**AALBORG UNIVERSITY**
**INSTITUTE OF ELECTRONIC SYSTEMS**
**DEPARTMENT OF CONTROL ENGINEERING**

Fredrik Bajers Vej 7 ▪ DK-9220 Aalborg Ø Phone +45 96 35 80 80

**TITLE:** Attitude Estimator for Rømer
**PROJECT PERIOD:** 10th semester, January 26th 2001 - June 7th 2001
**PROJECT GROUP:** IAS-1033

## Abstract

This project documents the estimator design, proposed for the attitude control subsystem of Rømer. Rømer requires fine pointing for observing stellar oscillations in nearby stars.

A detailed noise model of the Terma NEMO star tracker is derived, including optical distortion, centroiding errors and noise equivalent angle errors. In addition a noise model of the QRS11Pro gyroscope, and the model of the kinematics and dynamics of Rømer are derived.

Several Kalman filters are investigated, of which two are selected. A multi-rate Kalman filter and a steady-state Kalman filter. The multi-rate has the advantage of gyro bias estimation and 10Hz estimate availability. The steady-state filter requires less computation.

Algorithms for handling the delay of the attitude measurement are derived, of which the update correction algorithm is proposed. The algorithm analytically corrects for the propagated error, due to delays, when a measurement is received and incorporated in the filter.

The attitude model, noise models, and estimators are implemented in SIMULINK , and the error in pitch, yaw, and roll is investigated. The results are compared to the system requirements of the ACS. The results prove that the errors of the estimates are well within the requirements. The estimates of the steady-state Kalman filter used in fine mode have RMS values in the magnitude of arc seconds, while the requirements are in the magnitude of arc minutes.

It is concluded that the estimators are applicable for the Rømer ACS.

**GROUP MEMBERS:**
Dan Bhanderi

**SUPERVISOR:**
Thomas Bak

**NUMBER OF COPIES:** 8
**NUMBER OF PAGES:** 159

# Preface

This thesis is written by Dan Bhanderi, Group 1033, at Aalborg University (AAU), Denmark from January 26th 2001 to June 7th 2001.

The project is aimed for the attitude control team of the Danish Rømer satellite.

Figures and tables are numbered successively within each chapter, e.g. Figure 5.7 is the 7th figure in Chapter 5.

Literature references are denoted with the authors surname and the year of publishing in square brackets, [Wertz, 1978].

Abbreviations, symbols and definitions are given in the following pages.

Appendix A is a description of the quaternion notation and their algebra, which is used in this project.

AAU, Aalborg, June 7th 2001.

Dan Bhanderi

# Abbreviations, Definitions, and Symbols

The following abbreviations, definitions, and symbols are used in the report.

## Abbreviations

| | |
|---|---|
| A/D | Analog to Digital |
| ACS | Attitude Control System |
| CCD | Charged Coupled Device |
| CHU | Camera Head Unit |
| CoG | Center of Gravity |
| CoM | Center of Mass |
| DPU | Data Processing Unit |
| ECI | Earth Centered Inertial |
| FOV | Field of View |
| MONS | Measuring Oscillations in Nearby Stars |
| NEA | Noise Equivalent Angle |
| PSD | Power Spectrum Density |
| RMS | Root Mean Square |
| RSR | Rømer Standard Reference |
| RSS | Root Sum Square |
| SCB | Spacecraft Body |

# Definitions

Vectors are denoted with lower case bold: $\boldsymbol{v}$<br>
The elements of vector $\boldsymbol{v}_{\mathrm{i}}$ are denoted $v_{i,1}$, $v_{i,2}$, etc.<br>
The direction of the vector $\boldsymbol{v}$ is represented by the unit vector in that direction denoted as: $\hat{\boldsymbol{v}}$<br>
The length of vector $\boldsymbol{v}$ is denoted as: $|\boldsymbol{v}|$<br>
The estimate of vector $\boldsymbol{v}$ is denoted as: $\hat{\boldsymbol{v}}$[1]<br>
The working point of vector $\boldsymbol{v}$ is denoted as: $\bar{\boldsymbol{v}}$<br>
The small signal of vector $\boldsymbol{v}$ is denoted as: $\tilde{\boldsymbol{v}}$<br>
Matrices are denoted with uppercase bold: $\boldsymbol{M}$<br>
The element of the $i$th row and $j$th column of matrix $\boldsymbol{M}$ is denoted as: $\boldsymbol{M}_{i,j}$<br>
The $i$ by $i$ identity matrix is denoted as: $\mathbf{1}_{i \times i}$<br>
The $i$ by $j$ zero matrix is denoted as: $\mathbf{0}_{i \times j}$<br>
Cross product matrices are denoted as: $\mathbf{S}(\boldsymbol{v})$<br>
The transposed is denoted with a superscript T: $\boldsymbol{M}^{\mathrm{T}}$<br>
The rotation from frame $b$ to frame $s$ is represented by a rotation matrix as: $\boldsymbol{A}_b^s$<br>
Quaternions are denoted as: $\boldsymbol{q}$<br>
The scalar part of a quaternion $\boldsymbol{q}$ is denoted as: $q_4$<br>
The vector/complex part of a quaternion $\boldsymbol{q}$ is denoted as: $\boldsymbol{q}_{1:3}$<br>
The complex conjugated of $\boldsymbol{q}$ is denoted with an asterisk: $\boldsymbol{q}^*$<br>
Vector $\boldsymbol{v}$ in frame $b$ is denoted as: $\boldsymbol{v}^b$<br>
The time derivative of a vector $\boldsymbol{v}$ in frame $b$, given in frame $s$ is denoted as: $\left(\dot{\boldsymbol{v}}^b\right)^s$<br>
Functions resulting in a vector are denoted with lower case bold: $\mathbf{sin}(\boldsymbol{v})$<br>
Functions returning a matrix values are denoted in upper case bold: $\mathbf{S}(\boldsymbol{v})$<br>
Parenthesis $(\cdot)$ are used to enclose function parameters or dependencies as: $\boldsymbol{f}(\boldsymbol{x}, t)$<br>
The attitude matrix representing the rotation $\boldsymbol{q}$ is denoted as: $\boldsymbol{A}(\boldsymbol{q})$<br>
The a priori value of a vector $\boldsymbol{v}$ is denoted as: $\boldsymbol{v}^-$

[1] The $\hat{}$ operator is used for both unit vector and estimates to maintain notational agreement. The function of the operator should be clear by the context.

# Symbols

| | |
|---|---|
| $\boldsymbol{a}_i$ | Acceleration of particle |
| $d_{\mathrm{ca}}$ | Centroiding algorithm error period |
| $d_{\mathrm{opt}}$ | Residual optical error period |
| $\boldsymbol{F}$ | Linearized continuous system state matrix |
| $f$ | Focal length |
| $\boldsymbol{f}$ | Non-linear system function |
| $\boldsymbol{G}$ | Linearized continuous system input matrix |
| $\boldsymbol{H}$ | Output matrix |
| $\boldsymbol{h}$ | Non-linear system output function |
| $\boldsymbol{H}_{\mathrm{g}}$ | Rate update system output matrix |
| $\boldsymbol{J}$ | Moment of inertia |
| $\boldsymbol{K}$ | Kalman gain |
| $k_{\mathrm{sf}}$ | Gyro scale factor |
| $\boldsymbol{K}_\infty$ | Steady-state Kalman gain |
| $\boldsymbol{l}$ | Angular momentum |
| $m_{\mathrm{cut}}$ | Magnitude cut-off level |
| $m_{\mathrm{cut,high}}$ | Magnitude cut-off level in high rate level |
| $m_{\mathrm{cut,low}}$ | Magnitude cut-off level in low rate level |
| $m_{\mathrm{cut,mid}}$ | Magnitude cut-off level in medium rate level |
| $\boldsymbol{m}$ | Angular momentum of wheels |
| $m_i$ | Mass of particle |
| $n_a$ | Number of stars in the FOV |
| $\boldsymbol{n}ctrl$ | Control torque |
| $\boldsymbol{n}_{\mathrm{ext}}$ | Sum of external forces |
| $n_{\mathrm{pixels}}$ | Number of pixels per axis on the CCD |
| $n_s$ | Number of detectable stars in the FOV |
| $\boldsymbol{P}$ | State estimate covariance |
| $p_{\mathrm{ca}}$ | Centroiding algorithm error peak |
| $p_{\mathrm{opt}}$ | Residual optical error peak |
| $p_\epsilon$ | Perturbed projection factor |
| $\boldsymbol{P}_\infty$ | Steady-state estimate covariance |

| | |
|---|---|
| $\boldsymbol{Q}$ | System model noise strength |
| $\boldsymbol{q}$ | True attitude of the satellite |
| $\boldsymbol{q}_{\mathrm{m}}$ | Measured attitude |
| $\boldsymbol{q}_{\mathrm{st}}$ | Attitude of star tracker |
| $\boldsymbol{q}_{\epsilon}$ | Attitude estimation error |
| $\boldsymbol{q}_{\epsilon_{\mathrm{st}}}$ | Attitude error of star tracker measurement |
| $\boldsymbol{R}$ | System measurement noise strength |
| $\hat{\boldsymbol{r}}_{\mathrm{g}}$ | Gyro input axis |
| $\boldsymbol{r}_i$ | Position of particle |
| $\hat{\boldsymbol{r}}_{\mathrm{m}}$ | Intended input axis |
| $\boldsymbol{r}_{\mathrm{Star}_i}$ | Direction to the $i$'th star on the CCD |
| $\boldsymbol{s}_{\mathrm{Meas}_i}$ | Cartesian measured direction vector to the stars in the FOV |
| $\boldsymbol{s}_{\mathrm{True}_i}$ | Cartesian true direction vector to the stars in the FOV |
| $t$ | Time |
| $t_{\mathrm{exp}}$ | Camera exposure time |
| $\boldsymbol{u}$ | System input |
| $u_{\mathrm{g}}$ | Gyro voltage output |
| $\boldsymbol{v}$ | System measurement noise |
| $\boldsymbol{v}_i$ | Velocity of particle |
| $\boldsymbol{w}$ | System model noise |
| $w_{\mathrm{drift}}$ | Gyro drift noise |
| $w_{\mathrm{g}}$ | Gyro white noise |
| $\boldsymbol{x}$ | System state |
| $\boldsymbol{x}_{\mathrm{CCD}}$,$\boldsymbol{y}_{\mathrm{CCD}}$,$\boldsymbol{z}_{\mathrm{CCD}}$ | Axes of the CCD frame |
| $\boldsymbol{x}_{\mathrm{ECI}}$,$\boldsymbol{y}_{\mathrm{ECI}}$,$\boldsymbol{z}_{\mathrm{ECI}}$ | Axes of the ECI frame |
| $\boldsymbol{x}_{\mathrm{g}}$,$\boldsymbol{y}_{\mathrm{g}}$,$\boldsymbol{z}_{\mathrm{g}}$ | Axes of the gyro frame |
| $\boldsymbol{x}_{\mathrm{RSR}}$,$\boldsymbol{y}_{\mathrm{RSR}}$,$\boldsymbol{z}_{\mathrm{RSR}}$ | Axes of the RSR frame |
| $\boldsymbol{x}_{\mathrm{SCB}}$,$\boldsymbol{y}_{\mathrm{SCB}}$,$\boldsymbol{z}_{\mathrm{SCB}}$ | Axes of the SCB frame |
| $\boldsymbol{z}$ | System measurement vector |
| $\alpha_{\mathrm{FOV}}$ | Angular size of square FOV |
| $\boldsymbol{\beta}$ | Gyro bias |
| $\boldsymbol{\beta}_{\mathrm{m}}$ | Measured gyro bias |
| $\epsilon_{\mathrm{align}}$ | Gyro alignment error |
| $\epsilon_{\mathrm{bias}}$ | Gyro bias |
| $\boldsymbol{\epsilon}_{\mathrm{ca}_i}$ | Centroiding error |
| $\epsilon_{\mathrm{chrom}}$ | Chromatic aberration error |
| $\epsilon_{\mathrm{drift}}$ | Gyro drift |
| $\boldsymbol{\epsilon}_{\mathrm{nea}_i}$ | Error of noise equivalent angle |
| $\boldsymbol{\epsilon}_{\mathrm{opt}_i}$ | Residual optical error |
| $\epsilon_{\mathrm{RMS}}$ | 3-sigma RMS value of sinusoidal error |
| $\epsilon_{\mathrm{sf}}$,$\epsilon'_{\mathrm{sf}}$ | Scale factor errors |

| | |
|---|---|
| $\boldsymbol{\Gamma}$ | Discrete system input matrix |
| $\boldsymbol{\Omega}$ | Matrix of kinematic equation |
| $\boldsymbol{\omega}$ | True angular rate of the satellite |
| $\omega_{\mathrm{Earth}}$ | Angular rate of the Earth |
| $\omega_{\mathrm{g}}$ | Gyro measurement angular rate magnitude |
| $\boldsymbol{\omega}_{\mathrm{g}}$ | Gyro measurement angular rate |
| $\omega_{\mathrm{low,mid}}$ | Low to medium rate level threshold |
| $\boldsymbol{\omega}_{\mathrm{m}}$ | Measured angular rate |
| $\omega_{\mathrm{max}}$ | Worst case angular rate of stars on the CCD |
| $\omega_{\mathrm{mid,high}}$ | Medium to high rate level threshold |
| $\boldsymbol{\omega}_{\mathrm{st}}$ | Angular rate of star tracker |
| $\boldsymbol{\omega}_{\epsilon}$ | Rate estimation error |
| $\boldsymbol{\omega}_{\rho}$ | Horizontal and vertical angular rate of stars on the given in the CCD frame |
| $\phi$ | Roll |
| $\boldsymbol{\Phi}$ | Discrete system state matrix |
| $\psi$ | Yaw |
| $\boldsymbol{\rho}_{\mathrm{FOV}_i}$ | Angular distance to detectable stars in the FOV given in the ECI frame |
| $\boldsymbol{\rho}'_{\mathrm{FOV}_i}$ | Angular distance to all stars in the FOV given in the ECI frame |
| $\boldsymbol{\rho}_{\mathrm{M}_i}$ | Angular distance to stars after vertical and horizontal modulo shifting |
| $\boldsymbol{\rho}_{\mathrm{Meas}_i}$ | Measured angular distance to detectable stars in the FOV in the CCD frame |
| $\boldsymbol{\rho}_{\mathrm{P}_i}$ | Angular distance to stars after vertical and horizontal rotation |
| $\boldsymbol{\rho}_{\mathrm{Star}_i}$ | Angular distance to detectable stars in the FOV given in the CCD frame |
| $\rho\boldsymbol{x}$ | Reduced state vector |
| $\rho\boldsymbol{z}$ | Reduced measurement vector |
| $\sigma_{\mathrm{ca}}$ | 1-sigma value of the centroiding algorithm error |
| $\sigma_{\mathrm{drift}}$ | Gyro drift deviation |
| $\sigma_{\mathrm{g}}$ | Gyro white noise deviation |
| $\sigma_{\mathrm{st}}$ | 1-sigma value of the NEA |
| $\sigma_{\mathrm{st,high}}$ | 1-sigma value of the NEA in high rate level |
| $\sigma_{\mathrm{st,low}}$ | 1-sigma value of the NEA in low rate level |
| $\sigma_{\mathrm{st,mid}}$ | 1-sigma value of the NEA in medium rate level |
| $\theta$ | Pitch |
| $\xi_{\mathrm{err}_i}$ | Single angular direction error |
| $\xi_{\mathrm{Meas}_i}$ | Measured single angular direction to the stars on the CCD |
| $\xi_{\mathrm{Star}_i}$ | Single angular direction to the stars on the CCD |

# Contents

# List of Figures

# Chapter 1

# Introduction

Following the successful mission of The Ørsted Satellite, Denmark is currently developing a new satellite named Rømer.

Rømer's main payload is the MONS (Measuring Oscillations in Nearby Stars) telescope, which will measure stellar oscillation. From these measurements improved stellar models can be achieved. This project aims to find an attitude estimator Rømer.

## 1.1 Project Frame

Aalborg University (AAU) is in cooperation with The Danish Technical University responsible for the development of the attitude control system (ACS). The ACS has two main parts:

- Attitude Estimation
- Attitude Control

The attitude estimation algorithm provides an estimate of the satellite's current attitude to the controller. The controller uses this information to acquire inertial pointing of the satellite, which is necessary in order to point the MONS telescope towards a target star.

This project is concerned only with the attitude estimation part of the Rømer ACS.

## 1.2 Initial Problems

This project spawns from the need of an attitude estimator for Rømer. An estimation algorithm must be chosen, and an implementation provided in SIMULINK, which can be used in conjunction with the ACS simulation. In order to demonstrate the functionality of the attitude determination algorithm, sensors must be modeled with all relevant noise factors.

It is the goal of the attitude estimator to minimize the error between the true attitude and the noise inflicted attitude, acquired from sensors. In addition it is known that delays in sensor hardware requires an analysis of synchronization problems. This lead to the problem formulation of this project:

> The goal of this project is to develop an attitude estimator, to be used with the ACS of Rømer. The algorithm must be implemented in SIMULINK, for use with the ACS simulation. The simulation must include a detailed modeling of the sensor hardware intended for Rømer. The performance of the algorithm must be within the requirements of the attitude controller. In addition it is of interest to investigate the feasibility of performance, and choose an algorithm through a trade-off between performance and complexity of the algorithm. The delay in sensor hardware must be analyzed, and handled appropriately.

## 1.3 Solution Strategy

The attitude estimator algorithm and simulation model, is acquired through an investigation in the following areas:

- Analysis of sensor hardware intended for Rømer.
- Noise modeling and SIMULINK implementation of sensor hardware.
- Analysis of different estimator designs.
- Trade-off between performance and complexity of estimator.
- Derivation of synchronization algorithm.
- Implementation of algorithm in SIMULINK.
- Performance verification through simulation.

# Part I

# System

Chapter 2

# System Description

This chapter describes the Rømer system components, and the requirements to the ACS subsystem. A brief summary of the mission is given in the following.

## 2.1 Rømer Mission

The mission of Rømer originally contained to payloads, which were the MONS Telescope and the Micro Ballerina WATCH instruments. However it was chosen to focus on a single payload, hence the Micro Ballerina payload was discarded.

The satellite will be launched into the Molniya Orbit, which is an elliptical orbit with an apogee height of $39867$km and perigee height of $500$km.

The MONS Telescope is to be pointed at a number of target stars, and observe their oscillations. This data can be used to study the structure and evolution of stars, with more detail than has been possible so far.

The satellite is expected to be launched in the first half of 2004.

## 2.2 Hardware

Figure 2.1 shows the Rømer Satellite seen from the nominally sunlit side. The Sun Protection Lid, protects the MONS Telescope from sunlight. Solar panels are mounted on the two sides facing the Sun, in order to supply the power subsystem with power. The Field Monitor, is part of the MONS mission equipment, and supplies data necessary for analyzing the data from the MONS Telescope.

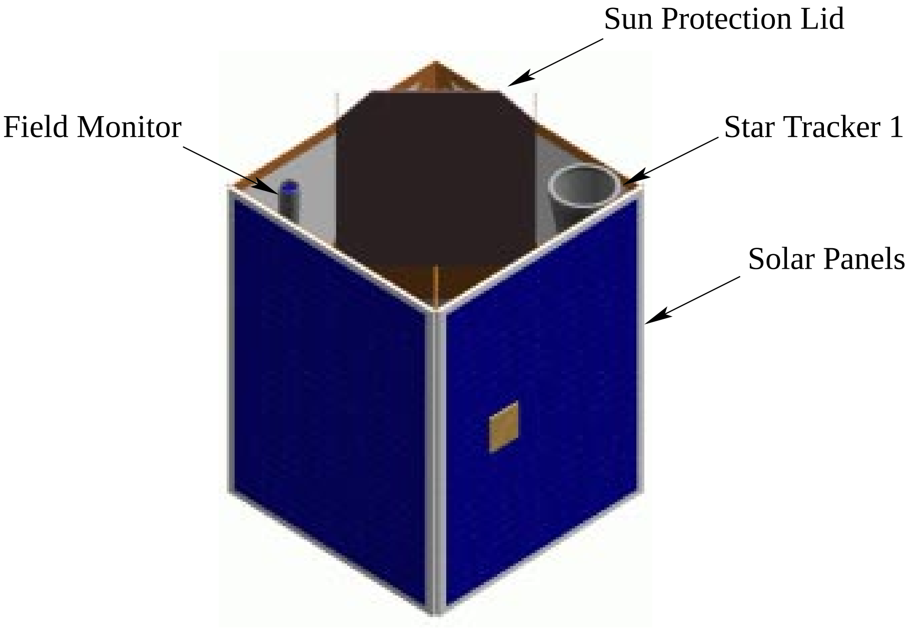


**Figure 2.1**: *Sunlit side of Rømer, [DSRI, 2001].*

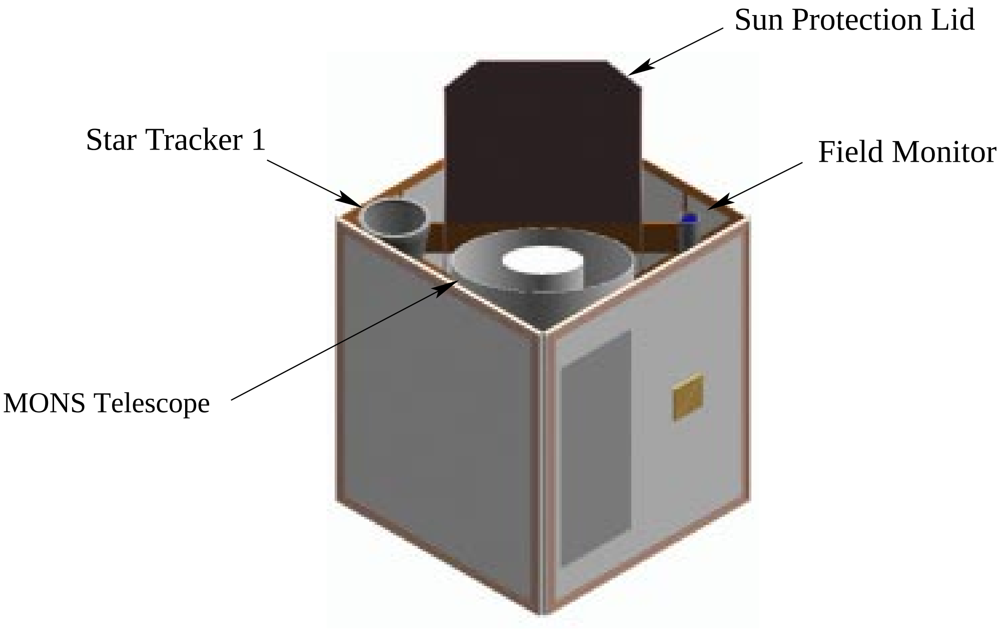


**Figure 2.2**: *Shadow side of Rømer, [DSRI, 2001].*

In Figure 2.2 Rømer is seen from the opposite side, the shadow side. The MONS telescope is revealed, mounted by the Sun Protection Lid.

