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

Last years directed space industry towards small satellites. Many countries which did not have possibility to enter this branch of industry before now create their own solutions. The goal of this work is to create fully functioning star-tracker program eligible to be used in future satellites as Polish sulition of determination of satellite attitude towards Earth. Work contains also description of individual parts and variants of solutions connected with star-tracker.

Streszczenie

Ostatnie lata ukierunkowały przemysł kosmiczny na małe satelity. Wiele krajów, które wcześniej nie miały możliwości wejścia w tą gałąź przemysłu, teraz tworzą własne rozwiązania. Celem niniejszej pracy dyplomowej jest stworzenie w pełni działającego programu star-tracker nadającego się do wykorzystania w przyszłych satelitach jako polskie rozwiązanie problemu określania orientacji satelity względem ziemi. Praca zawiera też opis poszczególnych części i wariantów rozwiazania problemów związanych ze star-trackerem.

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	1.0°(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]

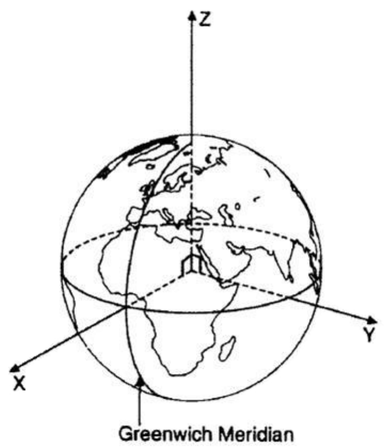

Figure 1: ECI frame, Image [10]

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.

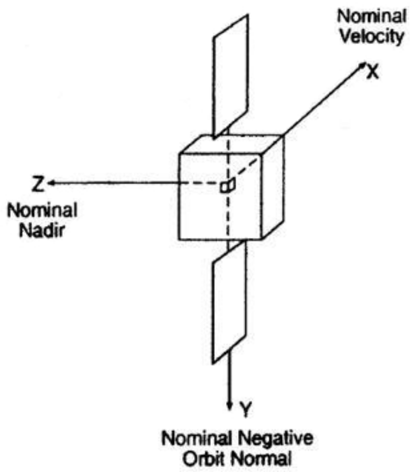

Figure 2: BODY frame, Image [10]

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]

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

Euler angles were first described by Leonhard Euler in 1776, and are used to represent the orientation of a body [10]. Three parameters are required for a full understanding of the orientation between two frames, one angle for the rotation around each of the axes. The angles are called roll, pitch and yaw and are usually written as phi, ro and psi. The Euler angles are often used for the definition of rotation matrices about the x, y and z-axis. In R 3 , the coordinate system rotations in a counter-clockwise direction looking towards the origin are given from:

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

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

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

2.3.2 Quaternions

[14]

[15]

[16]

[17]

[18]

[19]

[20]

Quaternions were first described by Sir William Rowan Hamilton in 1843 [11]. His inten- sion was to find an extension of vector algebra, and in 1845 Arthur Cayley published an article where he used multiplication of quaternions to describe rotations [12]. Three of the four elements of a quaternion give the coordinates for the axis of rotation, while the fourth is described by the angle of rotation [13]. A quaternion can be written as a four-dimensional vector:

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

The real part of the quaternion behaves like a scalar in the three-dimensional vector space. Using a rotation angle vi, the real part can be written as:

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

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

The imaginary part uses a unit vector given from n = n/——n—— (wzor) where the norm of the vector, ——n——, is defined as the square root of each of the squared elements of n added together. Throughout the thesis, vectors and matrices will be written in bold print. The imaginary part can be written as a vector:

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

There are several ways to write quaternions. Sometimes it is convenient to think of a quaternion as the sum of a scalar and a vector written as:

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

Complex numbers can be represented as matrices, and so can quaternions. A quaternion describes a point in 4D space, and can be represented by a 4 × 4 matrix by using a left- isoclinic rotation as proved in [14], and used in [16] and [17]:

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

The transpose of the matrix is the same as the conjugate of the quaternion:

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

Two quaternions are conjugate if they are orthogonal with respect to their inner product. [15]. The inverse of a quaternion q is defined as q -1 = 1/q .

