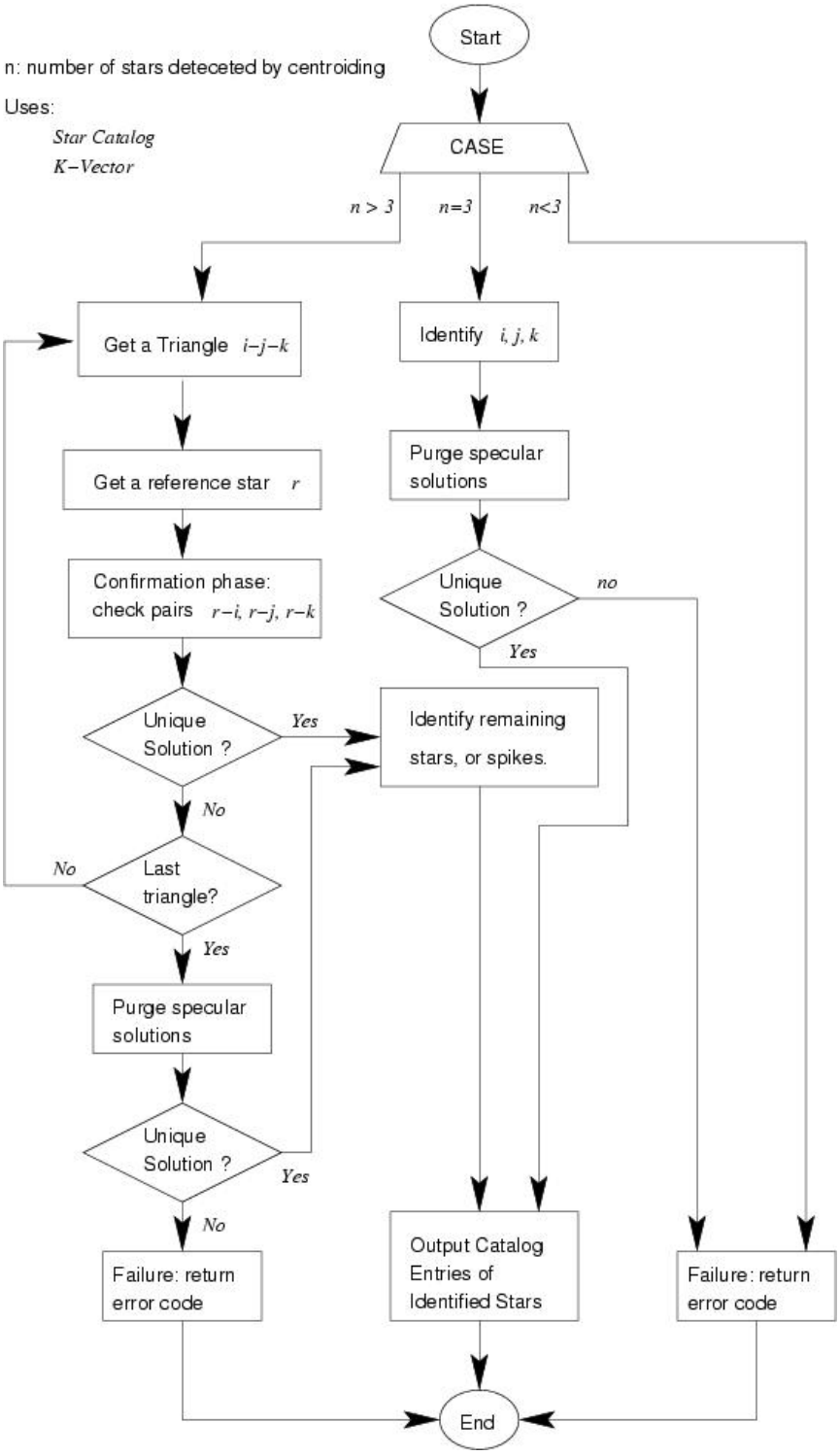

Figure 6: Pyramid Method Flowchart, Image [34]