At the opposite corner of the Field of View Camera, a star tracker placed, which is used for inertial attitude measurements, which is seen in Figures 2.1 and 2.2. A second star tracker is facing the opposite direction, visible in Figure 2.3, which reveals the inside of Rømer.

Figure 2.3 shows the four momentum wheels inside the Rømer, which are mounted in a tetrahedron configuration. This configuration allows the system to survive a single failure of a wheel. On each of the four wheels, a gyro is mounted, for measuring the angular velocity of the satellite.

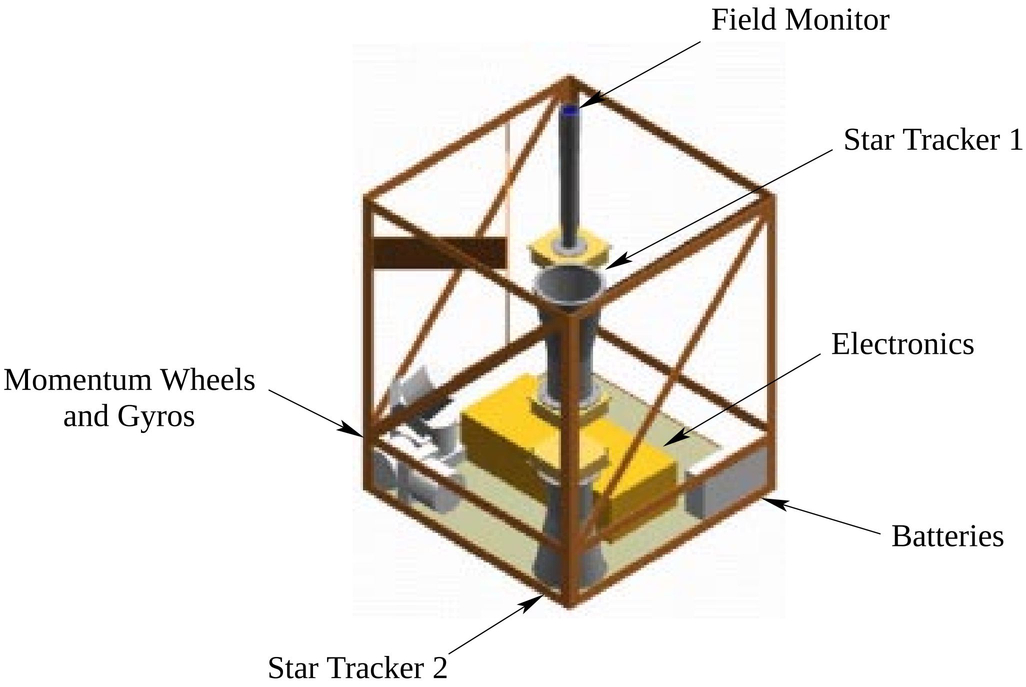


**Figure 2.3**: *Inside view of Rømer, [DSRI, 2001].*

In addition to the momentum wheels, magnetic coils are used as magnetorquers. These are used as for momentum unloading of the wheels, keeping them at nominal velocities. Sun sensors are used as secondary sensing devices, which can be used in the absence of star tracker measurements.

It is intended that only the star trackers and rate gyros are used in the fine pointing mode of Rømer, hence the estimator of this project will focus on the use of these two hardware arrays.

An exploded view of Rømer can be found in Annex I. The annex also contains the physical dimensions and mass properties of Rømer. In addition the electrical layout of the satellite is given. Note in the electrical layout (Figure I.4), that the calculation of the attitude from the CHU images, is executed on the CDH, and not on separate DPUs.

Since this project is mainly concerned with the star trackers and gyros used in the attitude estimator, hardware descriptions of these are given in the following.

## Star Tracker

Star trackers are used to acquire a measurement of attitude. This is done by tracking the stars in the camera's field of view (FOV). By identifying the stars in the FOV, the boresight of the camera is known, and thereby the attitude of the object, on which the star tracker is mounted, can be determined.

Figure 2.4 is an image of a star tracker from Terma . The star tracker has two main parts. The camera head unit (CHU) and the data processing unit (DPU). The star tracker used on Rømer is the NEMO star tracker from Terma. The data sheet of the star tracker is enclosed as Annex II.

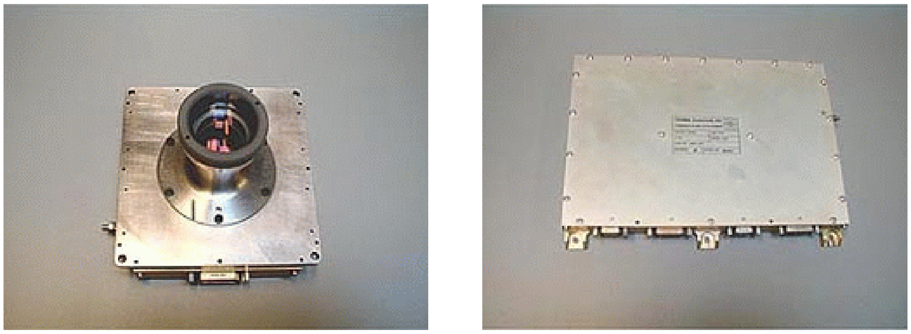


**Figure 2.4**: *Image of star tracker hardware which consists of a CHU (left) and a DPU (right).*

It is the purpose of the CHU to provide a digitized picture of the FOV, so the DPU can process the image in order to find the attitude. The CDU uses a lens to project the FOV onto a charged coupled device (CCD). In order to protect the lens from stray light, e.g. from the Sun or Earth, a baffle is mounted on top of the lens.

The output of the CCD is an analog signal which is converted by an A/D converter and sent to the DPU for image processing.

The DPU uses pattern recognition in order to identify the stars in the image. The stars are matched using a set of parameters, which have been pre-calculated for a selected set of stars stored in an integrated star catalogue. Upon calculating the parameters for the stars in the image, the star catalogue must return a star in the catalogue which has the same parameters. There should be only one solution in the star catalogue. Once the stars in the FOV are recognized, the star tracker uses these stars to track the attitude.

The parameters used in the star catalogue depend on the algorithm used for the pattern recognition. One set of parameters could include the distances and angular separation to the two brightest adjacent stars. It is important that the algorithm is robust to missing or extra stars in the FOV.

When a star is projected onto the CCD, a coordinate set of the location on the CCD can be obtained. If the star is focused on a single pixel, the coordinate set is quantized by the number of pixels on the CCD. However, if the star is slightly defocused, the light will affect several pixels within a certain radius. Using a centroid algorithm on the defocused image, will allow the calculation of a coordinate set, with a resolution in the sub-pixel range. This means that by defocusing the picture, the star location on the CCD, is calculated with a precision that lies within a fraction of a pixel.

## Gyro

Gyros are used to measure the angular velocity about a given axis, named the input axis of the gyro. In order to measure a three dimensional angular velocity of Rømer, a minimum of three gyros are required. The gyros used are rate sensing devices, which measure the angular velocity, not to be confused with rate integrating gyros, which measure angular displacement.

Figure 2.5 shows the QRS11Pro gyro from Systron, used on Rømer. The data sheet of the rate gyro is enclosed as Annex III.

**Figure 2.5**: *The QRS11Pro gyro used on Rømer, [SYSTRON, 2001].*

The QRS11Pro gyro uses piezoelectric quartz for measuring rotational motion about the input axis. The quartz is shaped as a dual headed fork. The vibration of one end of the fork makes angular motion measurable at the other end, due to the Coriolis effect, [SYSTRON, 2001].

The gyro is supplied with an A/D converter. The external electronics of the gyro, acts as a 2Hz filter on the measurement and the converter quantizes the value with a 12 bit resolution.

Chapter 3

# System Definitions

This chapter describes the pre-definitions of the system. This includes definition of operating modes and coordinate systems used in the estimator.

## 3.1 Coordinate Systems

The ACS points the satellite at a inertial fixed star, using sensors on the spacecraft. The estimator must supply the controller with an estimate of the rotation from the spacecraft body wrt. a star fixed frame. The geometry of the satellite is given in the Rømer Reference Frame. Hence three coordinate systems are defined:

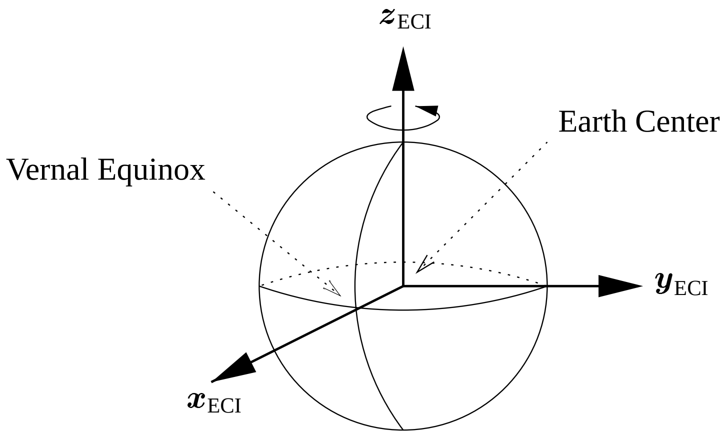
- Earth Centered Inertial (ECI) Coordinate System
- Spacecraft Body (SCB) Coordinate System
- Rømer Standard Reference (RSR) Coordinate System

The coordinate systems are defined in the following two sections.

### ECI Coordinate System

The ECI coordinate system is the inertial frame. It is star fixed, and the orientation of the spacecraft is controlled wrt. to this frame. The frame is shown in Figure 3.1.

The ECI coordinate system consists of the $\boldsymbol{x}_{\mathrm{ECI}}$, $\boldsymbol{y}_{\mathrm{ECI}}$, and $\boldsymbol{z}_{\mathrm{ECI}}$ axes. The origin is at the geocenter of the Earth. The $\boldsymbol{x}_{\mathrm{ECI}}\boldsymbol{y}_{\mathrm{ECI}}$ plane coincide with the equatorial plane of Earth. The $\boldsymbol{x}_{\mathrm{ECI}}$ axis is in the direction of the Sun at vernal equinox. The $\boldsymbol{z}_{\mathrm{ECI}}$ axis points



$\boldsymbol{z}_{\text{ECI}}$
Earth Center
Vernal Equinox
$\boldsymbol{y}_{\text{ECI}}$
$\boldsymbol{x}_{\text{ECI}}$

**Figure 3.1**: *Definition of the Earth Centered Inertial Coordinate System.*

towards the North Pole, and is a normal to the $\boldsymbol{x}_{\text{ECI}}\boldsymbol{y}_{\text{ECI}}$ plane. The $\boldsymbol{y}_{\text{ECI}}$ axis completes the right-handed system.

## SCB Coordinate System

The SCB coordinate system is used for attitude control. The coordinate system consists of the $\boldsymbol{x}_{\text{SCB}}$, $\boldsymbol{y}_{\text{SCB}}$, and $\boldsymbol{z}_{\text{SCB}}$ axes. It is centered at the CoM of the satellite, and the axes are aligned with the satellite's principal axes. This definition simplifies the description of the satellite dynamics.

## RSR Coordinate System

The RSR frame is used for the geometry of the satellite. All equipment locations are given in this frame. The origin of the frame is at the center of the adapter ring. The adapter ring is located at the bottom of the satellite, and is used for mounting the satellite on the launcher spacecraft. The coordinate system consists of the $\boldsymbol{x}_{\text{RSR}}$, $\boldsymbol{y}_{\text{RSR}}$, and the $\boldsymbol{z}_{\text{RSR}}$ axes. The $\boldsymbol{x}_{\text{RSR}}$ axis points in the direction of the nominal sun facing side. The $\boldsymbol{z}_{\text{RSR}}$ axis is in the direction of the MONS Telescope boresight, and perpendicular to the plane spanned by the adapter ring. The $\boldsymbol{y}_{\text{RSR}}$ completes the right handed system.

From an ACS point of view, the frame of interest is the SCB frame wrt. the inertial frame. The orientation of e.g. the star trackers can be calculated in the SCB frame, given the orientation in the RSR frame and the constant orientation of the RSR frame wrt. the SCB frame. Throughout this project the calculation of the orientation from the RSR frame to SCB frame is left out, and the orientations are simply assumed to be given in the SCB frame.

## 3.2 Operation Modes

The spacecraft operates in different modes, which must be defined from launcher release to end-of-life. Different ACS requirements exist in each mode, and the modes ensure that a minimum life support is available in case of a failure. The operation modes and transitions between modes, is defined in [Bak et al., 2001]. The operation modes of the attitude estimator is shown in Figure 3.2. A description of the modes is given in Table 3.1.

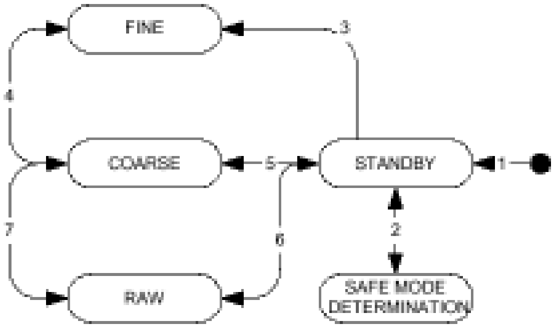


**Figure 3.2**: *Rømer attitude estimator operation modes.*

| Mode | Estimator Function |
|---|---|
| Standby | No estimation. |
| Safe Mode | Coarse Sun pointing attitude estimation (two axis) based on raw sun sensor and gyro measurements. |
| Coarse | Coarse three-axis attitude estimation, using on sun sensors, gyros, magnetorquers, and star trackers, if available. |
| Fine | Fine three-axis attitude and rate estimation, using gyros and star trackers. |

**Table 3.1**: *Operation modes of the Rømer Attitude Estimator.*

Figure 3.2 shows the transitions between each mode with a number. The description of each transition is as follows:

1. The system enters standby mode when initiated.
2. Safe mode is entered when reliable sun sensor and/or gyro measurements are available.

3. The estimator enters fine mode when when the error state covariance is below a predefined threshold.

4. Transition back to coarse mode from fine mode, happens when the error state covariance increases above the predefined threshold. Fine pointing is re-entered if the error state covariance decreases below the threshold.

5. Transition to standby mode occurs when three axis attitude determination is impossible, and transition back to coarse mode occurs when three axis attitude determination becomes available again.

6. Raw mode is entered when raw processing of data is desired.

7. Transition from coarse to raw mode and back is possible.

The estimator in this project is focused on the fine mode, where attitude estimation is acquired using star trackers and gyros. The covariance of the state error is tracked, in order to shift into coarse mode, in case the covariance exceeds the pre-defined threshold.

Chapter 4

# System Requirements

The performance of the estimator is evaluated from the controller requirements. This section summarizes the requirements of the satellite attitude controller in fine and coarse mode, stated in [Bak et al., 2001]. These requirements are set by the science requirements of the MONS Telescope pointing precision.

## 4.1 Coarse Pointing Requirements

In coarse pointing mode sun acquisition is done within 15 degrees accuracy. Re-orientation must be possible with a performance, which allows a $180^\circ$ re-orientation within ten minutes.

## 4.2 Fine Pointing Requirements

The requirements of the fine pointing are expressed in terms of absolute pointing and relative pointing. Absolute pointing is the actual pointing towards the target boresight axis of the MONS Telescope. Relative pointing specifies the pointing precision over a time interval. The pointing drift error is used to express the requirements of the MONS Telescope's long term drift.

The requirements are:

- The absolute pointing error of the MONS Telescope boresight axis is less than $2$am in pitch and yaw, $60$am in roll.

- The relative pointing error is less than $0.8$am over each $1$s interval, $0.35$am over each $0.1$s interval, in pitch and yaw. In roll it is less than $2$am over $60$s interval.
- The pointing drift error in pitch and yaw is less than:
  - $2.4$am over each time interval larger than $100$s.
  - $1.5$am over each $100$s interval.
  - $1.8$am over each $10$s interval.

Note that all numbers are $95\%$ confidence. The fine pointing performance expressed as a power spectrum density (PSD) function is given in Figure 4.1.

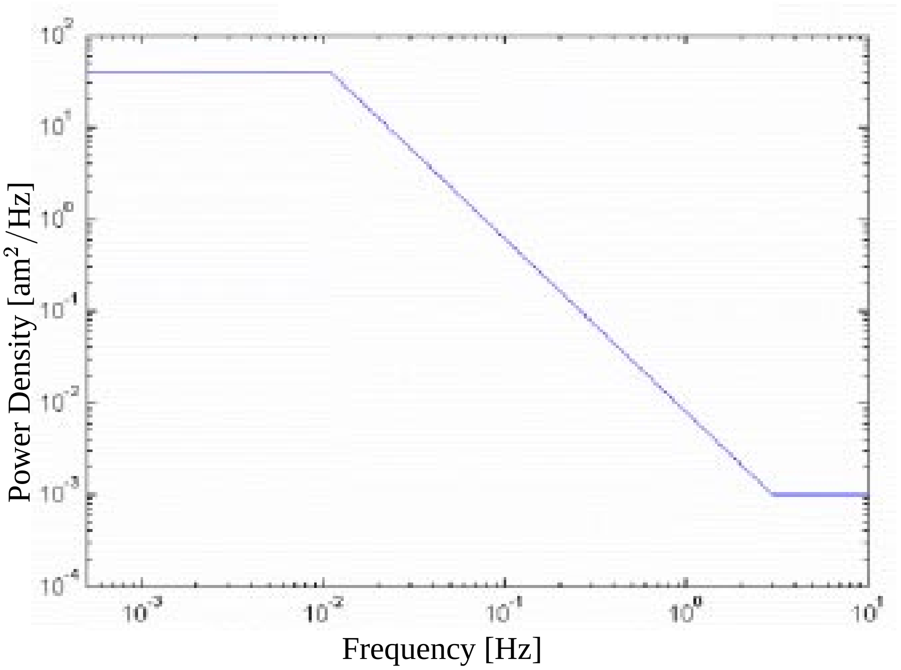


**Figure 4.1**: *Relative pointing error requirements as a function of frequency, [Noteborn, 2001].*

# Part II

# Modeling

Chapter 5

# Star Tracker Model

This chapter describes the model used for the star trackers on Rømer. The model derived in this chapter is based on the work in [Bayard, 1996].

The model is based on simulating the star tracker on pixel level, by projecting a simulated star image onto the CCD. Optical distortion is added as function of the position in the FOV, and the precision of the centroid algorithm is altered depending on the star centers on the pixels. The effect of the unmodeled quantities are added as a noise equivalent angle (NEA). This noise is assumed to be white Gaussian noise.

## 5.1 Outline and Definitions

The model does not include a star catalogue. Instead a known set of stars are selected, and shifted into the FOV, when the camera is rotated. Hence stars leaving the FOV to the right, re-enter on the left. Projecting the star image onto the CCD gives a set coordinates in the CCD frame, which is defined as follows:

- The $\boldsymbol{x}_{\mathrm{CCD}}$ axis is in the increasing horizontal pixel readout direction of the CCD.
- The $\boldsymbol{y}_{\mathrm{CCD}}$ axis is in the increasing vertical pixel readout direction of the CCD.
- The $\boldsymbol{z}_{\mathrm{CCD}}$ forms the right handed system, and is in the direction of the boresight.