In this report, the mathematics are based on unit quaternions, which satisfies the constraint:

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

The length of a unit quaternion is 1, which leads its inverse to be its conjugate.

2.4 Quaternion properties

2.4.1 Advantages of quaternions

The unit quaternion notation is compact, and round off errors are easier to handle than for matrix representation. The nearest orthonormal matrix to one that is not quite orthonormal, is difficult to find. Multiplying unit quaternions may similarly lead to quaternions that are no longer of unit length, but these can easily be normalized to make sure they correspond to valid rotations. The computational cost of normalizing a quaternion is much less than for normalizing a matrix. Quaternions are safe from a phenomenon called gimbal lock. When the pitch angle in a pitch/roll/yaw-system is rotated 90 ∘ up or down, and the yaw and roll correspond to the same motion, a degree of freedom of rotation can be lost. In a gimbal-based aerospace inertial navigation system, this could have disastrous results if the aircraft is in a steep dive or ascent [16]. The quaternion elements vary continuously over the unit sphere in R 4 , (denoted by S 3) as the orientation changes, avoiding this problem. Due to the possible spin in the CubeSat, combined with the singularity problem for Euler angles, an attitude estimation method based on unit quaternions is preferred [4].

2.4.2 Multiplication of quaternions

2.4.3 Quaternions and rotations

3 Star-tracker program

[21]

Generally star-tracker is divided into three main parts[22]:

- 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

[23]

[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[22].

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.

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

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

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

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

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

$$I_{top} = \sum_{i=2}^{x_{end}} I(i, y_{end}) \tag{16b}$$

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

$$I_{right} = \sum_{j=2}^{y_{end}} I(x_{start}, j) \tag{16d}$$

$$I_{border} = \frac{I_{top} + I_{bottom} + I_{left} + I_{right}}{4(a_{ROI} - 1)} \tag{16e}$$

$$\tilde{I}(x, y) = I(x, y) - I_{border} \tag{17}$$

$$B = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \tilde{I}(i, j) \tag{18}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{i \times \tilde{I}(i, j)}{B} \tag{19}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{j \times \tilde{I}(i, j)}{B} \tag{20}$$

$$u = \frac{\begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix}^T}{|| \begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix} ||} \tag{21}$$

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

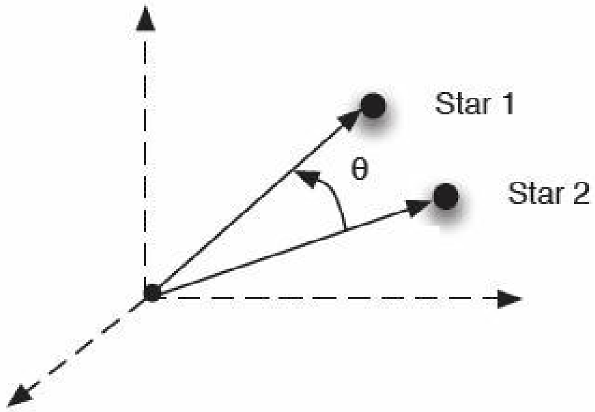

Figure 3: Vector angle method, Image [31]?

[31]

$$\theta = \cos^{-1}(\boldsymbol{r}_1 \cdot \boldsymbol{r}_2) \tag{22}$$

$$\boldsymbol{b}_i = A\boldsymbol{r}_i \tag{23}$$

$$\tilde{\boldsymbol{b}}_i = A\boldsymbol{r}_i + \boldsymbol{v}_i, \quad \boldsymbol{v}_i^T A\boldsymbol{r}_i = 0 \tag{24}$$

$$E\left\{\boldsymbol{v}_i\right\} = 0 \tag{25a}$$

$$E\left\{\boldsymbol{v}_i\boldsymbol{v}_i^T\right\} = \sigma_i^2[\boldsymbol{I} - (A\boldsymbol{r}_i)(A\boldsymbol{r}_i)^T] \tag{25b}$$

$$\boldsymbol{b}_i^T\boldsymbol{b}_j = \boldsymbol{r}_i^T A^T A\boldsymbol{r}_j = \boldsymbol{r}_i^T\boldsymbol{r}_j \tag{26}$$