3.2.5 Rate Matching

[35]

3.2.6 Voting

[36]

3.2.7 Grid

[37]

3.3 Star-catalogue and searching for matching stars

3.3.1 Star Catalogue Generation

$$\boldsymbol{u} = \begin{bmatrix} \cos\alpha\cos\delta \\ \sin\alpha\cos\delta \\ \sin\delta \end{bmatrix} \tag{56}$$

$$m_i \leq m_{max} \tag{57}$$

$$m_j \leq m_{max} \tag{58}$$

$$\boldsymbol{u_a^T u_b} \geq \cos\theta_{FOV} \tag{59}$$

3.3.2 Candidate Matching

3.3.3 Result Verification

3.3.4 k-vector

[38]

[39]

[40]

Trzeba dodać pogrubienia vectorów

$$z(x) = mx + q \tag{60}$$

$$m = \frac{y_{max} - y_{min} + \delta\epsilon}{n-1} \tag{61}$$

$$q = y_{min} - m - \delta\epsilon \tag{62}$$

$$\epsilon \approx 22.2 \times 10^{-16} \tag{63}$$

$$\delta\epsilon = (n-1)\epsilon \tag{64}$$

$$k(i) = j \quad where \quad s(j) \leq z(i) < s(j+1) \tag{65}$$

or

$$k(i) = j \quad wherejisthegreatestindexsuch \quad s(j) \leq y(I(i)) \quad issatisfied. \tag{66}$$

$$j_b = \left\lfloor \frac{y_a - q}{m} \right\rfloor \quad and \quad j_t = \left\lceil \frac{y_b - q}{m} \right\rceil \tag{67}$$

$$k_{start} = k(j_b) + 1 \quad and \quad k_{end} = k(j_t) \tag{68}$$

3.4 Attitude Determination

[1]

AIM (Attitude estimation using Image Matching)[3]

all [11] [41]

3.4.1 The Predictive Attitude Determination Algorithm ?

[42]

3.4.2 q-method

3.5 Wahba's problem

[21]

$$\sum_{j}^{n} ||r_j - Mb_j|| \tag{69}$$

3.5.1 QUEST

improvement to quest implementation [43]

kallman filtering [44]

$$\begin{aligned} J(\boldsymbol{q}) = \frac{1}{2}\sum_{j=1}^{n}\frac{1}{\sigma_j^2}(\boldsymbol{b}_j - \boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j)^T(\boldsymbol{b}_j - \boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j) = \\ \frac{1}{2}\sum_{j=1}^{n}\frac{1}{\sigma_j^2}(\boldsymbol{b}_j^T\boldsymbol{b}_j - 2\boldsymbol{b}_j^T\boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j + \boldsymbol{r}_j^T\boldsymbol{r}_j) \end{aligned} \tag{70}$$

$$J(\boldsymbol{q}) = \sum_{j=1}^{n}\frac{1}{\sigma_j^2}(1 - \boldsymbol{b}_j^T\boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j) \tag{71}$$

3.5.2 TRIAD

3.5.3 The Fast Optimal Attitude Matrix

3.5.4 DCM (Direction Cosine Matrix)

[45] and

[23]

$$\boldsymbol{B} = \sum_{i=1}^{n} \boldsymbol{b_i r_i^T} \tag{72}$$

$$\boldsymbol{B} = \boldsymbol{U S V^T} \tag{73}$$

$$\boldsymbol{U_+} = \boldsymbol{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{U} \end{bmatrix} \tag{74}$$

$$\boldsymbol{V_+} = \boldsymbol{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{V} \end{bmatrix} \tag{75}$$

$$\boldsymbol{A} = \boldsymbol{U_+ V_+^T} \tag{76}$$

4 Prototype

For now the following parts are finished in Python:

1. Centroiding
2. Planar Triangle Recognition with variations (nearly)
3. Pyramid alg
4. k-vector
5. QUEST (not started yet)

Testing

[46]

5 Complete program

6 Testing of star-tracker

[47]

References

[1] K. L. Jenssen, K. H. Yabar, and J. T. Gravdahl, "A comparison of attitude determination methods: theory and experiments," in *proceedings of the 62nd International Astronautical Congress, Cape Town, South Africa*, pp. 3–7, 2011.