The CCD frame is shown in Figure 5.1. As seen in the figure, the angles $\theta$ (pitch), $\psi$ (yaw) and $\phi$ (roll) represent the rotations about the $\boldsymbol{x}_{\mathrm{CCD}}$, $\boldsymbol{y}_{\mathrm{CCD}}$ and $\boldsymbol{z}_{\mathrm{CCD}}$ axes, respectively. The vector $\boldsymbol{r}_{\mathrm{Star}_i}$ is the vector from the center of the CCD to the projection of the $i$'th star onto the CCD.

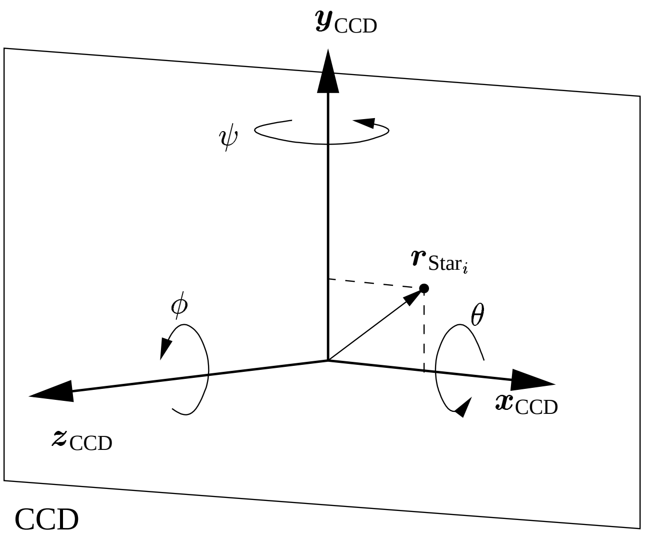

$\boldsymbol{y}_{\text{CCD}}$
$\psi$
$\boldsymbol{r}_{\text{Star}_i}$
$\phi$
$\theta$
$\boldsymbol{x}_{\text{CCD}}$
$\boldsymbol{z}_{\text{CCD}}$
CCD

**Figure 5.1**: *The CCD frame.*

In Figure 5.2 a star is projected onto the CCD using a standard pinhole camera model. Using this model, the direction to the $i$'th star can be found from the CCD readouts, which are the elements of the position vector of the star on the CCD. The direction of the star is given by the vector $\boldsymbol{\rho}_{\text{Star}_i}$, containing the horizontal and vertical angular distances to the $i$'th star.

From Figure 5.2 it is seen that the direction of the $i$'th star can be found by

$$\rho_{\text{Star}_i,1} = \tan^{-1}\left(\frac{r_{\text{Star}_i,1}}{f}\right) \tag{5.1}$$

$$\rho_{\text{Star}_i,2} = \tan^{-1}\left(\frac{r_{\text{Star}_i,2}}{f}\right) \tag{5.2}$$

given the horizontal readout $r_{\text{Star}_i,1}$ and vertical readout $r_{\text{Star}_i,2}$ from the CCD, and the focal length $f$. The focal length is the distance from the pinhole to the image plane, which is the CCD.

Based on a simulated attitude of the star tracker, the directions to the stars in the catalogue can be calculated. The objective of the model is to calculate the perturbed directions to the stars, caused by the major noise sources in the star tracker. Given the true and perturbed directions to the stars, an error attitude can be calculated. Adding this attitude error to the true attitude of the satellite, yields the simulated measurement acquired from a star tracker. An illustration of the modeling principle of the star tracker is shown in Figure 5.3.

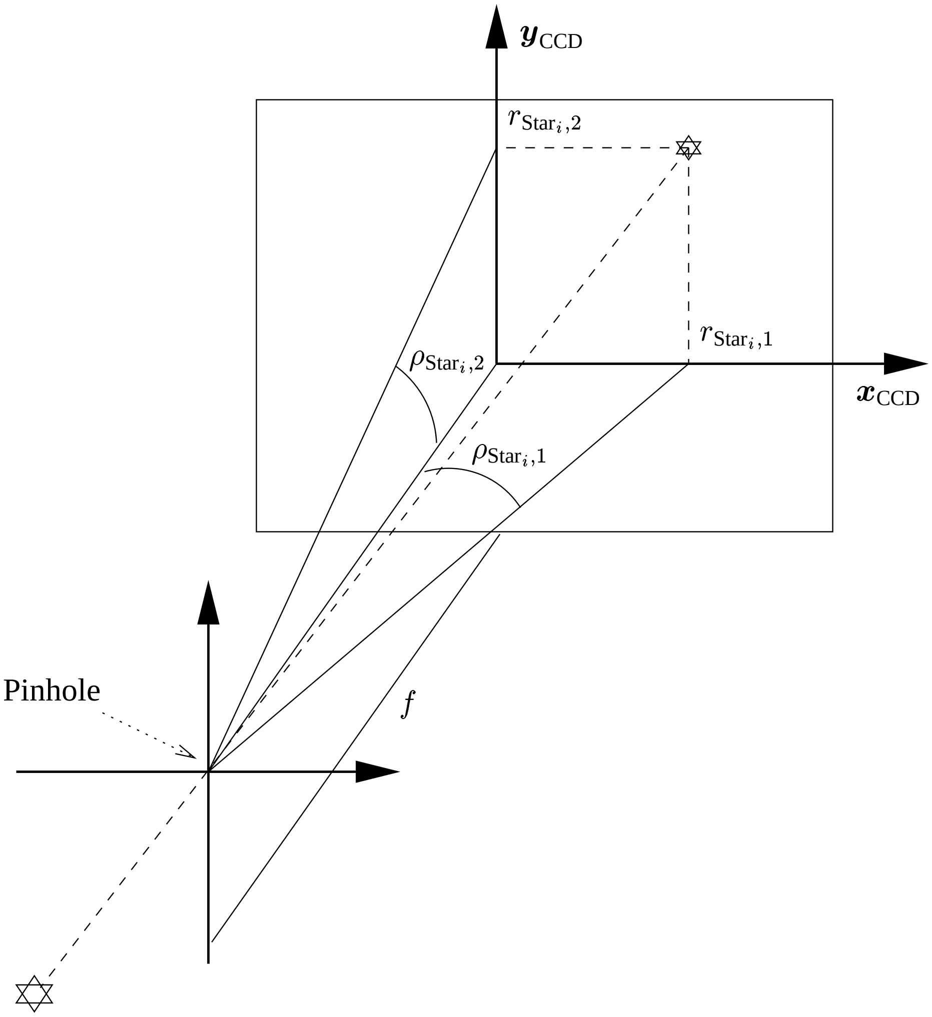


**Figure 5.2**: *The projection of a star onto the CCD using a standard pinhole camera model.*

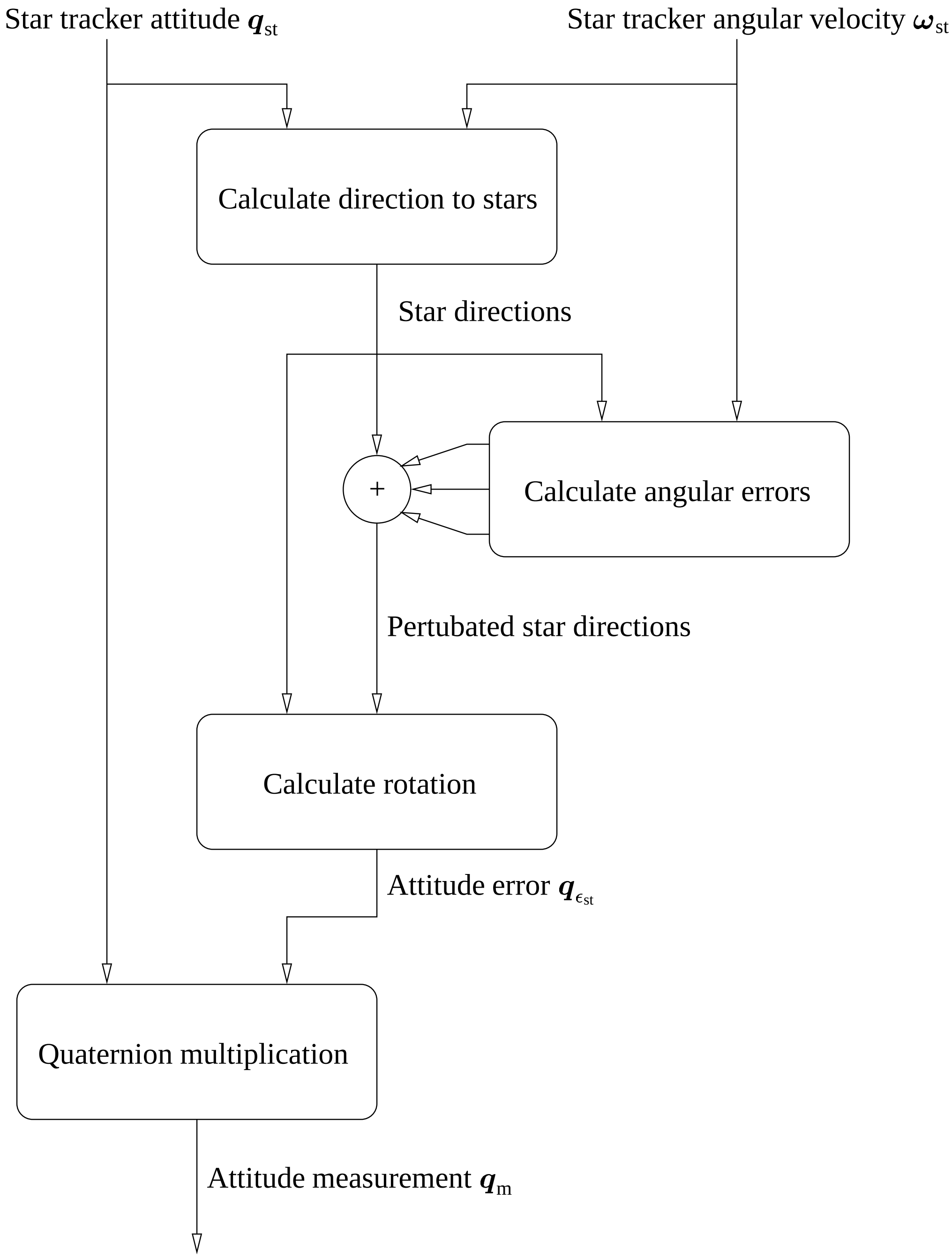

Star tracker attitude $\boldsymbol{q}_{\mathrm{st}}$
Star tracker angular velocity $\boldsymbol{\omega}_{\mathrm{st}}$
Calculate direction to stars
Star directions
+
Calculate angular errors
Pertubated star directions
Calculate rotation
Attitude error $\boldsymbol{q}_{\epsilon_{\mathrm{st}}}$
Quaternion multiplication
Attitude measurement $\boldsymbol{q}_{\mathrm{m}}$

**Figure 5.3**: *Illustration of modeling principle.*

## 5.2 Star Tracker Modeling Algorithm

The algorithm for modeling the star tracker follows the principle outlined in Figure 5.3. There are three major noise sources which influence the measurement:

- Star availability
- Centroiding algorithm performance
- Residual optical distortion

In addition, the remaining noise sources are added as white noise. This white noise is added as NEA, which is white noise added to the angle of the star direction. The star availability depends on the angular velocity of the satellite, since the dimmer stars will be undetectable as the angular velocity of the star tracker increases. The steps of the model are

1. Calculate the direction of the stars
2. Select a subset of stars based on angular velocity
3. Calculate centroiding algorithm performance
4. Calculate residual optical error
5. Add NEA
6. Calculate noise equivalent measurement error

The algorithm is based on a set of parameters which are calculated in the initialization of the algorithm. The initialization is described after the algorithm has been derived.

### Stars in the FOV

The first step of the model is to calculate the direction of the stars in the FOV. In order to avoid the modeling of a real star catalogue, a number of fictitious stars are placed in the FOV. When the camera changes orientation, the stars are moved in the FOV, and stars leaving in one side, enters at the opposite side. This means that the star identification, and the delays associated with it, are not included in the model at this point.

The set of stars in the FOV are represented by the vectors

$$\boldsymbol{\rho'}_{\mathrm{FOV}_i} = \begin{bmatrix} \rho'_{\mathrm{FOV}_i,1} \\ \rho'_{\mathrm{FOV}_i,2} \end{bmatrix} \qquad i = 1, 2, ..., n_{\mathrm{as}} \tag{5.3}$$

where $\rho'_{\mathrm{FOV}_i,1}$ and $\rho'_{\mathrm{FOV}_i,2}$ are the horizontal and vertical angular distances, respectively, given in an inertial frame. The number of all stars in the FOV is denoted $n_{\mathrm{as}}$. The number of stars used in the model varies with the angular velocity of the star tracker. The stars in Equation 5.3 are associated with a parameter representing the magnitude of the star. When the star tracker rotates with a given angular velocity, the stars with magnitude above a certain threshold are undetectable by the star tracker. Note that the magnitude of a star is inverse proportional with its brightness. Since some stars are undetectable, the stars used in the model are selected from the set in Equation 5.3 and re-indexed. This gives a new subset of stars, denoted as

$$\boldsymbol{\rho}_{\mathrm{FOV}_i} = \begin{bmatrix} \rho_{\mathrm{FOV}_i,1} \\ \rho_{\mathrm{FOV}_i,2} \end{bmatrix} \qquad i = 1, 2, ..., n_{\mathrm{s}} \tag{5.4}$$

where $n_{\mathrm{s}}$ denotes the number of stars used in the model, being a function of the star tracker angular velocity. The selection must be carried out at every sample time of the model, since the angular velocity varies in time.

To minimize the complexity of the model, the subset of stars are based on discrete values of the angular velocity. The performance of the star tracker is based on the angular velocity which is divided into three levels: low, medium and high. When the angular velocity is low, the subset of stars in Equation 5.4 is the full set of stars in the FOV of Equation 5.3. The number of stars in the subset, decreases as the angular velocity changes from low to medium, and again when changing from medium to high.

Upon selection of the subset of stars, the direction of the stars in the CCD frame, must be calculated. Due to the change in attitude of the satellite, the direction of the stars in the CCD frame is different than in the inertial frame. Given the attitude quaternion of the star tracker $\boldsymbol{q}_{\mathrm{st}}$, which varies with the attitude of the satellite, the equivalent 1-2-3 Euler angles $\theta$, $\psi$ and $\phi$ can be found by

$$\theta = \mathrm{atan2}\left(\frac{-2\left[q_1q_2 - q_3q_4\right]}{\cos\left(\phi\right)}, \frac{q_1^2 - q_2^2 - q_3^2 + q_4^2}{\cos\left(\phi\right)}\right) \tag{5.5}$$

$$\psi = \mathrm{atan2}\left(\frac{-2\left[q_3q_3 - q_1q_4\right]}{\cos\left(\phi\right)}, \frac{-q_1^2 - q_2^2 + q_3^2 + q_4^2}{\cos\left(\phi\right)}\right) \tag{5.6}$$

$$\phi = \mathrm{atan2}\left(2\left[q_1q_3 + q_2q_4\right], \cos\left(\phi\right)\right) \tag{5.7}$$

where

$$\cos\left(\phi\right) = \sqrt{\left[q_1^2 - q_2^2 - q_3^2 + q_4^2\right]^2 + 2^2\left[q_1q_2 - q_3q_4\right]^2} \tag{5.8}$$

The derivation of the above equations can be found in Appendix A.

Recalling Figure 5.1 and Figure 5.2, the angular directions to the $i$'th star in the CCD frame are changed by the pitch and yaw of the satellite attitude, written as

$$\rho_{\mathrm{P}_i,1} = \rho_{\mathrm{FOV}_i,1} + \psi \tag{5.9}$$
$$\rho_{\mathrm{P}_i,2} = \rho_{\mathrm{FOV}_i,2} + \theta \tag{5.10}$$

In order to keep the nominal set of stars in the FOV, the star directions are modulo shifted by the size of the FOV

$$\rho_{\mathrm{M}_i,1} = \mathrm{mod}_{\alpha_{\mathrm{FOV}}}\left(\rho_{\mathrm{P}_i,1} + \frac{\alpha_{\mathrm{FOV}}}{2}\right) - \frac{\alpha_{\mathrm{FOV}}}{2} \tag{5.11}$$
$$\rho_{\mathrm{M}_i,2} = \mathrm{mod}_{\alpha_{\mathrm{FOV}}}\left(\rho_{\mathrm{P}_i,2} + \frac{\alpha_{\mathrm{FOV}}}{2}\right) - \frac{\alpha_{\mathrm{FOV}}}{2} \tag{5.12}$$

Finally the stars on the CCD are rotated by the roll angle $\phi$, written as

$$\boldsymbol{\rho}_{\mathrm{Star}_i} = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \rho_{\mathrm{M}_i,1} \\ \rho_{\mathrm{M}_i,2} \end{bmatrix} \tag{5.13}$$

Given the available stars of the FOV in the CCD frame, the noise of the star tracker is calculated.

## Centroiding Algorithm Performance

The centroiding algorithm provides an estimate of the star location on the CCD, which has an accuracy that is better than the distance between the pixels. This is possible when the star is defocused, and hits more than a single pixel.

The performance of the centroiding algorithm, is degraded when the center of the defocused star circle is not centered on or between pixels. Hence it is a periodic function of the position on the CCD, with the pixel spacing as period. It is assumed that the error of the algorithm is zero in best case, shown in Figure 5.4.

The worst case performance must be known, and is a fraction of a pixel. When a star image on the CCD moves between pixels, the noise is modeled as a sine function, assuming that the performance is worst case when the star center is between the images in Figure 5.4, which generates a non-symmetrical pattern, illustrated in Figure 5.5.

It is seen from Figures 5.4 and 5.5 that the difference in performance is the loss of a symmetrical pattern as the center of the defocused star approaches the worst case position on a pixel.

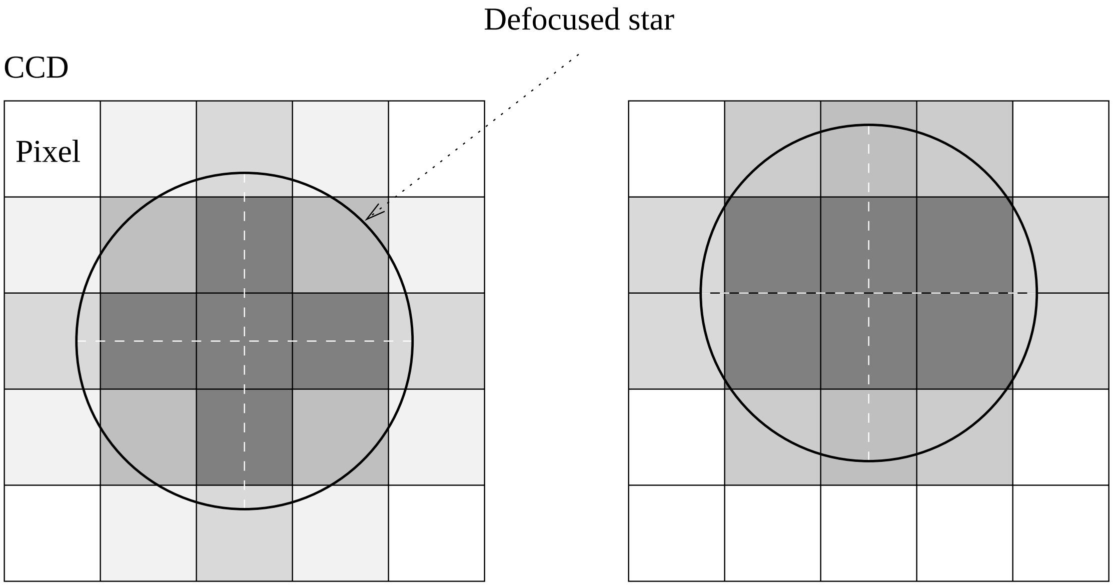


**Figure 5.4**: *Illustration of best case performance of the centroiding algorithm, when centered on (left) or between (right) a pixel.*

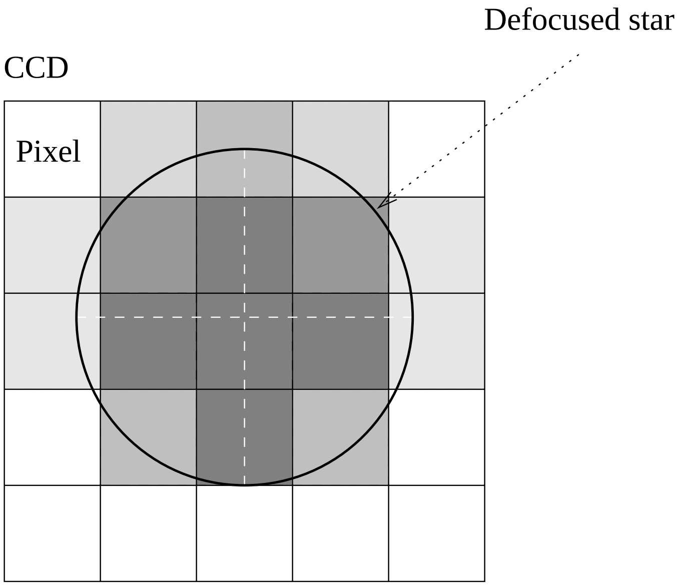


**Figure 5.5**: *Illustration of the worst case performance of the centroiding algorithm.*

Assuming equal pixel spacing in both horizontal and vertical direction, the periodic function of the noise in the star position on the CCD, $\boldsymbol{e}_{\text{ca}}$, is given by

$$\boldsymbol{e}_{\text{ca}_i} = p_{\text{ca}}\mathbf{sin}\left(\frac{2\pi\boldsymbol{\rho}_{\text{Star}_i}}{d_{\text{ca}}}\right) \tag{5.14}$$

where $p_{\text{ca}}$ is the worst case performance (peak) of the centroiding algorithm, and $d_{\text{ca}}$ is the period of the centroiding error, which is set to the pixel spacing in radians. Making the noise function periodic with the pixel space in radians instead of the pixel space on the CCD is not exact, but removes the need of knowing the focal length of the star tracker. If the error is calculated on the CCD instead of in the angular direction, the focal length is needed in order to add the noise to the vector $\boldsymbol{\rho}_{\text{Star}_i}$ with angular elements. This is seen from Equations 5.1 and 5.2.