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

$$z \equiv \tilde{\boldsymbol{b}}_i^T \tilde{\boldsymbol{b}}_j = \boldsymbol{r}_i^T \boldsymbol{r}_j + \boldsymbol{r}_i^T A^T \boldsymbol{v}_J + \boldsymbol{r}_j^T A^T \boldsymbol{v}_i + \boldsymbol{v}_i^T \boldsymbol{v}_j \tag{28}$$

$$E\{z\} = \boldsymbol{r}_i^T \boldsymbol{r}_j \tag{29}$$

$$p \equiv z - E\{z\} = \boldsymbol{r}_i^T A^T \boldsymbol{v}_J + \boldsymbol{r}_j^T A^T \boldsymbol{v}_i + \boldsymbol{v}_i^T \boldsymbol{v}_j \tag{30}$$

$$\begin{aligned} \sigma_p^2 \equiv E\{p\} = \\ \boldsymbol{r}_1^T A^T R_2 A \boldsymbol{r}_1 + \boldsymbol{r}_2^T A^T R_a A \boldsymbol{r}_2 + Trace(R_1 R_2) = \\ Trace(A\boldsymbol{r}_1 \boldsymbol{r}_1^T R_2) + Trace(A\boldsymbol{r}_2 \boldsymbol{r}_2^T R_1) + Trace(R_1 R_2) \end{aligned} \tag{31}$$

3.2.2 Spherical Triangle Matching

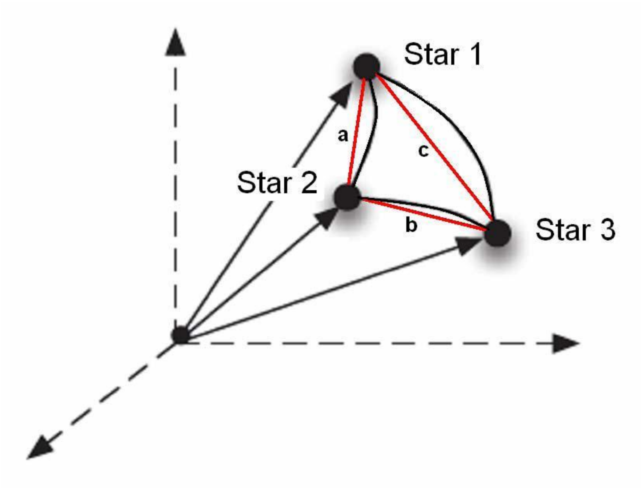

Figure 4: Spherical Triangle Method, Image [32]?

[32]

$$A = 4\tan^{-1}\sqrt{\tan\frac{s}{2}\tan\frac{s-a}{2}\tan\frac{s-b}{2}\tan\frac{s-c}{2}} \tag{32}$$

$$
\begin{aligned}
s &= \frac{1}{2}(a+b+c) \\
a &= \cos^{-1}\left(\frac{b_1 \cdot b_2}{|b_1||b_2|}\right) \\
b &= \cos^{-1}\left(\frac{b_2 \cdot b_3}{|b_2||b_3|}\right) \\
c &= \cos^{-1}\left(\frac{b_3 \cdot b_1}{|b_3||b_1|}\right) \\
I_p &= \sum \theta^2 dA
\end{aligned} \tag{34}
$$