[2] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs," *Sensors*, vol. 15, no. 8, pp. 19302–19330, 2015.

[3] T. Delabie, "A highly efficient attitude estimation algorithm for star trackers based on optimal image matching," in *AIAA Guidance, Navigation and Control Conference, Minneapolis, Minnesota*, 2012.

[4] E. Jalabert, E. Fabacher, N. Guy, S. Lizy-Destrez, W. Rappin, and G. Rivier, "Optimization of star research algorithm for ESMO star tracker," 2011.

[5] D. Felikson, J. Hahmall, M. F. Vess, and M. Ekinci, "On-Orbit Solar Dynamics Observatory (SDO) Star Tracker Warm Pixel Analysis," in *AIAA Guidance, Navigation and Control Conference, Portland, Oregon*, vol. 6728, 2011.

[6] M. W. Knutson, *Fast star tracker centroid algorithm for high performance CubeSat with air bearing validation.* PhD thesis, Massachusetts Institute of Technology, 2012.

[7] A. Rose, "STAR integrated tracker," *arXiv preprint nucl-ex/0307015*, 2003.

[8] D. Mortari and A. Romoli, "StarNav III: a three fields of view star tracker," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 1, pp. 1–57, IEEE, 2002.

[9] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, "CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation," 2000.

[10] W. J. Larson and J. R. Wertz, "Space mission analysis and design," tech. rep., Microcosm, Inc., Torrance, CA (US), 1992.

[11] C. D. Hall, "Spacecraft attitude dynamics and control," *Lecture Notes posted on Handouts page [online]*, vol. 12, no. 2003, 2003.

[12] S. M. R. C. P. Lima, "Comparison of small satellite attitude determination methods," 2000.

[13] L. Euler, "Formulae generales pro translatione quacunque corporum rigidorum," *Novi Acad. Sci. Petrop*, vol. 20, pp. 189–207, 1775.

[14] W. R. Hamilton, "LXXVIII. On quaternions; or on a new system of imaginaries in Algebra: To the editors of the Philosophical Magazine and Journal," 1844.

[15] A. Cayley, "XIII. On certain results relating to quaternions: To the editors of the Philosophical Magazine and Journal," 1845.

[16] R. Courant and D. Hilbert, "Methods of mathematical physics, Volume I," 1953.

[17] J. E. Mebius, "A matrix-based proof of the quaternion representation theorem for four-dimensional rotations," *arXiv preprint math/0501249*, 2005.

[18] M. Barile, "Conjugate elements."

[19] K. Shoemake, "Animating rotation with quaternion curves," in *ACM SIGGRAPH computer graphics*, vol. 19, pp. 245–254, ACM, 1985.

[20] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[21] G. Wahba, "A least squares estimate of satellite attitude," *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965.

[22] G. Ju and J. L. Junkins, "Overview of star tracker technology and its trends in research and development," *Advances in the Astronautical Sciences*, vol. 115, pp. 461–477, 2003.

[23] C. R. McBryde and E. G. Lightsey, "A star tracker design for CubeSats," in *Aerospace Conference, 2012 IEEE*, pp. 1–14, March 2012.

[24] M. A. Samaan, D. Mortari, T. Pollock, and J. L. Junkins, "Predictive centroiding for single and multiple FOVs star trackers," *Advances in the Astronautical Sciences*, vol. 112, pp. 59–71, 2002.

[25] B. B. Spratling and D. Mortari, "A survey on star identification algorithms," *Algorithms*, vol. 2, no. 1, pp. 93–107, 2009.

[26] Y. Dong, F. Xing, and Z. You, "Brightness independent 4-star matching algorithm for lost-in-space 3-axis attitude acquisition," *Tsinghua Science & Technology*, vol. 11, no. 5, pp. 543–548, 2006.