The centroiding error added to $\rho_{\text{Star}_i,1}$ is plotted as a function of $\rho_{\text{Star}_i,1}$ and $\rho_{\text{Star}_i,2}$ in Figure 5.6.

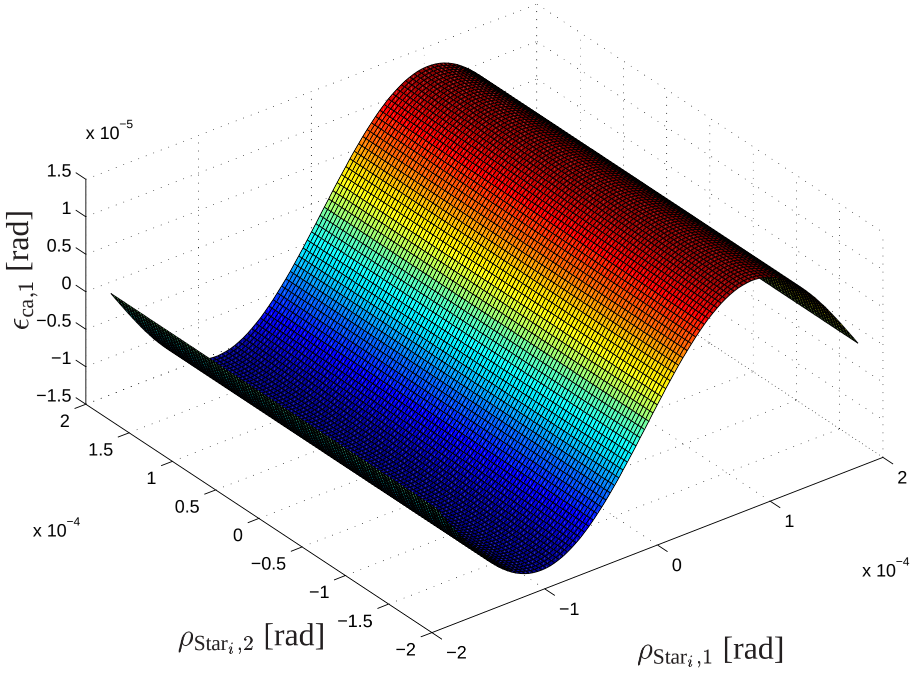


**Figure 5.6**: *Centroiding error of* $\rho_{Star_i,1}$.

The plot of the centroiding error in the vertical direction, $\rho_{\text{Star}_i,2}$ is the same as in Figure 5.6, except that the function is periodic with $\rho_{\text{Star}_i}2$ instead of $\rho_{\text{Star}_i,1}$. The plot is not shown as it is the plot in Figure 5.6 rotated $90°$.

## Residual Optical Error

The optical residual error is the error remaining in the optics after calibration, which is performed either on-ground or in-flight. The calibration error causes distortion in the image projected onto the CCD. The model of the error is based on a function from [Bayard, 1996], derived to simulate real-life experienced errors. The function describing the optical error $\boldsymbol{\epsilon}_{\text{opt}_i}$, in the angular direction of the $i$'th star, is given by

$$\boldsymbol{\epsilon}_{\text{opt}_i} = \frac{\boldsymbol{\rho}_{\text{Star}_i}}{\left|\boldsymbol{\rho}_{\text{Star}_i}\right|} p_{\text{opt}} \mathbf{sin}\left(\frac{2\pi\left|\boldsymbol{\rho}_{\text{Star}_i}\right|}{d_{\text{opt}}}\right) \tag{5.15}$$

where $p_{\text{opt}}$ is the worst case error (peak), and $d_{\text{opt}}$ is the period of the error. The values of the parameters should be set to

$$p_{\text{opt}} = \frac{1}{10} d_{\text{pixel}} \tag{5.16}$$

$$d_{\text{opt}} = \frac{1}{3} \alpha_{\text{FOV}} \tag{5.17}$$

where $\alpha_{\text{FOV}}$ is the angle of the square FOV, assuming that the horizontal and vertical distances are equal, and $d_{\text{pixel}}$ is the radial pixel spacing. These values are chosen in order to simulate the residual optical error experienced on the Cassini project..

The plot of the residual optical error in the horizontal direction $\rho_{\text{Star},1}$, as a function of $\rho_{\text{Star},1}$ and $\rho_{\text{Star},2}$ is given in Figure 5.7. The error is plotted over a FOV of $22^\circ$. The plot of the vertical error is left out, since it is Figure 5.7 rotated $90^\circ$.

## Noise Equivalent Angle

The noise equivalent angle is white noise added to the angular distances of the star direction. The noise originates from a number of optical and electrical inaccuracies. The NEA is given by

$$\boldsymbol{\epsilon}_{\text{nea}_i} = \begin{bmatrix} \epsilon_{\text{nea}_i,1} \\ \epsilon_{\text{nea}_i,1} \end{bmatrix} \sigma_{\text{st}} \tag{5.18}$$

where $\epsilon_{\text{nea}_i,1}$ and $\epsilon_{\text{nea}_i,2}$ are Gaussian white noise signals, with mean value of one, and $\sigma_{\text{st}}$ is the 1-sigma value of the star tracker's NEA. The 1-sigma value is increased as a function of the angular velocity of the satellite, following the principle described in star availability.

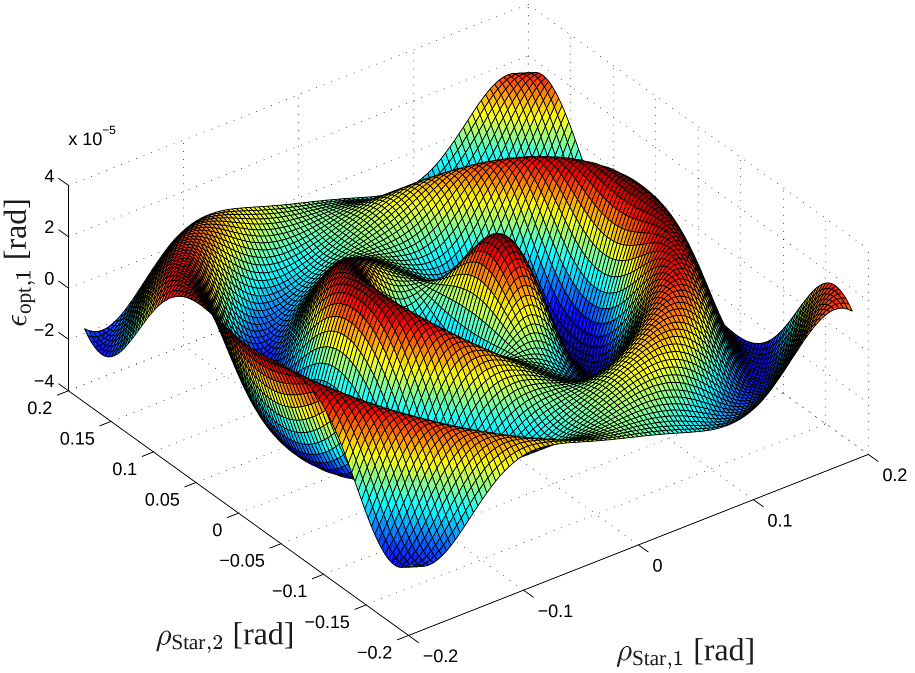


**Figure 5.7**: *Residual optical error of* $\rho_{Star,1}$.

## Attitude Measurement error

Given the angular measurement noise two vector sets can be found from the true star directions $\boldsymbol{\rho}_{\text{Star}_i}$ and the measured star directions $\boldsymbol{\rho}_{\text{Meas}_i}$, given by

$$\boldsymbol{\rho}_{\text{Meas}_i} = \boldsymbol{\rho}_{\text{Star}_i} + \boldsymbol{\epsilon}_{\text{ca}_i} + \boldsymbol{\epsilon}_{\text{opt}_i} + \boldsymbol{\epsilon}_{\text{nea}_i} \qquad i = 1, 2, ..., n_{\text{s}} \tag{5.19}$$

where $\boldsymbol{\rho}_{\text{Star}_i}$ is the true direction to the $i$'th star, $\boldsymbol{\epsilon}_{\text{ca}_i}$ is the error of the centroiding algorithm, $\boldsymbol{\epsilon}_{\text{opt}_i}$ is the residual optical error, and $\boldsymbol{\epsilon}_{\text{nea}_i}$ is the NEA.

An error quaternion $\boldsymbol{q}_{\epsilon_{\text{st}}}$, describing the rotation equivalent to the measurement error of the star tracker, is sought. Two vector sets are formed, containing the true and measured Cartesian coordinates in the CCD frame of the stars in the FOV. By observing Figure 5.2 it is seen that the Cartesian direction vectors are given by

$$\boldsymbol{s}_{\text{Star}_i} = \begin{bmatrix} -f\tan\left(\rho_{\text{Star}_i,1}\right) \\ -f\tan\left(\rho_{\text{Star}_i,2}\right) \\ f \end{bmatrix} \qquad i = 1, 2, ..., n_\text{s} \tag{5.20}$$

$$\boldsymbol{s}_{\text{Meas}_i} = \begin{bmatrix} -f\tan\left(\rho_{\text{Meas}_i,1}\right) \\ -f\tan\left(\rho_{\text{Meas}_i,2}\right) \\ f \end{bmatrix} \qquad i = 1, 2, ..., n_\text{s} \tag{5.21}$$

Normalizing the vectors yield

$$\hat{\boldsymbol{s}}_{\text{Star}_i} = -\begin{bmatrix} -\tan\left(\rho_{\text{Star}_i,1}\right) \\ \tan\left(\rho_{\text{Star}_i,2}\right) \\ 1 \end{bmatrix} \frac{1}{\sqrt{\tan\left(\rho_{\text{Star}_i,1}\right)^2 + \tan\left(\rho_{\text{Star}_i,2}\right)^2 + 1^2}} \qquad i = 1, 2, ..., n_\text{s} \tag{5.22}$$

$$\hat{\boldsymbol{s}}_{\text{Meas}_i} = \begin{bmatrix} \tan\left(\rho_{\text{Meas}_i,1}\right) \\ \tan\left(\rho_{\text{Meas}_i,2}\right) \\ 1 \end{bmatrix} \frac{1}{\sqrt{\tan\left(\rho_{\text{Meas}_i,1}\right)^2 + \tan\left(\rho_{\text{Meas}_i,2}\right)^2 + 1^2}} \qquad i = 1, 2, ..., n_\text{s} \tag{5.23}$$

From the two vector sets, Wahba's problem is described as a minimization of the performance index

$$L\left(\boldsymbol{A}\left(\boldsymbol{q}_{\epsilon_\text{st}}\right)\right) = \frac{1}{2}\sum_{j=1}^{n_\text{s}} w_i \left|\hat{\boldsymbol{s}}_{\text{Star}_i} - \boldsymbol{A}\left(\boldsymbol{q}_{\epsilon_\text{st}}\right)\hat{\boldsymbol{s}}_{\text{Meas}_i}\right|^2 \tag{5.24}$$

where $\boldsymbol{A}\left(\boldsymbol{q}_{\epsilon_\text{st}}\right)$ is the noise equivalent rotation, that fits the vectorized measurements to the data in the star catalogue. A number of algorithms exist to solve Wahba's problem of Equation 5.24. It is chosen to use the q-Method, which gives an analytical solution in the form of an attitude quaternion. The algorithm is described in Appendix B.

In order to obtain a minimum variance solution, the weight $w_i$ is calculated using

$$w_i^2 = \frac{1}{\sigma_\text{ca}^2 + \sigma_\text{st}^2} \tag{5.25}$$

where $\sigma_\text{ca}$ is the mean-square value of the centroiding error, calculated as

$$\sigma_\text{ca} = \frac{\sqrt{p_\text{ca}^2 + d_\text{pixel}^2}}{4} \tag{5.26}$$

Even though the centroiding error is not white, it is still used in the variance expression of Equation 5.25, since it is high frequency and independent of the NEA. Equation 5.26 arises from the general equation of the mean-square value of a sine function with amplitude of one.

From Equation 5.24 a noise equivalent quaternion $\boldsymbol{q}_{\epsilon_{\text{st}}}$ is acquired. Hence the measured attitude $\boldsymbol{q}_{\text{m}}$, given by the star tracker, is

$$\boldsymbol{q}_{\text{m}} = \boldsymbol{q}_{\epsilon_{\text{st}}} \boldsymbol{q}_{\text{st}} \tag{5.27}$$

where $\boldsymbol{q}_{\text{st}}$ is the true attitude of the star tracker. Note that quaternion multiplication is implied.

## Rate Dependent Performance

In order to simulate the decrease in performance of the star tracker, when the angular rate increases, two parameters of the model are calculated as functions of the angular rate: $m_{\text{cut}}$ and $\sigma_{\text{st}}$. The magnitude cut-off level $m_{\text{cut}}$ defines the number of stars to be selected from the star catalogue, based on magnitudes of the stars. The 1-sigma value of the NEA $\sigma_{\text{st}}$ defines the deviation of the white noise added by the star tracker.

In order to reduce model complexity, the rate is divided into three levels, low medium and high. Hence three values of $m_{\text{cut}}$ and $\sigma_{\text{st}}$ are calculated in the initialization of the model, and used dependent on the rate level.

The magnitude cut-off levels $m_{\text{cut,low}}$, $m_{\text{cut,mid}}$ and $m_{\text{cut,high}}$, are the levels of magnitude which are used to filter star catalogue in low, medium and high rate levels, respectively. The 1-sigma values of the NEA, used in the rate levels, are denoted $\sigma_{\text{st,low}}$, $\sigma_{\text{st,mid}}$ and $\sigma_{\text{st,high}}$.

The parameters $\omega_{\text{low,mid}}$ and $\omega_{\text{mid,high}}$ sets the values at which the angular rate of the star tracker changes level from low to medium and from medium to high, and vice versa. When the rate level changes, the parameters $\sigma_{\text{st}}$ and $m_{\text{cut}}$ are set to the appropriate values of the new level. The values of $m_{\text{cut}}$ and $\sigma_{\text{st}}$ are described as functions of the angular rate of the star tracker $\boldsymbol{\omega}_{\text{st}}$, by

$$m_{\text{cut}}\left(\boldsymbol{\omega}_{\text{st}}\right)=\begin{cases} m_{\text{cut,low}} & \text{if } \omega_{\text{max}}<\omega_{\text{low,mid}} \\ m_{\text{cut,mid}} & \text{if } \omega_{\text{low,mid}}\leq\omega_{\text{max}}<\omega_{\text{mid,high}} \\ m_{\text{cut,high}} & \text{if } \omega_{\text{mid,high}}\leq\omega_{\text{max}} \end{cases} \tag{5.28}$$

$$\sigma_{\text{st}}\left(\boldsymbol{\omega}_{\text{st}}\right)=\begin{cases} \sigma_{\text{st,low}} & \text{if } \omega_{\text{max}}<\omega_{\text{low,mid}} \\ \sigma_{\text{st,mid}} & \text{if } \omega_{\text{low,mid}}\leq\omega_{\text{max}}<\omega_{\text{mid,high}} \\ \sigma_{\text{st,high}} & \text{if } \omega_{\text{mid,high}}\leq\omega_{\text{max}} \end{cases} \tag{5.29}$$

where $\omega_{\text{max}}$ is the worst case angular rate of the stars on the CCD. The horizontal and vertical angular rate of the stars on the CCD, $\omega_{\rho,1}$ and $\omega_{\rho,2}$, can be expressed as functions of the angular rate of the star tracker $\boldsymbol{\omega}_{\text{st}}$, by

$$\omega_{\rho,1}=\left|\omega_{\text{st},2}\right|+\left|\omega_{\text{st},3}\right|\frac{\alpha_{\text{FOV}}}{\sqrt{2}} \tag{5.30}$$

$$\omega_{\rho,2}=\left|\omega_{\text{st},1}\right|+\left|\omega_{\text{st},3}\right|\frac{\alpha_{\text{FOV}}}{\sqrt{2}} \tag{5.31}$$

assuming that the roll rate $\omega_{\text{st},3}$ is acting on a star in the corner of the FOV, which is a worst case scenario. Based on the worst case angular rate of the stars on the CCD, the level parameter $\omega_{\text{max}}$ is set to the element of $\boldsymbol{\omega}_{\rho}$ with highest value. This can be expressed as

$$\omega_{\text{max}}=\begin{cases} \omega_{\rho,1} & \text{if } \omega_{\rho,1}\geq\omega_{\rho,2} \\ \omega_{\rho,2} & \text{if } \omega_{\rho,1}<\omega_{\rho,2} \end{cases} \tag{5.32}$$

Combining Equations 5.28, 5.29, 5.30, 5.31, and 5.32, yields the rate dependent parameters $m_{\text{cut}}$ and $\sigma_{\text{st}}$.

## Initial Parameters

The simulation model of the star tracker, makes use of several parameters, which must be given for the star tracker, or calculated based on the given data. The guide lines in [Bayard, 1996] are described first, which comply with experience from Cassini and SIRTF projects. Corrections have been made to these values, in order to reproduce data measured with the Terma star tracker. These changes are described in the next section.

The parameters which are needed are listed in Table 5.1.

The calculation of the parameters in Table 5.1 is based on a known set of parameters describing the characteristics of the star tracker. These fundamental parameters are

| Symbol | Description |
|---|---|
| $\alpha_{\text{FOV}}$ | Angular distance of the horizontal and vertical FOV. |
| $p_{\text{ca}}$ | Centroiding error peak. |
| $d_{\text{ca}}$ | Centroiding error period. |
| $p_{\text{opt}}$ | Optical distortion peak. |
| $d_{\text{opt}}$ | Optical distortion period. |
| $\sigma_{\text{st,low}}$ | 1-sigma value of the NEA at low rate. |
| $\sigma_{\text{st,mid}}$ | 1-sigma value of the NEA at medium rate. |
| $\sigma_{\text{st,high}}$ | 1-sigma value of the NEA at high rate. |
| $m_{\text{cut,low}}$ | Star magnitude filtering level at low rate. |
| $m_{\text{cut,mid}}$ | Star magnitude filtering level at medium rate. |
| $m_{\text{cut,high}}$ | Star magnitude filtering level at high rate. |
| $\omega_{\text{low,mid}}$ | Low/medium rate level. |
| $\omega_{\text{mid,high}}$ | Medium/high rate level. |

**Table 5.1**: *Initial parameters for the star tracker model.*

- The FOV of the camera, $\alpha_{\text{FOV}}$.
- Number of pixels per axis on the CCD, $n_{\text{pixels}}$.
- 1-sigma value of the NEA disregarding rate performance decrease, $\sigma_{\text{st,low}}$.
- Camera exposure time, $t_{\text{exp}}$.

The calculation of the parameters in Table 5.1 using the above four fundamental parameters is described in the following.