3.2.3 Planar Triangle

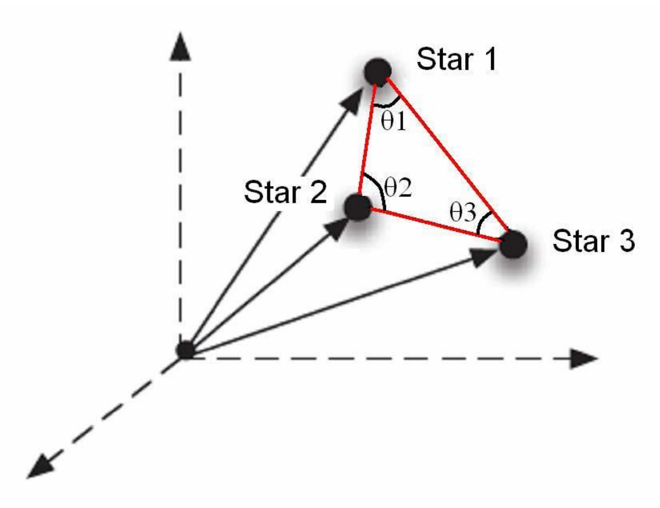

Figure 5: Planar Triangle Method, Image [33]?

[33]

$$s = \frac{1}{2}(a+b+c) \tag{35a}$$

$$a = ||\boldsymbol{u_p} - \boldsymbol{u_q}|| \tag{35b}$$

$$b = ||\boldsymbol{u_q} - \boldsymbol{u_r}|| \tag{35c}$$

$$c = ||\boldsymbol{u_p} - \boldsymbol{u_r}|| \tag{35d}$$

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{36}$$

$$J = A\frac{(a^2 + b^2 + c^2)}{36} \tag{37}$$

Derivatives

$$H = \begin{bmatrix} \boldsymbol{h}_1^T & \boldsymbol{h}_2^T & \boldsymbol{h}_3^T \end{bmatrix} \tag{38}$$

$$\boldsymbol{h}_1^T \equiv \frac{\delta A}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_1} + \frac{\delta A}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_1} \tag{39a}$$

$$\boldsymbol{h}_2^T \equiv \frac{\delta A}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_2} + \frac{\delta A}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_2} \tag{39b}$$

$$\boldsymbol{h}_3^T \equiv \frac{\delta A}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_3} + \frac{\delta A}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_3} \tag{39c}$$

$$\frac{\delta A}{\delta a} = \frac{u_1 - u_2 + u_3 + u_4}{4A} \tag{40a}$$

$$\frac{\delta A}{\delta b} = \frac{u_1 + u_2 - u_3 + u_4}{4A} \tag{40b}$$

$$\frac{\delta A}{\delta c} = \frac{u_1 + u_2 + u_3 - u_4}{4A} \tag{40c}$$

$$u_1 = (s-a)(s-b)(s-c) \tag{41a}$$

$$u_2 = s(s-b)(s-c) \tag{41b}$$

$$u_3 = s(s-a)(s-c) \tag{41c}$$

$$u_4 = s(s-a)(s-b) \tag{41d}$$

$$\frac{\delta a}{\delta \boldsymbol{b}_1} = (\boldsymbol{b}_1 - \boldsymbol{b}_2)^T/a, \quad \frac{\delta a}{\delta \boldsymbol{b}_2} = -\frac{\delta a}{\delta \boldsymbol{b}_1} \tag{42a}$$

$$\frac{\delta b}{\delta \boldsymbol{b}_2} = (\boldsymbol{b}_2 - \boldsymbol{b}_3)^T/b, \quad \frac{\delta b}{\delta \boldsymbol{b}_3} = -\frac{\delta b}{\delta \boldsymbol{b}_2} \tag{42b}$$

$$\frac{\delta c}{\delta \boldsymbol{b}_1} = (\boldsymbol{b}_1 - \boldsymbol{b}_3)^T/c, \quad \frac{\delta c}{\delta \boldsymbol{b}_3} = -\frac{\delta c}{\delta \boldsymbol{b}_1} \tag{42c}$$

$$\sigma_A^2 = HRH^T \tag{43}$$

$$R \equiv \begin{bmatrix} R_1 & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & R_2 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & R_3 \end{bmatrix} \tag{44}$$