[27] D. Mortari, "SP-search: A new algorithm for star pattern recognition," *Advances in the Astronautical Sciences*, vol. 102, no. Pt II, pp. 1165–1174, 1999.

[28] S. S. Miri and M. E. Shiri, "Star identification using Delaunay triangulation and distributed neural networks," *International Journal of Modeling and Optimization*, vol. 2, no. 3, p. 234, 2012.

[29] T. Lindblad, C. S. Lindsey, Å. Eide, Ö. Solberg, and A. Bolseth, "Star Identification using Neural Networks,"

[30] C. Li, K. Li, L. Zhang, S. Jin, and J. Zu, "Star pattern recognition method based on neural network," *Chinese Science Bulletin*, vol. 48, no. 18, pp. 1927–1930, 2003.

[31] D. Gottlieb, "Star pattern recognition techniques," *Spacecraft Attitude Determination and Control, The Netherlands*, pp. 257–266, 1978.

[32] C. L. Cole and J. Crassidus, "Fast star pattern recognition using spherical triangles," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit. Providence, Rhode Island: AIAA*, 2004.

[33] C. L. Cole and J. L. Crassidis, "Fast star-pattern recognition using planar triangles," *Journal of guidance, control, and dynamics*, vol. 29, no. 1, pp. 64–71, 2006.

[34] D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junkins, "The pyramid star identification technique," *Navigation*, vol. 51, no. 3, pp. 171–183, 2004.

[35] "Recursive mode star identification algorithms,"

[36] M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum, "Geometric voting algorithm for star trackers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 441–456, 2008.

[37] C. Padgett and K. Kreutz-Delgado, "A grid algorithm for autonomous star identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 202–213, 1997.

[38] D. Mortari and J. Rogers, "A k-vector Approach to Sampling, Interpolation, and Approximation," *The Journal of the Astronautical Sciences*, vol. 60, no. 3-4, pp. 686–706, 2013.

[39] D. Mortari, "A fast on-board autonomous attitude determination system based on a new star-ID technique for a wide FOV star tracker," *Advances in the Astronautical Sciences*, vol. 93, pp. 893–904, 1996.

[40] D. Mortari and B. Neta, "K-vector range searching techniques," *Adv. Astronaut. Sci*, vol. 105, pp. 449–464, 2000.

[41] F. L. Markley and D. Mortari, "How to estimate attitude from vector observations," 1999.

[42] K. J. Park and J. L. Crassidis, "Attitude determination methods using pseudolite signal phase measurements," *Navigation*, vol. 53, no. 2, pp. 121–133, 2006.

[43] Cheng Yang and Shuster Malcolm D., "Improvement to the Implementation of the QUEST Algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 301–305, 2013. doi: 10.2514/1.62549.

[44] M. Shuster, "Kalman filtering of spacecraft attitude and the QUEST model," *Journal of the Astronautical Sciences*, vol. 38, pp. 377–393, 1990.

[45] J.-N. Juang, H.-Y. Kim, and J. L. Junkins, "An efficient and robust singular value method for star pattern recognition and attitude determination," 2003.

[46] M. Kruijff, E. Heide, C. De Boom, and N. Heiden, "Star sensor algorithm application and spin-off," in *54th International Astronautical Congress of the International Astronautical Federation(IAF)*, 2003.

[47] J.-J. Kim, J. Tappe, A. Jordan, and B. Agrawal, *Star Tracker Attitude Estimation for an Indoor Ground-Based Spacecraft Simulator*. Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, aug 2011. doi:10.2514/6.2011-6270.

List of Tables

List of Figures

Robot Learning Darmstadt Problems with Euler Angles: Not Unique: Many angles result in the same rotation Hard to quantify differences between two Euler Angles Unit-Quaternion Solves the problems of singularities with the Euler Angles Easier to compute differences of orientations Important if we want to control the orientation of the end-effector See Siciliano or Spong Textbook!

Polar moment