The centroiding error peak $p_{\text{ca}}$ is set to a fraction of a pixel. Experience from the SIRTF project suggests $1/40$ of a pixel, giving

$$p_{\text{ca}} = \frac{1}{40} \tag{5.33}$$

This value is chosen to achieve a 3-sigma RMS value of the centroiding error of $\epsilon_{\text{RMS}} = 0.05$. The 3-sigma RMS value of a sinusoidal error $p_{\text{ca}}\text{sin}\,(\omega x)$, is given by

$$\epsilon_{\text{RMS}} = \frac{3}{\sqrt{2}} p_{\text{ca}} \approx 2 p_{\text{ca}} \tag{5.34}$$

$$\Updownarrow$$

$$p_{\text{ca}} \approx \frac{1}{2}\epsilon_{\text{RMS}} = \frac{1}{40} \tag{5.35}$$

The period of the centroiding error $d_{\text{ca}}$ is set to the radial pixel spacing $d_{\text{pixel}}$, given by

$$d_{\text{ca}} = d_{\text{pixel}} = \frac{\alpha_{\text{FOV}}}{n_{\text{pixels}}} \tag{5.36}$$

The optical error is typically proportional with the size of the FOV. Based on experience of the Cassini project, $p_{\text{opt}}$ is given by

$$p_{\text{opt}} = \alpha_{\text{FOV}} \times 10^{-4} \tag{5.37}$$

In general, the optical error after a ground calibration is expected to be of low spatial frequency, hence $d_{\text{opt}}$ is chosen to be

$$d_{\text{opt}} = 0.6\frac{\alpha_{\text{FOV}}}{2} \tag{5.38}$$

The rate levels $\omega_{\text{low,mid}}$ and $\omega_{\text{mid,high}}$ are found from the amount of pixels the stars on the CCD will move in a single exposure. Change from low to medium level, is set when the image is smeared $2/3$ of a pixel. When $8.7$ pixels are smeared in a single exposure, the level changes from medium to high, i.e.

$$\omega_{\text{low,mid}} = \frac{2}{3}\frac{d_{\text{pixel}}}{t_{\text{exp}}} \tag{5.39}$$

$$\omega_{\text{mid,high}} = 8.7\frac{d_{\text{pixel}}}{t_{\text{exp}}} \tag{5.40}$$

The deviation of the NEA at low level rate must be specified for the star tracker. The values in medium and high rate levels are calculated from $\sigma_{\text{st,low}}$ by

$$\sigma_{\text{st,mid}} = 2\sigma_{\text{st,low}} \tag{5.41}$$

$$\sigma_{\text{st,high}} = 10\sigma_{\text{st,low}} \tag{5.42}$$

The magnitude cut-off value in low rate level is set to include all stars from the star catalogue. A value of ten is adequate. The values in medium and high rate level are calculated using a $-2.5\log\left(\omega_{\text{max}}/d_{\text{pixel}}\right)$ rate dependency [Bayard, 1996], where $\omega_{\text{max}}/d_{\text{pixel}}$ is the angular rate of the star tracker in pixels per second. The rate $\omega_{\text{max}}$ is calculated using Equations 5.30, 5.31, and 5.32.

## 5.3 Modeling the Terma NEMO Star Tracker

The star tracker model described in Section 5.2, is a general model. The calculation of the initial parameters, adapts the model to replicate experience from star trackers used in the Cassini and SIRTF projects. It is not expected that the guidelines for calculating the initial parameters comply with the Terma NEMO star tracker. Hence the initial parameters are corrected in order to comply with measurements from the star tracker conducted by Terma .

The parameters supplied in [Paulsen and Maresi, 2000], are RMS errors from the centroiding error, residual optical error and noise equivalent angle, observed for a single star. The maximum occurring chromatic aberration error is also stated. The parameters are listed in Table 5.2.

| **Source** | **Value** |
|---|---|
| NEA | $1.9$as (RMS) |
| Chromatic Aberration | $1.6$as (Max) |
| Centroiding Error | $2.0$as (RMS) |
| Residual Distortion | $4.0$as (RMS) |
| Total | $5.1$as (RSS) |

**Table 5.2**: *Single star performance observed on the second flight model of the NEMO star tracker [Paulsen and Maresi, 2000].*

The chromatic aberration is not modeled, hence it is chosen to increase the RMS error of the NEA to the RSS value of the NEA and chromatic aberration

$$\text{RSS}\left(\epsilon_{\text{nea}}, \epsilon_{\text{chrom}}\right) = \sqrt{1.9^2 + 1.6^2}\text{as} = 2.5\text{as} \tag{5.43}$$

where $\epsilon_{\text{chrom}}$ is the error of the chromatic aberration.

The parameters of the model, subject to a correction, are $p_{\text{ca}}$ and $p_{\text{opt}}$, which are the amplitudes of the periodic centroiding and optical errors, respectively. The parameters are found by multiplying the recommended values by a correction factor. This correction factor is found by adding only the noise source at hand, then adjusting the amplitude, such that the resulting angular error is consistent with the values of Table 5.2.

In order to compare the values from Table 5.2 with model performances, the vectorized pointing error must be transformed to a single parameter error, which is the angular separation between the true star direction and the measured star direction. Since the elements of a star direction vector $\boldsymbol{\rho}_{\text{Star}_i}$ are given in radians, the angular separation between the true and noise inflicted star directions, is calculated using rules of cosine for spherical triangles (see Figure 5.8)

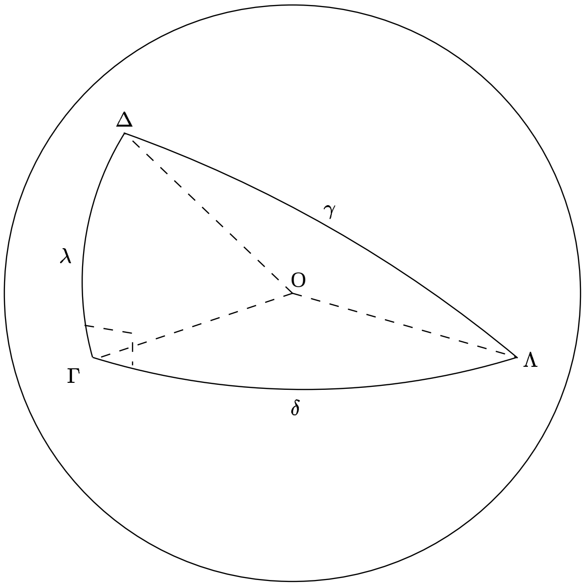


**Figure 5.8**: *Illustration of spherical triangle.*

$$\begin{aligned}\cos(\gamma) &= \cos(\delta)\cos(\lambda) + \sin(\delta)\sin(\lambda)\cos(\Gamma)\\ &= \cos(\delta)\cos(\lambda)\end{aligned} \tag{5.44}$$

which holds when $\Gamma$ is right angled. Realizing that $\delta = \rho_{\text{Star}_i,1}$ and $\lambda = \rho_{\text{Star}_i,2}$, and denoting the sought single angular direction of the $i$'th star $\xi_{\text{Star}_i}$, yields

$$\xi_{\text{Star}_i} = \cos^{-1}\left(\cos\left(\rho_{\text{Star}_i,1}\right)\cos\left(\rho_{\text{Star}_i,2}\right)\right) \tag{5.45}$$

which holds since the angle between the horizontal and vertical directions is right.

The associated noise inflicted direction is given by

$$\xi_{\text{Meas}_i} = \cos^{-1}\left(\cos\left(\rho_{\text{Meas}_i,1}\right)\cos\left(\rho_{\text{Meas}_i,2}\right)\right) \tag{5.46}$$

where $\boldsymbol{\rho}_{\text{Meas}_i}$ is the star direction with the noise source, which is sought fitted to the values of Table 5.2, is added. The single angular error in the star direction $\xi_{\text{err}_i}$, is the difference between the directions of Equations 5.45 and 5.46

$$\xi_{\text{err}_i} = \xi_{\text{Star}_i} - \xi_{\text{Meas}_i} \tag{5.47}$$

Calculating the single angle error in the star directions for all possible star directions in the FOV, yields RMS values of the single star performance which can be compared with the values of Table 5.2. Table 5.3 shows the correction factors, giving the sought single star performance.

| Parameter | Correction Factor |
|---|---:|
| $p_{\text{ca}}$ | 0.03697 |
| $p_{\text{opt}}$ | 0.7204 |
| $\sigma_{\text{st,low}}$ | 1.3073 |

**Table 5.3**: *Correction factors needed to reflect the Terma NEMO star tracker.*

The specified RMS value of the NEA, specified for the Terma star tracker, should be multiplied by $1.3073$, in order to include the chromatic error. The peak of the error functions of the centroiding error and the optical error, given in Equations 5.33 and 5.38, are multiplied by $0.03697$ and $0.7204$, respectively.

## Simulation Results

The model of the NEMO star tracker has been implemented as MATLAB functions, and a SIMULINK interface is supplied. The night sky test from [Paulsen and Maresi, 2000] is simulated, and the results are compared to those of Terma.

The data from Terma originate from a night sky test, where the star tracker is pointed to the sky, and rotated by the rotation of the Earth. The pointing performance of the star tracker is shown in Figure 5.9.

The roll performance during the same test is shown in Figure 5.10. The performance of the roll is significantly poorer than the pointing performance. The RMS errors of the pointing and roll performance are $1.2$as and $6.2$as, respectively.

The SIMULINK model parameters are initialized from the fundamental parameters of the NEMO star tracker. The parameters are listed in Table 5.4. Note that the magnitude thresholds are all set to ten, since the rate dependent star selection has not been implemented at this point. This is discussed in the conclusion of the section.

The simulated performance of the star tracker model is shown in Figure 5.11. The RMS errors of the pointing and roll performance are $0.99$as and $6.19$as, respectively.

In this simulation, twelve stars were used from the star catalogue. The number of stars used by the star tracker in the Terma night sky test is not stated in the reference. Sources at Terma state, that the typical number of stars used by the star tracker is around 50. With this number of stars, the simulation model performance is better than in the test. In order to examine the performance of the star tracker wrt. the number of stars used, the

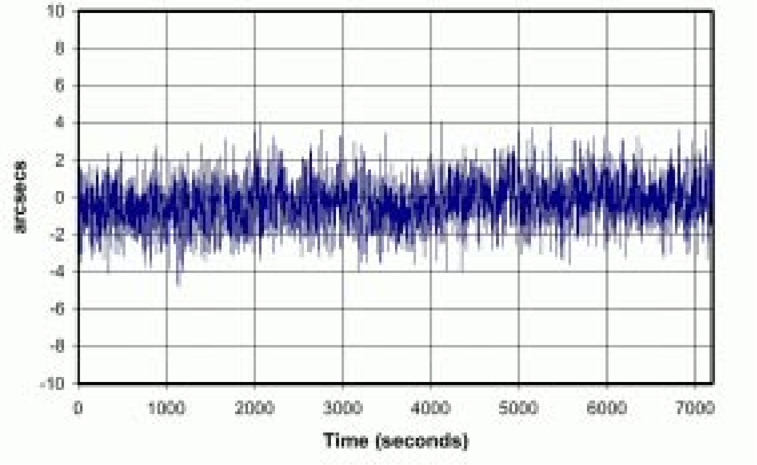


**Figure 5.9**: *NEMO star tracker pointing performance in night sky test [Paulsen and Maresi, 2000].*

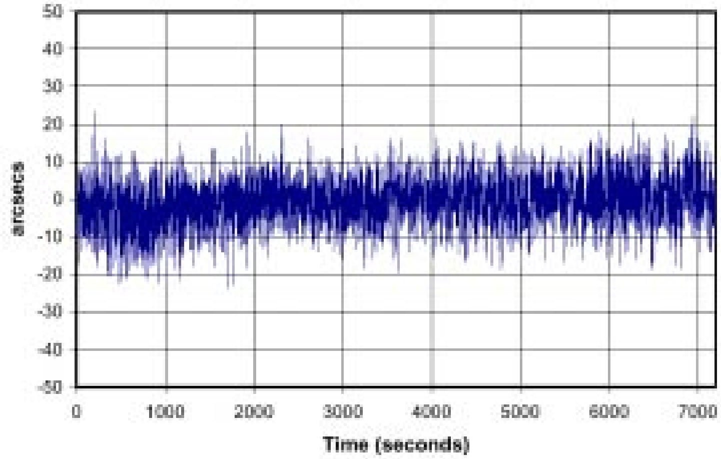


**Figure 5.10**: *NEMO star tracker roll performance in night sky test [Paulsen and Maresi, 2000].*

| **Fundamental Parameters** | |
|---|---:|
| FOV | $22^\circ$ |
| Pixels | $1024$ |
| NEA | $1.9$as |
| Camera Exposure Time | $150$ms |
| **Model Parameters** | |
| $\alpha_{\text{FOV}}$ | $0.3840$rad |
| $p_{\text{ca}}$ | $0.01386 \times 10^{-3}$rad |
| $d_{\text{ca}}$ | $0.3750 \times 10^{-3}$rad |
| $p_{\text{opt}}$ | $0.02766 \times 10^{-3}$rad |
| $d_{\text{opt}}$ | $0.1152$rad |
| $\sigma_{\text{st,low}}$ | $0.01211 \times 10^{-3}$rad |
| $\sigma_{\text{st,mid}}$ | $0.02422 \times 10^{-3}$rad |
| $\sigma_{\text{st,high}}$ | $0.1211 \times 10^{-3}$rad |
| $m_{\text{cut,low}}$ | 10 |
| $m_{\text{cut,mid}}$ | 10 |
| $m_{\text{cut,high}}$ | 10 |
| $\omega_{\text{low,mid}}$ | $1.667 \times 10^{-3}$rad/s |
| $\omega_{\text{mid,high}}$ | $21.75 \times 10^{-3}$rad/s |

**Table 5.4**: *The fundamental parameters of the Terma star tracker and the model parameters.*

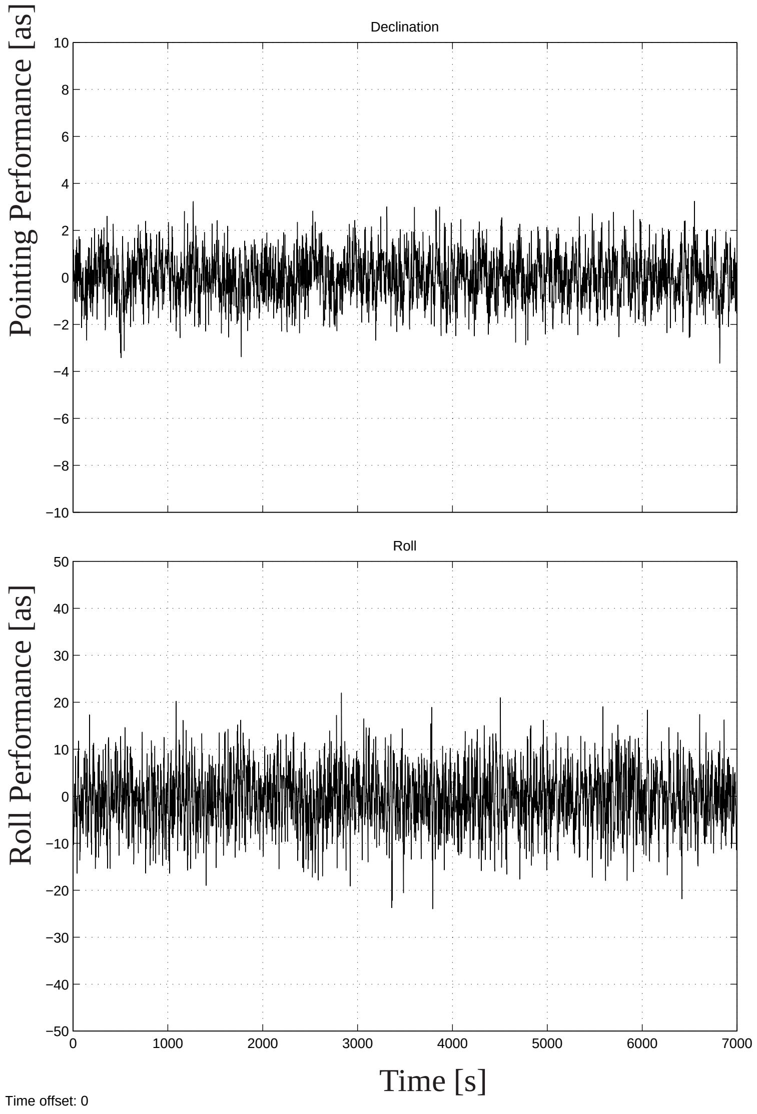


**Figure 5.11**: *Star tracker simulation of night sky test using 12 stars.*

same simulation has been run repeatedly with different number of stars in the catalogue. The result is plotted in Figure 5.12.

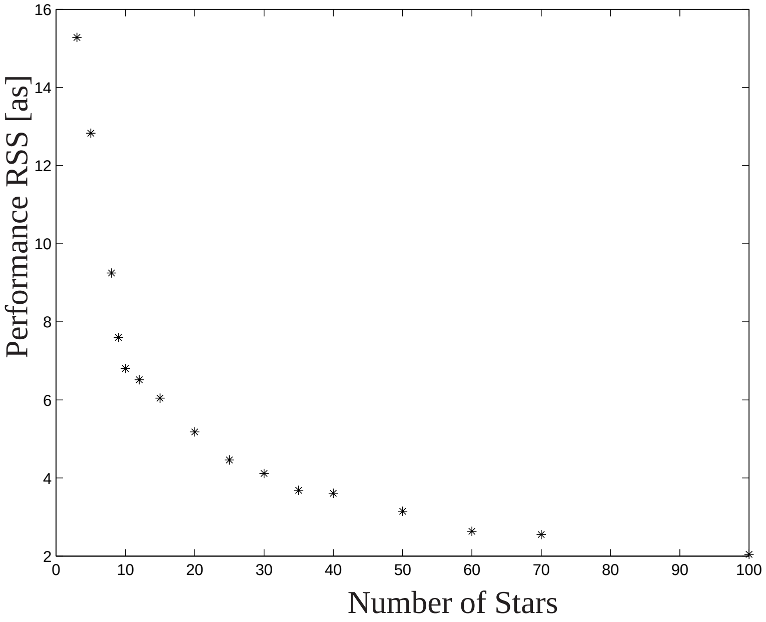


**Figure 5.12**: *Star tracker performance in night sky simulation as a function of number of stars used from the catalogue.*

It is seen from the data plotted in Figure 5.12 that when using 50 stars, the RSS performance of the star tracker is $3.15$as. From the simulation result, the pointing performance is found to be $0.48$as and $3.11$as for the roll. This is significantly better than the test results. The model should be verified with more tests, and the tests should show the number of stars used by the star tracker.

The angular rate of the night sky test, is the angular rate of the Earth's rotation

$$\begin{aligned}\omega_{\text{Earth}} &= \frac{2\pi}{24 \cdot 3600}\text{rad/s} \\ &= 0.7272 \times 10^{-6}\text{rad/s}\end{aligned} \tag{5.48}$$

hence the model operates in the low rate level. From Table 5.4 it is seen that when the rate of the star tracker increases to $1.667 \times 10^{-3}$rad the model is in medium level, and in high level when the rate increases to $21.75 \times 10^{-3}$rad/s. No test results have been acquired in order to verify the credibility of these values.

The night sky test simulation has been repeated with higher angular rate, in order to compare the results of the performance, when increasing the angular rate of the star tracker.

The results of the simulation in low, medium and high level is shown in Table 5.5. In medium level, the angular rate of the star tracker was set to $0.01$rad. This value is between $\sigma_{\text{low,mid}}$ and $\sigma_{\text{mid,high}}$ from Table 5.4, hence the model will operate in medium level. The pointing performance has an RMS value of $1.76$as and roll performance of $10.6$as RMS.

| Rate Level | Pointing (RMS) | Roll (RMS) | Total (RSS) |
|---|---|---|---|
| Low | $0.99$as | $6.20$as | $6.28$as |
| Medium | $1.76$as | $10.6$as | $10.7$as |
| High | $7.80$as | $50.3$as | $50.9$as |

**Table 5.5**: *Results of rate dependent simulation.*

The simulation was repeated with an angular rate of $30 \times 10^{-3}$rad$/$s, which is higher than $\sigma_{\text{mid,high}}$. Hence the model will operate in high level. The pointing performance has an RMS value of $7.80$as and and roll performance RMS value of $50.3$as.

## Discussion of Results

The model correctly decreases the performance of the star tracker, when the angular rate increases, as seen in Table 5.5. This is done by increasing the NEA when the angular rate increases. The star catalogue selection is not used in the simulation, hence all stars are used regardless of angular rate. The magnitude cut-off has been set to ten at all rate levels. The suggested $-2.5\log\left(\omega_{\text{max}}/d_{\text{pixel}}\right)$ dependency, where $\omega_{\text{max}}$ is the angular rate of the star tracker in pixel per seconds, yields values that exclude all the stars from the catalogue in both medium and high rate level.

In order to investigate the magnitude thresholds further, the calculations are shown. From Table 5.4 it is seen that the when the rate level exceeds $1.667 \times 10^{-3}$as, the model will operate in medium rate level. The magnitude cut-off level is sought for this level. The pixel spacing in radians is

$$d_{\text{pixel}} = \frac{\alpha_{\text{FOV}}}{n_{\text{pixels}}} = \frac{0.3840}{1024} = 0.3750 \times 10^{-3}\text{rad} \tag{5.49}$$

using the values of Table 5.4. The angular rate level $\omega_{\text{low,mid}}$ is set to the rate where $2/3$ of a pixel is smeared in a single exposure

$$\begin{aligned}\omega_{\text{low,mid}} &= \frac{2}{3}\frac{d_{\text{pixel}}}{t_{\text{exp}}} \\ &= \frac{2}{3}\frac{0.3750 \times 10^{-3}}{0.15}\text{rad} \\ &= 1.667 \times 10^{-3} \end{aligned} \tag{5.50}$$

as recommended in [Bayard, 1996]. Calculating the magnitude threshold as suggested in [Bayard, 1996], yields

$$m_{\text{cut}} = -2.5\log\left(\frac{1.667 \times 10^{-3}}{0.3750 \times 10^{-3}}\right) = -3.73 \tag{5.51}$$

This value will exclude all stars from the catalogue, since the brightest star is Sirius with a magnitude of $-1.46$. This implies that the suggested magnitude threshold is inapplicable for the NEMO star tracker. Either the smearing factor of $2/3$ in Equation 5.50 is too large, or the rate dependency for the NEMO star tracker should be described by a different function, than in Equation 5.51.