Polar Moment

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

$$\bar{\boldsymbol{h}}_1^T \equiv \frac{\delta J}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_1} + \frac{\delta J}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_1} + \frac{\delta J}{\delta A}\boldsymbol{h}_1^T \tag{46a}$$

$$\bar{\boldsymbol{h}}_2^T \equiv \frac{\delta J}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_2} + \frac{\delta J}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_2} + \frac{\delta J}{\delta A}\boldsymbol{h}_2^T \tag{46b}$$

$$\bar{\boldsymbol{h}}_3^T \equiv \frac{\delta J}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_3} + \frac{\delta J}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_3} + \frac{\delta J}{\delta A}\boldsymbol{h}_3^T \tag{46c}$$

$$\frac{\delta J}{\delta a} = Aa/18, \quad \frac{\delta J}{\delta a} = Ab/18, \quad \frac{\delta J}{\delta a} = Ac/18 \tag{47a}$$

$$\frac{\delta J}{\delta A} = (a^2 + b^2 + c^2)/36 \tag{47b}$$

$$\sigma_J^2 = \bar{H}R\bar{H}^T \tag{48}$$

3.2.4 Pyramid

[34]

3.2.5 Rate Matching

[35] to be removed?

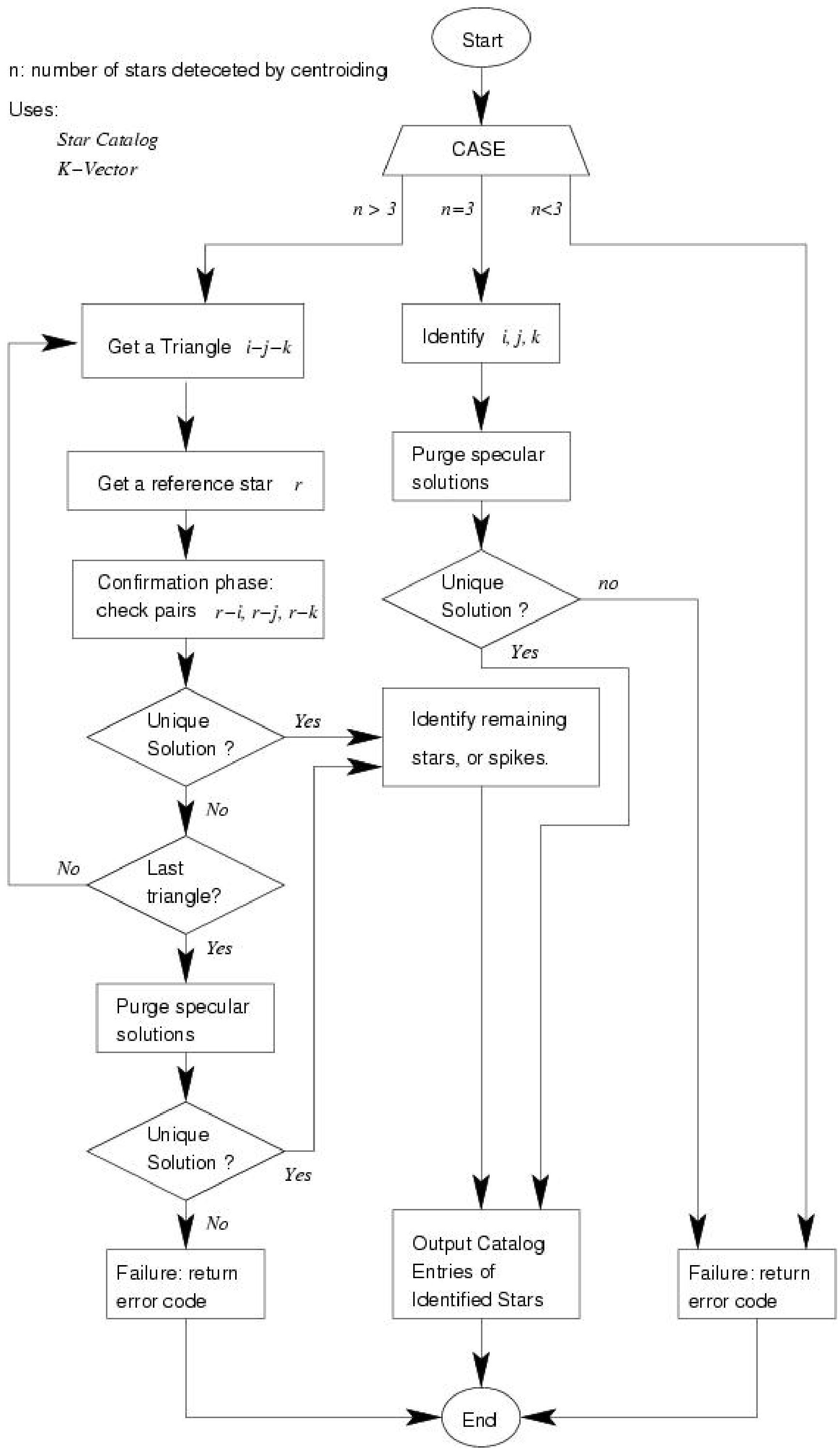

Figure 6: Pyramid Method Flowchart, Image [34]

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

$$\boldsymbol{u} = \begin{bmatrix} \cos\alpha\cos\delta \\ \sin\alpha\cos\delta \\ \sin\delta \end{bmatrix} \tag{49}$$

$$m_i \leq m_{max} \tag{50}$$

$$m_j \leq m_{max} \tag{51}$$

$$\boldsymbol{u_a^T}\boldsymbol{u_b} \geq \cos\theta_{FOV} \tag{52}$$

3.3.2 Candidate Matching

to be removed?

3.3.3 Result Verification

to be removed?

3.3.4 k-vector

The k-vector database is built a priori for some given working magnitude threshold and for the star tracker maximum angular aperture. Essentially, the k-vector table is a structural database of all cataloged star pairs that could possibly fit in the camera FOV over the whole sky. The star pairs are ordered with increasing interstar angle. The data stored are the k index, the cosine of the interstar angle, and the master catalog indices I[k] and J[k] of the kth star pair. The k-vector access logic is invoked in real time for a minimal set of star pairs in elementary measured star