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

Contents

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 Caltech in 1999. 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.

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.

Table 1

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

2.1.2 ECEF frame

2.1.3 NED frame

2.1.4 BODY frame

2.2 Attitude representations

2.2.1 Euler angles

2.2.2 Quaternions

2.3 Quaternion properties

2.3.1 Advantages of quaternions

2.3.2 Multiplication of quaternions

2.3.3 Quaternions and rotations

2.4 Wahba's problem

Hello [1]

$$\sum_{j}^{n} ||r_j - Mb_j|| \tag{1}$$

2.5 Cholesky factorization

2.6 Lyapunov analysis

3 Star-tracker program

Generally star-tracker is divided into three main parts[2]:

- 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

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[2].

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.

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

$$y_{start} = y - \frac{a_{ROI} - 1}{2} \tag{3}$$

$$x_{end} = x_{start} + a_{ROI} \tag{4}$$

$$y_{end} = y_{start} + a_{ROI} \tag{5}$$

$$I_{bottom} = \sum_{i=1}^{x_{end}-1} I(i, y_{start}) \tag{6}$$

$$I_{top} = \sum_{i=2}^{x_{end}} I(i, y_{end}) \tag{7}$$

$$I_{left} = \sum_{j=1}^{y_{end}-1} I(x_{start}, j) \tag{8}$$

$$I_{right} = \sum_{j=2}^{y_{end}} I(x_{start}, j) \tag{9}$$

$$I_{border} = \frac{I_{top} + I_{bottom} + I_{left} + I_{right}}{4(a_{ROI} - 1)} \tag{10}$$

$$\tilde{I}(x, y) = I(x, y) - I_{border} \tag{11}$$

$$B = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \tilde{I}(i, j) \tag{12}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{i \times \tilde{I}(i, j)}{B} \tag{13}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{j \times \tilde{I}(i, j)}{B} \tag{14}$$

$$u = \frac{\begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix}^T}{|| \begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix} ||} \tag{15}$$

3.2 Star identification

3.2.1 Angle Matching

3.2.2 Spherical Triangle Matching

3.2.3 Planar Triangle

$$s = \frac{1}{2}(a + b + c) \tag{16}$$

$$a = ||\boldsymbol{u_p} - \boldsymbol{u_q}|| \tag{17}$$

$$b = ||\boldsymbol{u_q} - \boldsymbol{u_r}|| \tag{18}$$

$$c = ||\boldsymbol{u_p} - \boldsymbol{u_r}|| \tag{19}$$

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{20}$$

$$J = A\frac{(a^2 + b^2 + c^2}{36} \tag{21}$$

3.2.4 Pyramid

3.2.5 Rate Matching

3.2.6 Voting

3.2.7 Grid

3.2.8 Different elements of algorithm

3.3 Star Catalogue Generation

$$\boldsymbol{u} = \begin{bmatrix} \cos\alpha\cos\delta \\ \sin\alpha\cos\delta \\ \sin\delta \end{bmatrix} \tag{22}$$

$$m_i \leq m_{max} \tag{23}$$

$$m_j \leq m_{max} \tag{24}$$

$$\boldsymbol{u_a^T u_b} \geq \cos\theta_{FOV} \tag{25}$$

3.4 Candidate Matching

3.5 Result Verification

3.6 k-vector

3.7 Attitude Determination

3.7.1 QUEST

3.7.2 TRIAD

3.7.3 The Fast Optimal Attitude Matrix

3.7.4 q-method

3.7.5 DCM (direction cosine matrix) - (Singular Value Decomposition?)

[2]

$$\boldsymbol{B} = \sum_{i=1}^{n} \boldsymbol{b_i r_i^T} \tag{26}$$

$$\boldsymbol{B} = \boldsymbol{USV^T} \tag{27}$$

$$\boldsymbol{U_+} = \boldsymbol{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{U} \end{bmatrix} \tag{28}$$

$$\boldsymbol{V_+} = \boldsymbol{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{V} \end{bmatrix} \tag{29}$$

$$\boldsymbol{A} = \boldsymbol{U_+ V_+^T} \tag{30}$$

4 Prototype

5 Complete program

6 Testing of star-tracker

References

[1] 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.

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

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