If the number of photons that must reach a single pixel, in order for a light source to be detectable, is known, the required magnitude of a star, given the angular rate, can be found. This is the function that describes the threshold of the star magnitudes. This has not been analyzed in this project.

It should be noted that there is no verification of the magnitude cut-off levels in [Bayard, 1996], instead all the cut-off levels have been set to ten.

As mentioned in the above, no test results are available to verify the results of the rate dependent performance. Tests should be conducted, which can validate or invalidate the model results.

When conducting the simulations, it has been noticed that biases can occur in the performances of the star tracker. This is due to low spatial frequency of the optical distortion. Since the star catalogue has been randomly generated, the bias depends on the star locations on the error function of the optical error.

The low spatial frequency is suggested in [Bayard, 1996], based on the expected error after ground calibration of the SIRTF star tracker, which has a FOV of $3^\circ$. This is significantly smaller than the $22^\circ$ FOV of the NEMO star tracker. If the frequency of the optical is increased, it will act more like a white noise, and it is expected that the RMS performance of the star tracker will decrease. Since the performance is better in the simulation than in the test, when using 50 stars as stated by Terma, this effect is desired. This also implies that the reason that the performance of the model is better than in the test, is the low spatial frequency of the optical error.

## 5.4 Conclusion

In this chapter, a model of the NEMO star tracker has been derived. The model has been implemented in SIMULINK and the model parameters are tuned to fit the NEMO star tracker. The simulation results are compared with a night sky test conducted by Terma .

One major property of a star tracker is the decreased accuracy of rotation about the boresight axis. The model correctly simulates this characteristic. When using twelve stars in a random generated star catalogue, the result of the model is close to the measurements of the night sky test. The number of stars used in the test has not been observed, but staff at Terma states that the quantity is around 50. When using 50 stars in the simulation the results are significantly better than the in test. This is also the experience of Terma simulations.

The low spatial frequency of the optical error, could be the cause of this. The frequency of the optical error is based on the size of the FOV. The FOV of the NEMO star tracker is seven times larger than the SIRTF star tracker, hence this indeed is the parameter where the two trackers differ most. Increased frequency of the optical error increases the RMS error of the performance, and reduces the probability of bias, since the function of the error at high frequency looks more like a random function.

The angular rate degradation is done by increasing the NEA and decreasing the number of stars used from the star tracker. No test have been conducted to verify the rate performance degradation, instead the performance is presented in the three rate levels.

The magnitude filtering of the star catalogue has not been successfully implemented since the recommended magnitude thresholds, yields values that exclude all stars from the catalogue. Instead all stars are used, disregarding the rate of the star tracker.

Chapter 6

# Gyro Model

This section describes the model used for the rate gyros mounted on the satellite. Several error sources inflict the measurement of a rate gyro. Based on a trade of between model complexity and accuracy, the modeled error sources are stated in Table 6.1.

| | |
|---|---|
| **Scale Factor Error** | The scale factor describes the relationship between the measured output voltage and the angular rate of the input axis. This scale factor is not known exact. |
| **Alignment Error** | The mounting of the gyro is inflicted with an error, resulting in an input axis that is different from the intended. |
| **Drift** | The measurement from the gyro has a drift, which is modeled as a random walk. |
| **Bias** | The bias is the expected gyro output when the input rate is zero. |
| **White Noise** | White noise is added to the measurement due to electrical perturbations. |

**Table 6.1**: *Modeled rate gyro error sources.*

The direction of the angular rate measured by the gyro, is defined by the orientation of the gyro. The vector around which angular rate is measured, is the input axis of the gyro, denoted $\hat{\boldsymbol{r}}_{\mathrm{g}}$.

The gyro cannot measure the direction of the angular rate. The direction is given by the orientation of the gyro. The magnitude of the angular rate is measured by the gyro, and the direction is the direction of the gyro input axis.

Given an arbitrary angular rate of the satellite $\boldsymbol{\omega}$, on which the gyro is mounted, the measured angular rate is the projection of $\boldsymbol{\omega}$ onto the input axis. The projection is shown in Figure 6.1, denoted $\boldsymbol{\omega}_\mathrm{g}$. Hence the angular rate magnitude measured by the gyro $\omega_\mathrm{g}$, is given by

$$\omega_\mathrm{g} = \boldsymbol{\omega}^\mathrm{T}\hat{\boldsymbol{r}}_\mathrm{g} \tag{6.1}$$

assuming zero noise. Given the magnitude measurement from the gyro and the input axis $\hat{\boldsymbol{r}}_\mathrm{g}$, the angular rate is

$$\boldsymbol{\omega}_\mathrm{g} = \left[\boldsymbol{\omega}^\mathrm{T}\hat{\boldsymbol{r}}_\mathrm{g}\right]\hat{\boldsymbol{r}}_\mathrm{g} \tag{6.2}$$

Equation 6.1 gives the true angular rate of the input axis. In the following sections, the equation is expanded to include the errors stated in Table 6.1.

## 6.1 Scale Factor Error

In order to include the scale factor error, the projection factor is written as

$$\boldsymbol{\omega}^\mathrm{T}\hat{\boldsymbol{r}}_\mathrm{g} = k_\mathrm{sf}u_\mathrm{g} \tag{6.3}$$

where $k_\mathrm{sf}$ is the true scale factor and $u_\mathrm{g}$ is the voltage output from the gyro. Including the scale factor error, the perturbed projection factor $p_\epsilon$ is

$$p_\epsilon = \left[k_\mathrm{sf} + \epsilon'_\mathrm{sf}\right]u_\mathrm{g} \tag{6.4}$$

which is written in terms of the true projection factor as

$$\begin{aligned} p_\epsilon &= \boldsymbol{\omega}^\mathrm{T}\hat{\boldsymbol{r}}_\mathrm{g} + \epsilon'_\mathrm{sf}u_\mathrm{g} \\ &= \boldsymbol{\omega}^\mathrm{T}\hat{\boldsymbol{r}}_\mathrm{g}\left[1 + \epsilon_\mathrm{sf}\right] \end{aligned} \tag{6.5}$$

Inserting the perturbed projection factor into Equation 6.1, yields

$$\omega_\mathrm{g} = \boldsymbol{\omega}^\mathrm{T}\hat{\boldsymbol{r}}_\mathrm{g}\left[1 + \epsilon_\mathrm{sf}\right] \tag{6.6}$$

The scale factor error $\epsilon_\mathrm{sf}$ is a constant, in order to minimize model complexity. In reality, the scale factor error changes with the temperature of the gyro.

## 6.2 Alignment Error

When mounting the gyro, the input axis of the gyro is sought aligned with some vector on the satellite. In Figure 6.1 a gyro frame has been placed with the $\boldsymbol{z}_\mathrm{g}$ axis pointing in the direction of the intended input axis $\hat{\boldsymbol{r}}_\mathrm{m}$, and centered in the mounting point of the gyro. The angle $\epsilon_\mathrm{align}$ is the angular separation between the sought input axis $\hat{\boldsymbol{r}}_\mathrm{m}$ and the true input axis $\hat{\boldsymbol{r}}_\mathrm{g}$.

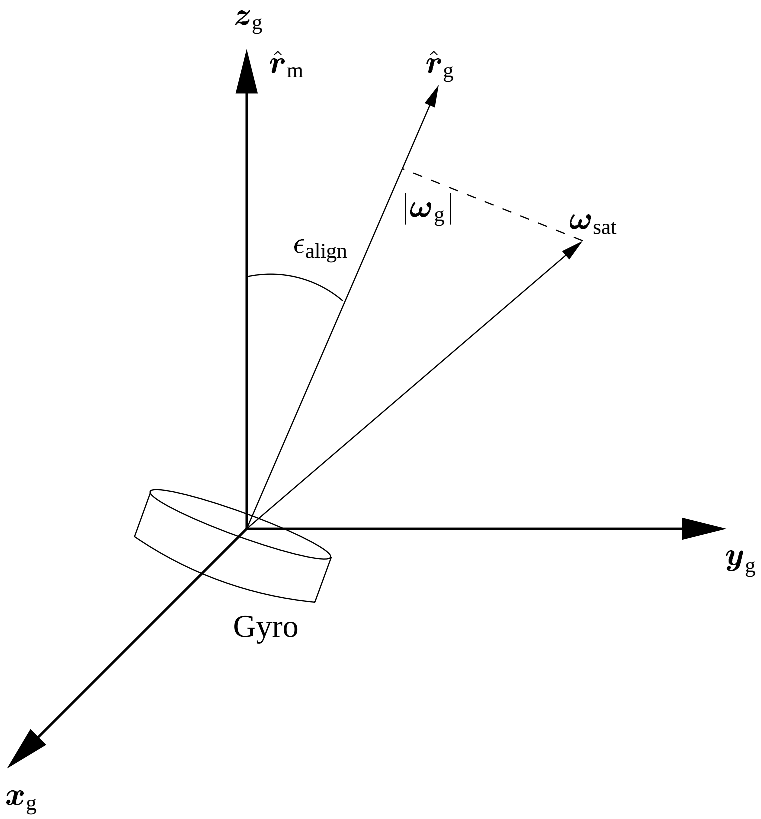

$\boldsymbol{z}_\mathrm{g}$
$\hat{\boldsymbol{r}}_\mathrm{m}$
$\hat{\boldsymbol{r}}_\mathrm{g}$
$|\boldsymbol{\omega}_\mathrm{g}|$
$\boldsymbol{\omega}_\mathrm{sat}$
$\epsilon_\mathrm{align}$
$\boldsymbol{y}_\mathrm{g}$
Gyro
$\boldsymbol{x}_\mathrm{g}$

**Figure 6.1**: *Misalignment of the gyro wrt. the gyro frame.*

The true input axis is written in terms of the sought input axis, as

$$\hat{\boldsymbol{r}}_\mathrm{g} = \boldsymbol{A}\left(\epsilon_\mathrm{align}\right)\hat{\boldsymbol{r}}_\mathrm{m} \tag{6.7}$$

Inserting into equation 6.6, yields

$$\omega_\mathrm{g} = \boldsymbol{\omega}^\mathrm{T}\boldsymbol{A}\left(\epsilon_\mathrm{align}\right)\hat{\boldsymbol{r}}_\mathrm{m}\left[1 + \epsilon_\mathrm{sf}\right] \tag{6.8}$$

As with the scale factor error the alignment error is modeled as a constant noise source, even though thermal fluctuations perturb the alignment of the gyro. This is again a trade off between model complexity and model accuracy.

## 6.3 Drift

The drift $\epsilon_{\text{drift}}$ is an error which is time correlated. The error is added to the gyro measurement of Equation 6.8, which becomes

$$\omega_{\text{g}} = \boldsymbol{\omega}^{\text{T}} \boldsymbol{A}\left(\epsilon_{\text{align}}\right) \hat{\boldsymbol{r}}_{\text{m}} \left[1 + \epsilon_{\text{sf}}\right] + \epsilon_{\text{drift}} \tag{6.9}$$

Note that the drift error is added in the direction of the input axis, which is expressed in terms of the alignment error. The drift error is modeled as a random walk, by the differential equation

$$\dot{\epsilon}_{\text{drift}} = w_{\text{drift}} \tag{6.10}$$

where $w_{\text{drift}}$ is zero mean uncorrelated noise, with deviation $\sigma_{\text{drift}}$.

## 6.4 Bias

The bias is the expected zero input value of the angular rate of the input axis. Hence it is a constant added to the gyro measurement. Equation 6.9 is expanded to

$$\omega_{\text{g}} = \boldsymbol{\omega}^{\text{T}} \boldsymbol{A}\left(\epsilon_{\text{align}}\right) \hat{\boldsymbol{r}}_{\text{m}} \left[1 + \epsilon_{\text{sf}}\right] + \epsilon_{\text{drift}} + \epsilon_{\text{bias}} \tag{6.11}$$

## 6.5 White Noise

White noise is added in the direction of the input axis in order to simulate electrical perturbations, and other unmodeled disturbances. The final equation of the gyro measurement is

$$\omega_{\text{g}} = \boldsymbol{\omega}^{\text{T}} \boldsymbol{A}\left(\epsilon_{\text{align}}\right) \hat{\boldsymbol{r}}_{\text{m}} \left[1 + \epsilon_{\text{sf}}\right] + \epsilon_{\text{drift}} + \epsilon_{\text{bias}} + w_{\text{g}} \tag{6.12}$$

where $w_{\text{g}}$ is zero mean Gaussian white noise with a deviation of $\sigma_{\text{g}}$.

## 6.6 Modeling the QRS11Pro Rate Gyro

From Equation 6.12 the model of the measurement is given. It is chosen to include the direction of the angular rate in the model. Recalling Equation 6.2, the vector of the angular rate is given by

$$\boldsymbol{\omega}_{\mathrm{g}} = \left[\boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{A}\left(\epsilon_{\mathrm{align}}\right) \hat{\boldsymbol{r}}_{\mathrm{m}} \left[1 + \epsilon_{\mathrm{sf}}\right] + \epsilon_{\mathrm{drift}} + \epsilon_{\mathrm{bias}} + w_{\mathrm{g}}\right] \hat{\boldsymbol{r}}_{\mathrm{m}} \tag{6.13}$$

where $\boldsymbol{A}\left(\epsilon_{\mathrm{align}}\right) \hat{\boldsymbol{r}}_{\mathrm{m}}$ is the true input axis in terms of the sought input axis, $\epsilon_{\mathrm{sf}}$ is the scale factor error, $\epsilon_{\mathrm{drift}}$ is the drift of the measurement, $\epsilon_{\mathrm{bias}}$ is the zero input bias, and $w_{\mathrm{g}}$ is white noise.

Note that the direction of the angular rate is the input axis $\hat{\boldsymbol{r}}_{\mathrm{m}}$, and not the true input axis $\hat{\boldsymbol{r}}_{\mathrm{g}}$, since the true input axis is unknown. The best estimate of the angular rate, assuming no knowledge of the alignment error, is the estimated rate magnitude in the direction of the sought input axis.

From the specifications of the gyro, the parameters of Equation 6.13 are found. Table 6.2 shows the parameters used for the simulation model. The misalignment angle is an expected value.

| **Parameter** | **Symbol** | **Value** |
|---|---|---|
| Scale Factor Error | $\epsilon_{\mathrm{sf}}$ | $1\%$ |
| Misalignment Angle | $\epsilon_{\mathrm{align}}$ | $0.5^\circ$ |
| Rate Drift Deviation | $\sigma_{\mathrm{drift}}$ | $0.0002^\circ/\mathrm{s}^2$ |
| Bias | $\epsilon_{\mathrm{bias}}$ | $0.5^\circ/\mathrm{s}$ |
| White Noise Deviation | $\sigma_{\mathrm{g}}$ | $\sqrt{2} \times 0.01^\circ/\mathrm{s}$ |

**Table 6.2**: *Model parameters used in the gyro simulation.*

The white noise deviation is found from the white noise specification of $0.01^\circ/\mathrm{s}/\sqrt{\mathrm{Hz}}$. The low pass 2Hz filter in the external electronics filters the white noise with a frequency above 2Hz, regardless of the sampling frequency.

### Simulation Results

The gyro model has been implemented in SIMULINK , and the result is shown in Figure 6.2.

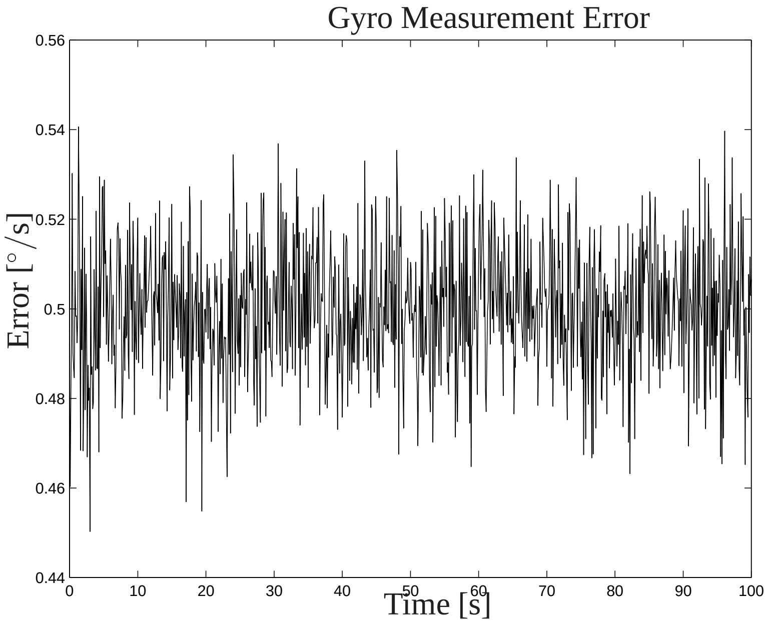


**Figure 6.2**: *Angular rate measurement noise calculated by the gyro noise model.*

## Discussion of Results

The result of the simulation gives the expected result. No data from real life tests with the QRS1Pro gyro have been available, hence the model lacks verification.

The verification of the model is simply done, by comparing the output noise with the specification of the gyro. Comparing the parameters of Table 6.2 with the Figure 6.2 confirms the correct implementation of the model. However the specifications should be modified to reflect the true behavior of the gyro, which only can be done through a real life test. This is not done in this project, instead the parameters of Table 6.2 are used.

Chapter 7

# Modeling Satellite Attitude

This chapter derives the equations used for modeling the kinematics and dynamics of satellite rotations, and has previously been published in [Bhanderi, 2001].

The direct cosine matrices contain nine parameters with three degrees of freedom. Due to this redundancy, numerous ways of representing the satellite attitude with a minimum set of parameters have been developed. Euler angles describe the rotation around the principal axes and use therefore only three parameters. However some singularities arise for some rotations, which is why Euler angles are commonly used when the attitude of the object involved, is known to be within a certain margin [Wertz, 1978].

Quaternions use four parameters with a single constraint, to represent attitude, and are subject to no singularities. This is useful when considering that the attitude of a satellite is usually unknown after the release from the launcher. For this reason quaternions are commonly used in space applications and also for this project. Appendix A on page 133 gives a brief description of the quaternions and their algebra.

The modeling of a satellite's rotation is divided into the kinematic equation and the dynamic equation. The kinematic equation describes the change in the attitude parameters of the satellite, regardless of the forces acting on it. The dynamic equation describes the time dependent parameters as functions of outer forces.