polygons (three for a triangle, six for a four-star pyramid, etc.); the fact that the vertices between adjacent measured star pairs share a common cataloged star is the key observation leading to logic for efficiently identifying the stars by simply comparing the k-vector accessed catalog indices from the several sets of candidate star pairs (which must contain the common measured pivot star, if it is in the catalog). [38]

[39]

[40]

Trzeba dodać pogrubienia vectorów

$$z(x) = mx + q \tag{53}$$

$$m = \frac{y_{max} - y_{min} + \delta\epsilon}{n-1} \tag{54}$$

$$q = y_{min} - m - \delta\epsilon \tag{55}$$

$$\epsilon \approx 22.2 \times 10^{-16} \tag{56}$$

$$\delta\epsilon = (n-1)\epsilon \tag{57}$$

$$k(i) = j \quad where \quad s(j) \leq z(i) < s(j+1) \tag{58}$$

or

$$k(i) = j \quad \text{where j is the greatest index such} \quad s(j) \leq y(I(i)) \quad \text{is satisfied.} \tag{59}$$

$$j_b = \left\lfloor \frac{y_a - q}{m} \right\rfloor \quad and \quad j_t = \left\lceil \frac{y_b - q}{m} \right\rceil \tag{60}$$

$$k_{start} = k(j_b) + 1 \quad and \quad k_{end} = k(j_t) \tag{61}$$

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

$$\boldsymbol{s}_b = \boldsymbol{R}^{bi}\boldsymbol{s}_i \quad \boldsymbol{m}_b = \boldsymbol{R}^{bi}\boldsymbol{m}_i \tag{62}$$

$$\begin{aligned} J &= \frac{1}{2}\sum w_k(\boldsymbol{v}_{kb} - \boldsymbol{R}^{bi}\boldsymbol{v}_{ki})^T(\boldsymbol{v}_{kb} - \boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \\ &= \frac{1}{2}\sum w_k(\boldsymbol{v}_{kb}^T\boldsymbol{v}_{kb} + \boldsymbol{v}_{ki}^T\boldsymbol{v}_{ki} + 2\boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \end{aligned} \tag{63}$$

$$J = \sum w_k(1 - \boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \tag{64}$$

$$g(\boldsymbol{R}) = \sum w_k\boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki} \tag{65}$$

$$\boldsymbol{R} = (q_4^2 - \boldsymbol{q}^T\boldsymbol{q})1 + 2\boldsymbol{q}\boldsymbol{q}^T - 2q_4\boldsymbol{q}^{\boldsymbol{x}} \tag{66}$$

$$\bar{\boldsymbol{q}}^T\bar{\boldsymbol{q}} = 1 \tag{67}$$

$$g(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T \boldsymbol{K} \bar{\boldsymbol{q}} \tag{68}$$

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{S} - \sigma \boldsymbol{I} & \boldsymbol{Z} \\ \boldsymbol{Z}^T & \sigma \end{bmatrix} \tag{69}$$

$$\boldsymbol{B} = \sum_{k=1}^{N} w_k (\boldsymbol{v}_{kb} \boldsymbol{v}_{ki}^T) \tag{70}$$

$$\boldsymbol{S} = \boldsymbol{B} + \boldsymbol{B}^T \tag{71}$$

$$\boldsymbol{Z} = \begin{bmatrix} B_{23} - B_{32} & B_{32} - B_{13} & B_{12} - B_{21} \end{bmatrix}^T \tag{72}$$

$$\sigma = tr[\boldsymbol{B}] \tag{73}$$

$$g'(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T \boldsymbol{K} \bar{\boldsymbol{q}} - \lambda \bar{\boldsymbol{q}}^T \bar{\boldsymbol{q}} \tag{74}$$

$$\boldsymbol{K} \bar{\boldsymbol{q}} = \lambda \bar{\boldsymbol{q}} \tag{75}$$

$$g(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T \boldsymbol{K} \bar{\boldsymbol{q}} = \bar{\boldsymbol{q}}^T \lambda \bar{\boldsymbol{q}} = \lambda \bar{\boldsymbol{q}}^T \bar{\boldsymbol{q}} = \lambda \tag{76}$$

3.4.3 Wahba's problem

[43]

$$\sum_{j}^{n} ||r_j - M b_j|| \tag{77}$$

3.4.4 QUEST

improvement to quest implementation [44]

kallman filtering [45]

$$J(\boldsymbol{q}) = \frac{1}{2}\sum_{j=1}^{n}\frac{1}{\sigma_j^2}(\boldsymbol{b}_j - \boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j)^T(\boldsymbol{b}_j - \boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j) = \frac{1}{2}\sum_{j=1}^{n}\frac{1}{\sigma_j^2}(\boldsymbol{b}_j^T\boldsymbol{b}_j - 2\boldsymbol{b}_j^T\boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j + \boldsymbol{r}_j^T\boldsymbol{r}_j) \tag{78}$$

$$J(\boldsymbol{q}) = \sum_{j=1}^{n}\frac{1}{\sigma_j^2}(1 - \boldsymbol{b}_j^T\boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j) \tag{79}$$

3.4.5 TRIAD

a must

3.4.6 The Fast Optimal Attitude Matrix

to be removed?

3.4.7 DCM (Direction Cosine Matrix)

[46] and

[22]

$$\boldsymbol{B} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\boldsymbol{b_i r_i^T} \tag{80}$$

$$\boldsymbol{B} = \boldsymbol{USV^T} \tag{81}$$

$$\boldsymbol{U}_+ = \boldsymbol{U}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & det\boldsymbol{U}\end{bmatrix} \tag{82}$$

$$\boldsymbol{V}_{+} = \boldsymbol{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{V} \end{bmatrix} \tag{83}$$

$$\boldsymbol{A} = \boldsymbol{U}_{+}\boldsymbol{V}_{+}^{\boldsymbol{T}} \tag{84}$$

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

[47]

5 Complete program

6 Testing of star-tracker

[48]

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. 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.

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

[23] 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.

[24] C. C. Liebe, “Accuracy performance of star trackers-a tutorial,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 587–599, 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] G. Wahba, “A least squares estimate of satellite attitude,” *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965.

[44] 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.

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

[46] 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.

[47] 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.

[48] 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