## 7.1 Kinematic Equation

Let the attitude of a satellite at time $t$ and $t + \Delta t$ be denoted $\boldsymbol{q}\,(t)$ and $\boldsymbol{q}\,(t + \Delta t)$. If the rotation of the satellite in the time period $\Delta t$ is denoted $\boldsymbol{q}\,(\Delta t)$, the propagation of the attitude from $t$ to $t + \Delta t$ can be written

$$\boldsymbol{q}\left(t+\Delta t\right)=\boldsymbol{q}\left(t\right)\boldsymbol{q}\left(\Delta t\right) \tag{7.1}$$

Writing $\boldsymbol{q}\left(\Delta t\right)$ in terms of rotation angle $\Delta\phi$ around the vector $\boldsymbol{u}$ in time $\Delta t$, yields

$$\boldsymbol{q}\left(\Delta t\right)=\begin{bmatrix} u_1\sin\left(\frac{\Delta\phi}{2}\right)\\ u_2\sin\left(\frac{\Delta\phi}{2}\right)\\ u_3\sin\left(\frac{\Delta\phi}{2}\right)\\ \cos\left(\frac{\Delta\phi}{2}\right)\end{bmatrix} \tag{7.2}$$

Assuming that $\boldsymbol{u}$ and $\Delta\phi$ are constant over the time $\Delta t$, and using the definition of the quaternion product, Equation 7.1 is written

$$\boldsymbol{q}\left(t+\Delta t\right)=\left[\cos\left(\frac{\Delta\phi}{2}\right)\mathbf{1}_{4\times 4}+\sin\left(\frac{\Delta\phi}{2}\right)\begin{bmatrix} 0 & u_3 & -u_2 & u_1\\ -u_3 & 0 & u_1 & u_2\\ u_2 & -u_1 & 0 & u_3\\ -u_1 & -u_2 & -u_3 & 0\end{bmatrix}\right]\boldsymbol{q}\left(t\right) \tag{7.3}$$

where $\mathbf{1}_{4\times 4}$ is the 4 by 4 identity matrix. For infinite small time steps, $\Delta\phi$ can be approximated by

$$\Delta\phi=\left|\boldsymbol{\omega}\right|\Delta t \tag{7.4}$$

where $\boldsymbol{\omega}$ is the instantaneous angular velocity of the satellite. Using small angle approximations of sine and cosine, Equation 7.3 can be expressed as

$$\boldsymbol{q}\left(t+\Delta t\right)=\left[\mathbf{1}_{4\times 4}+\frac{\Delta t}{2}\boldsymbol{\Omega}\right]\boldsymbol{q}\left(t\right) \tag{7.5}$$

where

$$\boldsymbol{\Omega}=\left|\boldsymbol{\omega}\right|\begin{bmatrix} 0 & u_3 & -u_2 & u_1\\ -u_3 & 0 & u_1 & u_2\\ u_2 & -u_1 & 0 & u_3\\ -u_1 & -u_2 & -u_3 & 0\end{bmatrix} \tag{7.6}$$

Realizing that $\boldsymbol{u} = \hat{\boldsymbol{\omega}}$, Equation 7.6 can be written

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{7.7}$$

The differential equation of $\boldsymbol{q}\,(t)$ is defined as

$$\dot{\boldsymbol{q}}\,(t) = \lim_{\Delta t \to 0} \frac{\boldsymbol{q}\,(t + \Delta t) - \boldsymbol{q}\,(t)}{\Delta t} \tag{7.8}$$

Inserting Equation 7.5 yields the sought kinematic differential equation

$$\dot{\boldsymbol{q}}\,(t) = \frac{1}{2}\boldsymbol{\Omega}\boldsymbol{q}\,(t) \tag{7.9}$$

## 7.2 Dynamic Equation

The dynamic equation of motion is derived from the change in angular momentum of the satellite. An expression for the change in angular velocity, as a function of the applied torques is sought. The angular momentum $\boldsymbol{l}$, is given by

$$\begin{aligned} \boldsymbol{l} &= \sum_{i=1}^{k} \boldsymbol{l}_i \\ &= \sum_{i=1}^{k} \boldsymbol{r}_i \times m_i \boldsymbol{v}_i \end{aligned} \tag{7.10}$$

where $\boldsymbol{r}_i$ is the position of the $i$th particle with mass $m_i$ and velocity $\boldsymbol{v}_i$. Taking the time derivative of Equation 7.10, yields

$$\dot{\boldsymbol{l}} = \sum_{i=1}^{k} \left[\boldsymbol{v}_i \times m_i \boldsymbol{v}_i + \boldsymbol{r}_i \times m_i \boldsymbol{a}_i\right] \tag{7.11}$$

$\boldsymbol{a}_i$ being the acceleration of the $i$th particle. The first term under the summation of Equation 7.11 is a cross product of two parallel vectors, which is zero. Realizing that $m_i \boldsymbol{a}_i$ is the force acting on the $i$th particle, yields

$$\dot{\boldsymbol{l}} = \boldsymbol{n}_{\text{ext}} \tag{7.12}$$

where $\boldsymbol{n}_{\text{ext}}$ is the sum of external torques acting on the satellite. Equation 7.12 only holds if the internal torques sum up to zero [Wertz, 1978]. An expression of the derivative of the angular momentum in terms of the satellite's angular velocity is sought, in order to obtain the dynamic equation.

In the inertial ECI frame, denoted $I$, the angular momentum of the satellite can be expressed as a function of the angular velocity $\boldsymbol{\omega}$ and the moment of inertia matrix $\boldsymbol{J}$ of the satellite, by

$$\boldsymbol{l}^I = \boldsymbol{J}^I \boldsymbol{\omega} \tag{7.13}$$

The values of the inertia matrix $\boldsymbol{J}$ can be found in Annex I. Since the moment of inertia is more conveniently expressed in the SCB frame, denoted $B$, the angular momentum is found in the body frame. The attitude matrix $\boldsymbol{A}_I^B$, represents the rotation from the inertial frame to the satellite body frame, which is used to represent Equation 7.13 in the body frame, yielding

$$\boldsymbol{l}^B = \boldsymbol{A}_I^B \boldsymbol{l}^I \tag{7.14}$$

The derivative of $\boldsymbol{l}^B$ is given by

$$\begin{aligned} \dot{\boldsymbol{l}}^B &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{A}_I^B \boldsymbol{l}^I\right) \\ &= \dot{\boldsymbol{A}}_I^B \boldsymbol{l}^I + \boldsymbol{A}_I^B \dot{\boldsymbol{l}}^I \end{aligned} \tag{7.15}$$

In order to obtain an expression for $\dot{\boldsymbol{A}}_I^B$, consider the kinematic equation for rotating systems, which for the angular momentum vector $\boldsymbol{l}$ is

$$\left(\dot{\boldsymbol{l}}^I\right)^B = \dot{\boldsymbol{l}}^B + \boldsymbol{\omega} \times \boldsymbol{l}^B$$

$$\Updownarrow$$

$$\dot{\boldsymbol{l}}^B = \left(\dot{\boldsymbol{l}}^I\right)^B - \boldsymbol{\omega} \times \boldsymbol{l}^B \tag{7.16}$$

Since $\left(\dot{\boldsymbol{l}}^I\right)^B = \boldsymbol{A}_I^B \dot{\boldsymbol{l}}^I$, combining Equations 7.15 and 7.16, gives

$$\begin{aligned} \dot{\boldsymbol{A}}_I^B \boldsymbol{l}^I &= -\boldsymbol{\omega} \times \boldsymbol{l}^B \\ &= -\boldsymbol{\omega} \times \left[\boldsymbol{A}_I^B \boldsymbol{l}^I\right] \end{aligned} \tag{7.17}$$

Defining the cross product matrix function as

$$\mathbf{S}\left(\boldsymbol{v}\right) \triangleq \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{7.18}$$

where $\boldsymbol{v}$ is an arbitrary vector, Equation 7.17 is written

$$\dot{\boldsymbol{A}}_I^B \boldsymbol{l}^I = -\mathbf{S}\left(\boldsymbol{\omega}\right) \boldsymbol{A}_I^B \boldsymbol{l}^I \tag{7.19}$$

Since Equation 7.19 holds for all $\boldsymbol{l}^I$, the sought expression for the derivative of the attitude matrix is

$$\dot{\boldsymbol{A}}_I^B = -\mathbf{S}\left(\boldsymbol{\omega}\right) \boldsymbol{A}_I^B \tag{7.20}$$

Inserting Equation 7.20 into Equation 7.15, gives

$$\dot{\boldsymbol{l}}^B = -\mathbf{S}\left(\boldsymbol{\omega}\right) \boldsymbol{A}_I^B \boldsymbol{l}^I + \boldsymbol{A}_I^B \dot{\boldsymbol{l}}^I \tag{7.21}$$

Recalling from Equation 7.12, that the derivative of the angular momentum is the external torques and applying the attitude matrix rotations in Equation 7.21, yields

$$\dot{\boldsymbol{l}}^B = -\mathbf{S}\left(\boldsymbol{\omega}\right) \boldsymbol{l}^B + \boldsymbol{n}_{\text{ext}}^B \tag{7.22}$$

Finally the angular momentum is expressed in terms of the moment of inertia and the angular velocity, as given in Equation 7.13. Solving with respect to $\dot{\boldsymbol{\omega}}$, gives the sought nonlinear differential equation, written in the form

$$\dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}\left[\boldsymbol{n}_{\text{ext}} - \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}\right] \tag{7.23}$$

where the superscript of frame is left out, since all vectors and matrices are given in the SCB frame.

### Dynamics of Non-rigid Bodies

The dynamics given in Equation 7.23 only hold for rigid bodies. Since Rømer is equipped with reaction wheels, the body is non-rigid. Because of the angular momentum of the wheels, denoted $\boldsymbol{m}$, the total angular momentum is given by

$$\boldsymbol{l} = \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{m} \tag{7.24}$$

Using Equation 7.24 in the derivation of the dynamics above, yields

$$\dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}\left[\boldsymbol{n}_{\text{ext}} - \boldsymbol{\omega} \times \left[\boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{m}\right] - \dot{\boldsymbol{m}}\right] \tag{7.25}$$

where $\dot{\boldsymbol{m}}$ is the sum of the internal torques, generated by the momentum wheels. For simplification, the net angular momentum and the external torques are assumed to be negligible. The control torque of the wheels $\boldsymbol{n}_{\text{ctrl}}$, is defined as

$$\boldsymbol{n}_{\text{ctrl}} = -\dot{\boldsymbol{m}} \tag{7.26}$$

hence the dynamic equation becomes

$$\dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}\left[\boldsymbol{n}_{\text{ctrl}} - \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}\right] \tag{7.27}$$

## 7.3 System Equation

Combining the kinematic equation of Equation 7.9 and the dynamic equation of Equation 7.27, yields the non-linear differential equation

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\right) \\ &= \begin{bmatrix} \frac{1}{2}\boldsymbol{\Omega}\boldsymbol{q}\left(t\right) \\ \boldsymbol{J}^{-1}\left[\boldsymbol{n}_{\text{ctrl}} - \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}\right] \end{bmatrix} \end{aligned} \tag{7.28}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{bmatrix} \tag{7.29}$$

$$\boldsymbol{u} = \boldsymbol{n}_{\text{ctrl}} \tag{7.30}$$

defining the control torques as the input to the system.

# Part III

# Estimation

Chapter 8

# Kalman Filters

This chapter describes the Kalman filters considered for the Rømer estimator. The chapter is based on the theory of [Grewal and Andrews, 1993] and [Maybeck, 1982]. Kalman filters are preferred, since they can incorporate noise information, to produce statistically optimal estimates.

In order to estimate the state vector $\boldsymbol{x}$ from the noise inflicted measurements, a Kalman filter is designed. It is chosen to design a discrete-discrete Kalman filter, which can be directly implemented in an embedded system, like the Rømer on-board computer. A discrete-discrete filter uses a discrete model of the system with discrete measurement updates, since it is based on a discretization of the continuous differential equation and a discrete output equation.

The Kalman filter propagates a previous estimation of the states, using the state space equations, and corrects the propagation using measurements. In the following sections, the Kalman filter is described in general, followed by the derivation of a steady-state Kalman gain. Finally an extended filter is designed, which includes gyro bias estimation.

## 8.1 Linear Kalman Filter

The Kalman filter is based on a general state space description. For a discrete-discrete filter, both the model differential equation and the output equation are discrete. The model differential equation is written

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \boldsymbol{w}_k \tag{8.1}$$

where $\boldsymbol{w}_k$ is a vector of random variables, representing the process noise, which is assumed to be Gaussian white noise. The statistics of $\boldsymbol{w}_k$ are

$$\mathrm{E}\left(\boldsymbol{w}_k\right) = \mathbf{0} \tag{8.2}$$

$$\mathrm{E}\left(\boldsymbol{w}_k \boldsymbol{w}_{k+n}^{\mathrm{T}}\right) = \boldsymbol{Q}_k \Delta\left(n\right) \tag{8.3}$$

where $\mathrm{E}\left(\boldsymbol{w}_k\right)$ is the expected value function, $\Delta\left(n\right)$ is Kronecker's delta function, and $\boldsymbol{Q}_k$ is the strength of the process noise. Note the shorthand

$$\boldsymbol{x}_k \equiv \boldsymbol{x}\left(kT\right) \tag{8.4}$$

where $T$ is the time period between samples.

The discrete output equation is written

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{8.5}$$

where $\boldsymbol{v}_k$ is a vector of random variables, representing measurement noise, assumed to be Gaussian white noise with statistics

$$\mathrm{E}\left(\boldsymbol{v}_k\right) = \mathbf{0} \tag{8.6}$$

$$\mathrm{E}\left(\boldsymbol{v}_k \boldsymbol{v}_{k+n}^{\mathrm{T}}\right) = \boldsymbol{R}_k \Delta\left(n\right) \tag{8.7}$$

where $\boldsymbol{R}_k$ is the strength of the measurement noise.

Starting with an estimate at time $(kT)$, denoted $\hat{\boldsymbol{x}}_k$, the predictor of the filter calculates an a priori estimate using Equation 8.1. No noise is added since the expected value is zero. The propagated estimate, denoted $\hat{\boldsymbol{x}}_{k+1}^{-}$, is an estimate of the state at time $(k+1)\,T$, based only on the dynamics described by the difference equation of the system. Given the measurements at time $(k+1)\,T$, $\boldsymbol{z}_{k+1}$, the a priori estimate can be corrected into the a posteriori estimate at time $(k+1)\,T$, denoted $\hat{\boldsymbol{x}}_{k+1}$.

The covariance matrices of the a priori and a posteriori estimation errors at time $kT$, are defined as

$$\boldsymbol{P}_k^{-} = \mathrm{E}\left(\left[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^{-}\right]\left[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^{-}\right]^{\mathrm{T}}\right) \tag{8.8}$$

$$\boldsymbol{P}_k = \mathrm{E}\left(\left[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k\right]\left[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k\right]^{\mathrm{T}}\right) \tag{8.9}$$

The a priori estimate of $\boldsymbol{P}$ is propagated using the difference equation

$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{\Phi}_k \boldsymbol{P}_k \boldsymbol{\Phi}_k^{\mathrm{T}} + \boldsymbol{Q}_k \tag{8.10}$$

derived in [Grewal and Andrews, 1993]. As with the state estimate, the a posteriori covariance matrix $\boldsymbol{P}_{k+1}$ is obtained by correcting the a priori covariance matrix $\boldsymbol{P}_{k+1}^{-}$. An illustration of the Kalman filter is given in Figure 8.1. Note that the closed loop of the Kalman filter requires initial values of the state estimate and covariance of the estimation error, denoted $\hat{\boldsymbol{x}}_0$ and $\boldsymbol{P}_0$ respectively.

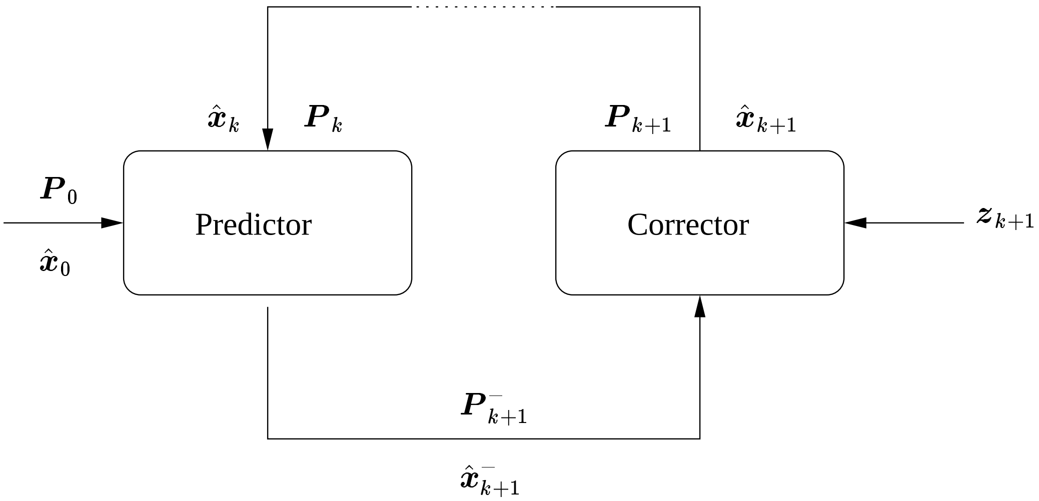


**Figure 8.1**: *Illustration of the Kalman filter.*

The corrector of the filter updates the predicted estimates using the output equation of Equation 8.5 and the measurement vector $\boldsymbol{z}_k$. The update of the state estimate is given by

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}_k \left[ \boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^{-} \right] \tag{8.11}$$

where $\boldsymbol{K}_k$ is the Kalman gain. The Kalman gain is calculated using

$$\boldsymbol{K}_k = \boldsymbol{P}_k^{-} \boldsymbol{H}_k^{\mathrm{T}} \left[ \boldsymbol{H}_k \boldsymbol{P}_k^{-} \boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k \right]^{-1} \tag{8.12}$$

It is important to note from Equation 8.11, that the Kalman gain is a weight factor, which weighs error in the measurement against the predicted state. From Equation 8.12 it is seen that if $\boldsymbol{R}_k$ approaches zero, the Kalman gain will increase. This results in a large update in Equation 8.11, which means that the new measurements are weighed higher than the predicted state. If $\boldsymbol{P}_k$ approaches zero, the Kalman gain will decrease, and the predicted state is weighed higher. When $\boldsymbol{R}_k$ and $\boldsymbol{P}_k$ increase, they will have the opposite effects on the Kalman gain. These effects are in agreement with the expected, when considering the interpretation of $\boldsymbol{R}_k$ and $\boldsymbol{P}_k$.

The corrector equation for the covariance matrix is

$$\boldsymbol{P}_k = \boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_k^- \tag{8.13}$$

where $\boldsymbol{K}_k$ is the Kalman gain of Equation 8.12. An alternate equation of the corrector, which is numerically more stable, is [Maybeck, 1982]

$$\boldsymbol{P}_k = \left[\mathbf{1} - \boldsymbol{K}_k \boldsymbol{H}_k\right] \boldsymbol{P}_k^- \left[\mathbf{1} - \boldsymbol{K}_k \boldsymbol{H}_k\right]^{\mathrm{T}} + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^{\mathrm{T}} \tag{8.14}$$

## 8.2 Steady-State Kalman Gain

The Kalman filter requires the computation of a Kalman gain between each sample. For linear time invariant systems, this gain will converge to a constant, named the steady-state Kalman gain, which can be pre-calculated. In the following, the steady-state Kalman gain is derived.

From Equation 8.12, it is seen that $\boldsymbol{K}_k$ varies with the covariance matrix prediction. For time invariant systems, $\boldsymbol{H}_k$ and $\boldsymbol{R}_k$ are constant, hence if a steady-state solution of the covariance matrix can be found, this solution can be used to calculate the steady-state Kalman gain.

In order to find the steady-state covariance matrix, denoted $\boldsymbol{P}_\infty$, the equations of the predictor and corrector are combined.

The propagation of the covariance matrix is accomplished using the discrete model equation of Equation 8.10. The corrector equation is a correction of the propagation, which can be inserted directly Equation 8.10, as

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{\Phi} \left[\boldsymbol{P}_k - \boldsymbol{K}_k \boldsymbol{H} \boldsymbol{P}_k\right] \boldsymbol{\Phi}^{\mathrm{T}} + \boldsymbol{Q} \tag{8.15}$$

assuming that the process noise is constant for the time invariant system. Note that the subscript $k$ is left out for the time invariant matrices.

The steady-state of $\boldsymbol{P}$ will result in zero dynamics, which means that the propagation equation of $\boldsymbol{P}$ in Equation 8.15 results in an unchanged variance. Inserting Equation 8.12 in Equation 8.15, this property is written as

$$\boldsymbol{P}_k = \boldsymbol{\Phi}\left[\boldsymbol{P}_k - \boldsymbol{P}_k^- \boldsymbol{H}^{\mathrm{T}}\left[\boldsymbol{H}\boldsymbol{P}_k^-\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\right]^{-1}\boldsymbol{H}\boldsymbol{P}_k\right]\boldsymbol{\Phi}^{\mathrm{T}} + \boldsymbol{Q} \qquad (8.16)$$

assuming that the measurement noise is constant. Re-organizing Equation 8.16 gives

$$\boldsymbol{\Phi}\boldsymbol{P}_\infty\boldsymbol{\Phi}^{\mathrm{T}} - \boldsymbol{P}_\infty - \boldsymbol{\Phi}\boldsymbol{P}_\infty\boldsymbol{H}^{\mathrm{T}}\left[\boldsymbol{H}\boldsymbol{P}_\infty\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\right]^{-1}\boldsymbol{H}\boldsymbol{P}_\infty\boldsymbol{\Phi}^{\mathrm{T}} + \boldsymbol{Q} = \boldsymbol{0} \qquad (8.17)$$

Equation 8.17 takes the form of an Algebraic Riccati Equation. The solution to this Riccati equation is the steady-state covariance matrix $\boldsymbol{P}_\infty$. The matrix is obtained using the MATLAB function *dare* (Discrete Algebraic Riccati Equation), hence the analytic solution will not be discussed in this project.

Given $\boldsymbol{P}_\infty$ the steady-state Kalman gain $\boldsymbol{K}_\infty$ can be calculated using Equation 8.12, giving

$$\boldsymbol{K}_\infty = \boldsymbol{P}_\infty\boldsymbol{H}^{\mathrm{T}}\left[\boldsymbol{H}\boldsymbol{P}_\infty\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\right]^{-1} \qquad (8.18)$$

Recall that $\boldsymbol{H}$ and $\boldsymbol{R}$ are time invariant, hence $\boldsymbol{K}_\infty$ can be calculated once, and used in the corrector as a constant. The Kalman filter is then simplified to two equations. The propagation of the state, using the discrete system model equation, and the corrector:

$$\boldsymbol{x}_k^- = \boldsymbol{\Phi}\boldsymbol{x}_{k-1} + \boldsymbol{\Gamma}\boldsymbol{u}_{k-1} \qquad (8.19)$$

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_\infty\left[\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-\right] \qquad (8.20)$$

where $\boldsymbol{K}_\infty$ is calculated using Equation 8.18. When the state error is converged, the covariance of the state estimate is $\boldsymbol{P}_\infty$.

## 8.3 Extended Kalman Filter

The previous two filters are based on a linear model of the systems. When a system is described by a non-linear model equation, the equations must be linearized around a nominal state. If the nominal state is constant, then the previous filters can be applied on the linearized system.

In some cases, however, a linearization around a single trajectory is inadequate, especially if it is not guaranteed that the state of the system is always close to the nominal values. In these situations an extended Kalman filter is applied, which re-linearizes the system around the current estimate.

The non-linear system, is described by the differential equation

$$\dot{\boldsymbol{x}}\left(t\right) = \boldsymbol{f}\left(\boldsymbol{x}\left(t\right), \boldsymbol{u}\left(t\right), t\right) + \boldsymbol{w}\left(t\right) \tag{8.21}$$

where $\boldsymbol{w}\left(t\right)$ is the continuous model noise. The continuous Gaussian white noise has the statistics

$$\mathrm{E}\left(\boldsymbol{w}\left(t\right)\right) = \boldsymbol{0} \tag{8.22}$$

$$\mathrm{E}\left(\boldsymbol{w}\left(t\right)\boldsymbol{w}\left(t+\tau\right)^{\mathrm{T}}\right) = \boldsymbol{Q}\left(t\right)\delta\left(\tau\right) \tag{8.23}$$

where $\delta\left(\tau\right)$ is Dirac's delta function, and $\boldsymbol{Q}\left(t\right)$ is the noise strength. The non-linear output equation is given by

$$\boldsymbol{z}_k = \boldsymbol{h}_k\left(\boldsymbol{x}_k\right) + \boldsymbol{v}_k \tag{8.24}$$

where $\boldsymbol{v}_k$ is the measurement noise with the same statistics as in Equation 8.5.

Using a Taylor expansion of Equation 8.21 around a nominal trajectory $\bar{\boldsymbol{x}}$ and a nominal input $\bar{\boldsymbol{u}}$, yields

$$\begin{aligned}\dot{\boldsymbol{x}}\left(t\right) =& \boldsymbol{f}\left(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, t\right) + \left.\frac{\partial \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, t\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}, \boldsymbol{u}=\bar{\boldsymbol{u}}} \\ &+ \left.\frac{\partial \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, t\right)}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}, \boldsymbol{u}=\bar{\boldsymbol{u}}} \left[\boldsymbol{u}\left(t\right) - \bar{\boldsymbol{u}}\right] + \text{h.o.t.} + \boldsymbol{w}\left(t\right)\end{aligned} \tag{8.25}$$

where h.o.t. are the higher order terms. Considering the perturbation of the state from the nominal trajectory, denoted $\tilde{\boldsymbol{x}} = \boldsymbol{x}\left(t\right) - \bar{\boldsymbol{x}}$ and a perturbed input $\tilde{\boldsymbol{u}} = \boldsymbol{u}\left(t\right) - \bar{\boldsymbol{u}}$, Equation 8.25 is written

$$\dot{\tilde{\boldsymbol{x}}}\left(t\right) = \left.\frac{\partial \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, t\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}, \boldsymbol{u}=\bar{\boldsymbol{u}}} \tilde{\boldsymbol{x}}\left(t\right) + \left.\frac{\partial \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, t\right)}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}, \boldsymbol{u}=\bar{\boldsymbol{u}}} \tilde{\boldsymbol{u}}\left(t\right) + \boldsymbol{w}\left(t\right) \tag{8.26}$$

assuming that the higher order terms are negligible. An approximated linearized system equation of the state perturbation can now be expressed as

$$\dot{\tilde{\boldsymbol{x}}}\left(t\right)=\boldsymbol{F}\left(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},t\right)\tilde{\boldsymbol{x}}+\boldsymbol{G}\left(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},t\right)\tilde{\boldsymbol{u}}+\boldsymbol{w}\left(t\right) \tag{8.27}$$

where $\boldsymbol{F}\left(\bar{\boldsymbol{x}},\boldsymbol{u}\left(t\right),t\right)$ and $\boldsymbol{G}\left(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},t\right)$ are the Jacobians of $\boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{u}\left(t\right),t\right)$ with respect to the state and the input, given by

$$\boldsymbol{F}\left(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},t\right)=\left.\frac{\partial\boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{u},t\right)}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}},\boldsymbol{u}=\bar{\boldsymbol{u}}} \tag{8.28}$$

$$\boldsymbol{G}\left(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},t\right)=\left.\frac{\partial\boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{u},t\right)}{\partial\boldsymbol{u}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}},\boldsymbol{u}=\bar{\boldsymbol{u}}} \tag{8.29}$$

In a similar manner, a linear expression of the output equation in Equation 8.24 is found to be

$$\tilde{\boldsymbol{z}}_k=\boldsymbol{H}_k\left(\bar{\boldsymbol{x}}\right)\tilde{\boldsymbol{x}}_k+\boldsymbol{v}_k \tag{8.30}$$

where $\boldsymbol{H}\left(\bar{\boldsymbol{x}},t\right)$ is the Jacobian of $\boldsymbol{h}_k\left(\boldsymbol{x}_k\right)$, given by

$$\boldsymbol{H}_k\left(\bar{\boldsymbol{x}}\right)=\left.\frac{\partial\boldsymbol{h}_k\left(\boldsymbol{x}\right)}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{8.31}$$

The corrector equation is given by

$$\hat{\tilde{\boldsymbol{x}}}_k=\boldsymbol{K}_k\left[\boldsymbol{z}_k-\bar{\boldsymbol{z}}\right] \tag{8.32}$$

where $\bar{\boldsymbol{z}}$ is given by

$$\bar{\boldsymbol{z}}=\boldsymbol{h}_k\left(\bar{\boldsymbol{x}}\right) \tag{8.33}$$

The estimate of the state at time $kT$ is then given by

$$\hat{\boldsymbol{x}}_k=\bar{\boldsymbol{x}}+\hat{\tilde{\boldsymbol{x}}}_k \tag{8.34}$$

The covariance equations are the same as for the linear filter, using the linearized and discretized matrices for $\boldsymbol{\Phi}$ and $\boldsymbol{H}$.

Up to this point, the nominal state $\bar{\boldsymbol{x}}$ has been written as constant in time. This is the form of a linearized filter. The difference to the extended filter is that the nominal state is chosen to be the current estimate, hence all linearizations are done at each time step. Obviously this is a computational disadvantage. However when the true state differs significantly from the nominal state, e.g. in situations where a nominal trajectory may not be determined in advance, re-linearization is necessary.

In the following the the equations of the linearized filter are summarized, but with a re-linearization around the current estimate, hence the equations of the extended Kalman filter are stated.

If the state estimate at time $(k-1)\,T$ is denoted $\hat{\boldsymbol{x}}_{k-1}$, the system gradients are calculated around the estimate using

$$\boldsymbol{F}\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}, t\right) = \left.\frac{\partial \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, t\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}=\boldsymbol{u}_{k-1}} \qquad (8.35)$$

$$\boldsymbol{H}\left(\hat{\boldsymbol{x}}_{k-1}, t\right) = \left.\frac{\partial \boldsymbol{h}\left(\boldsymbol{x}, t\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}} \qquad (8.36)$$

The nominal value of the input is chosen to be the last measured input. The calculation of $\boldsymbol{G}$ is left out, since the extended Kalman filter uses the non-linear differential equation to propagate the state. Hence $\boldsymbol{G}$ is not used.

The state estimate is propagated one sampling period of time $T$, from $\hat{\boldsymbol{x}}_{k-1}$ to the a priori estimate $\hat{\boldsymbol{x}}_k$ of time $k$, using the non-linear differential equation

$$\dot{\boldsymbol{x}}\left(t\right) = \boldsymbol{f}\left(\boldsymbol{x}\left(t\right), \boldsymbol{u}\left(t\right), t\right) \qquad (8.37)$$

The propagation of the covariance $\boldsymbol{P}_{k-1}$ of the estimate $\hat{\boldsymbol{x}}_{k-1}$ may be done using the discrete equation

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_{k-1}\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}\right)\boldsymbol{P}_{k-1}\boldsymbol{\Phi}_{k-1}\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}\right)^{\mathrm{T}} + \boldsymbol{Q}_{k-1} \qquad (8.38)$$

where $\boldsymbol{\Phi}_{k-1}\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}\right)$ is the the discrete equivalent of $\boldsymbol{F}\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}, (k-1)\,T\right)$.

The corrector equations which update the a priori estimate $\hat{\boldsymbol{x}}_k^-$ and covariance $\boldsymbol{P}_k^-$, are given by

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left[\boldsymbol{z}_k - \boldsymbol{h}_k\left(\hat{\boldsymbol{x}}_k^-\right)\right] \qquad (8.39)$$

$$\boldsymbol{P}_k = \left[\boldsymbol{1} - \boldsymbol{K}_k\boldsymbol{H}_k\left(\hat{\boldsymbol{x}}_k^-\right)\right]\boldsymbol{P}_k^-\left[\boldsymbol{1} - \boldsymbol{K}_k\boldsymbol{H}_k\left(\hat{\boldsymbol{x}}_k^-\right)\right]^{\mathrm{T}} + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^{\mathrm{T}} \qquad (8.40)$$

where the Kalman gain $\boldsymbol{K}_k$ is calculated, using

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k \left(\hat{\boldsymbol{x}}_k^-\right)^{\mathrm{T}} \left[\boldsymbol{H}_k \left(\hat{\boldsymbol{x}}_k^-\right) \boldsymbol{P}_k^- \boldsymbol{H}_k \left(\hat{\boldsymbol{x}}_k^-\right)^{\mathrm{T}} + \boldsymbol{R}_k\right]^{-1} \tag{8.41}$$

These are the equations of the extended Kalman filter. In the case of Rømer, the equations simplify due to a linear output equation. These equations are shown in the next chapter, where the filter used for the attitude estimate of Rømer is derived.

Chapter 9

# Rømer Estimator Design

This chapter describes the estimator designed for Rømer, based on the Kalman filters described in the previous chapter. Some specific alternation to the theory is necessary, due to the use of quaternions in the state. In addition the system model is expanded to include gyro biases, in order to estimate them. The filter types used are discussed, and different implementation strategies are considered, that deal with low frequency and delayed star tracker measurement.

## 9.1 Choice of Filter

In the previous chapter, three Kalman filters were described. A linear filter, a linear filter with steady-state Kalman gain, and an extended Kalman filter.

The linear filters can be applied if the non-linear system is linearized around a nominal state, which is done analytically in Appendix C. The steady-state filter is derived because of its computational advantage. However when the true state diverges from the nominal state, a re-linearization is necessary, which is done in the extended filter.

When Rømer operates in coarse mode, re-orientation of the satellite must be possible. Hence the re-linearization is necessary, which suggests the use of an extended filter. When the MONS Telescope has been pointed towards a target star, and the satellite changes to fine mode, the state is close to the nominal trajectory. If it can be shown that a linear filter can meet the requirements in fine mode, a steady-state filter is preferred, in order to minimize the load of the on-board computer.

An advantage of this strategy, is that the on-board computer is loaded in coarse mode, when no science is done. When fine mode commences, a steady-state filter frees computation time for the science payload, which allows a larger number of parallel science to be executed.

## 9.2 Extended Kalman Filter Design

The extended Kalman filter is used to estimate the attitude and angular rate of the satellite. In addition the gyro bias is included in the state. This improves the estimate of the angular rate and thereby the attitude estimate. The system state $\boldsymbol{x}$ is defined as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \\ \boldsymbol{\beta} \end{bmatrix} \tag{9.1}$$

and the measurement vector $\boldsymbol{z}$ as

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{m}} \\ \boldsymbol{\omega}_{\mathrm{m}} + \boldsymbol{\beta}_{\mathrm{m}} \end{bmatrix} \tag{9.2}$$

Note that the gyro biases $\boldsymbol{\beta}$ are actually the sum of the bias and drift of the gyros, and are defined in the direction of the body axes. Hence it is not the bias of each gyro which is estimated. Instead the equivalent biases on the 1st, 2nd, and 3rd element of the angular rate in the body frame are estimated.

The input vector $\boldsymbol{u}$ is defined as

$$\boldsymbol{u} = \boldsymbol{n}_{\mathrm{ctrl}} \tag{9.3}$$

where $\boldsymbol{n}_{\mathrm{ctrl}}$ is the torque generated by the controller. Note that time dependency notation is left out. The non-linear differential equation is

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\right) \\ \boldsymbol{0}_{3\times 3} \end{bmatrix} \tag{9.4}$$

where

$$\boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\right) = \begin{bmatrix} \frac{1}{2}\boldsymbol{\Omega}\boldsymbol{q}\left(t\right) \\ \boldsymbol{J}^{-1}\left[\boldsymbol{n}_{\mathrm{ctrl}} - \boldsymbol{\omega}_{\mathrm{s}} \times \boldsymbol{J}\boldsymbol{\omega}_{\mathrm{s}}\right] \end{bmatrix} \tag{9.5}$$

derived in Chapter 7. The dynamics of the gyro biases are zero, since the expected value of the drift derivative is zero.

The estimated measurement $\hat{\boldsymbol{z}}_k$ is given by the output equation

$$\hat{\boldsymbol{z}}_k = \boldsymbol{H}\hat{\boldsymbol{x}}_k \tag{9.6}$$

Note that the output equation is linear and time invariant. The matrix $\boldsymbol{H}$ is given by

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix} \tag{9.7}$$

where the identity matrices $\mathbf{1}$ and the zero matrices $\mathbf{0}$ all are $3$ by $3$ matrices. The resulting output estimate is

$$\hat{\boldsymbol{z}} = \begin{bmatrix} \hat{\boldsymbol{q}} \\ \hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\beta}} \end{bmatrix} \tag{9.8}$$

Given the measurement vector $\boldsymbol{z}_k$, the measurement residual $\tilde{\boldsymbol{z}}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k$ is calculated using quaternion multiplication on the first four elements of $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$. The rotation $\tilde{\boldsymbol{q}}$ between the rotations $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$ is given by

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_{\mathrm{m}}\hat{\boldsymbol{q}}^* \tag{9.9}$$

Hence the measurement residual $\hat{\boldsymbol{z}}_k$ is in fact given by

$$\tilde{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{m}}\hat{\boldsymbol{q}}^* \\ \boldsymbol{\omega}_{\mathrm{m}} - \hat{\boldsymbol{\omega}} + \boldsymbol{\beta}_{\mathrm{m}} - \hat{\boldsymbol{\beta}} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{q}} \\ \tilde{\boldsymbol{\omega}} + \tilde{\boldsymbol{\beta}} \end{bmatrix} \tag{9.10}$$

but the notation of the residual will still be $\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k$.

The measurement residual contains seven states, of the ten states in the system state vector. A problem of the covariance arises when using four elements of the quaternion in the state. Using four states for the quaternion causes the covariance matrix $\boldsymbol{P}$ to be singular, [Lefferts et al., 1982]. In order to overcome this problem, the state of the measurement residual is reduced by one dimension, by leaving out the scalar element of the quaternion. Hence the residual measurement state vector reduces to the six states

$$\rho\tilde{\boldsymbol{z}} = \begin{bmatrix} \tilde{\boldsymbol{q}}_{1:3} \\ \tilde{\boldsymbol{\omega}} + \tilde{\boldsymbol{\beta}} \end{bmatrix} \tag{9.11}$$

As a consequence, the covariance matrix $\boldsymbol{P}$ reduces to a $9$ by $9$ matrix, containing the covariance of only nine states of the state vector $\boldsymbol{x}$. The 10th state which is not included is the quaternion scalar.

The correction term $\rho\hat{\hat{\boldsymbol{x}}}_k$, which is the correction of the a priori estimate, is calculated using the corrector equation

$$\rho\hat{\hat{\boldsymbol{x}}}_k = \boldsymbol{K}_k \rho\tilde{\boldsymbol{z}} \tag{9.12}$$

where the Kalman gain $\boldsymbol{K}$ is calculated using the equation

$$\boldsymbol{K} = \boldsymbol{P}^{-}\boldsymbol{H}^{\mathrm{T}}\left[\boldsymbol{H}\boldsymbol{P}^{-}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\right]^{-1} \tag{9.13}$$

Note that the reduction of the measurement state and the covariance matrix, results in a reduced state correction vector. This vector is expanded to the ten states of the filter, before updating the a priori estimate $\hat{\boldsymbol{x}}_k^-$.

The quaternion state of the state vector is expanded from three states to four states by setting the fourth element to one, and normalizing the quaternion. This is done under the assumption that the correction term $\rho\hat{\hat{\boldsymbol{x}}}_k$ is small. Using small angle approximation on the fourth element, yields the constant value of one. The expansion of the correction vector is written as

$$\hat{\hat{\boldsymbol{x}}} = \begin{bmatrix} \hat{\hat{\boldsymbol{q}}}_{1:3} / \sqrt{1 + \left|\hat{\hat{\boldsymbol{q}}}_{1:3}\right|^2} \\ 1 / \sqrt{1 + \left|\hat{\hat{\boldsymbol{q}}}_{1:3}\right|^2} \\ \hat{\hat{\boldsymbol{\omega}}} \\ \hat{\hat{\boldsymbol{\beta}}} \end{bmatrix} \tag{9.14}$$

The a posteriori estimate $\hat{\boldsymbol{x}}_k$ is given by

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \hat{\hat{\boldsymbol{x}}}_k \tag{9.15}$$

Due to the use of quaternions in the state, quaternion multiplication is used for the first four elements of the state, as with the measurement residual. The rotation estimate $\hat{\boldsymbol{q}}$ given by the a priori rotation estimate $\hat{\boldsymbol{q}}^-$ and a correction $\hat{\hat{\boldsymbol{q}}}$ is calculated using

$$\hat{\boldsymbol{q}} = \hat{\hat{\boldsymbol{q}}}\hat{\boldsymbol{q}}^- \tag{9.16}$$

Hence the a posteriori estimate is given by

$$\hat{\boldsymbol{x}}_k = \begin{bmatrix} \hat{\tilde{\boldsymbol{q}}}\hat{\boldsymbol{q}}^- \\ \hat{\boldsymbol{\omega}}^- + \hat{\tilde{\boldsymbol{\omega}}} \\ \hat{\boldsymbol{\beta}}^- + \hat{\tilde{\boldsymbol{\beta}}} \end{bmatrix} \tag{9.17}$$

but the notation is left unchanged, as with the measurement residual.

A block diagram of the extended Kalman filter is given in Figure 9.1, showing the propagator and the corrector using the reduced state.

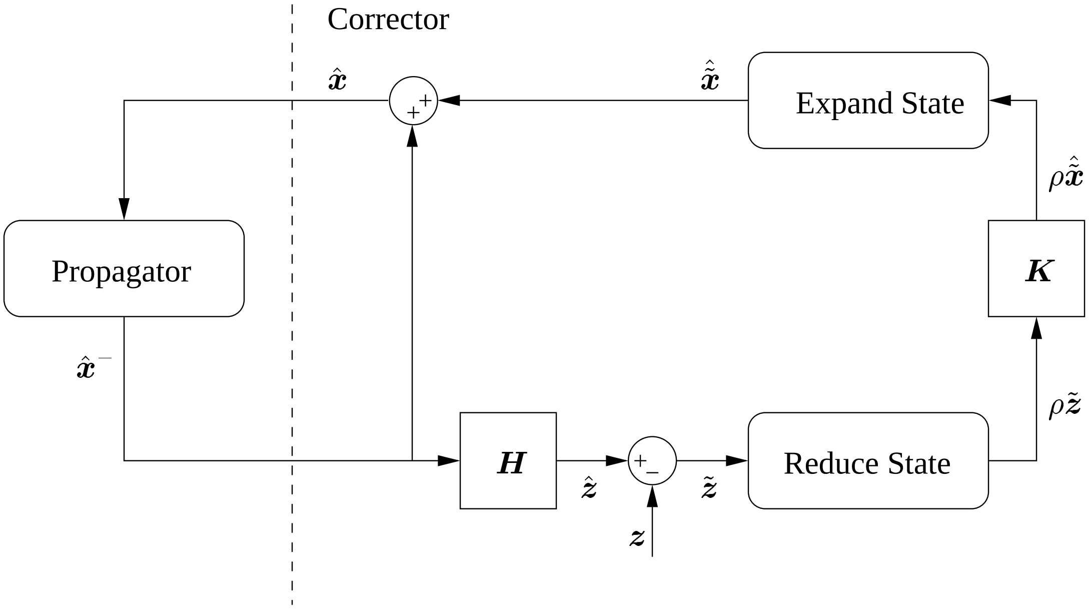


**Figure 9.1**: *Block diagram of the state reduction and expansion in the corrector of the extended Kalman filter.*

The covariance is calculated using the linearized system matrix $\boldsymbol{F}$, given by

$$\boldsymbol{F}\left(\hat{\boldsymbol{x}}\right) = \begin{bmatrix} -\mathbf{S}\left(\hat{\boldsymbol{\omega}}\right) & \frac{1}{2}\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}^{-1}\left[\mathbf{S}\left(\boldsymbol{J}\hat{\boldsymbol{\omega}}\right) - \mathbf{S}\left(\hat{\boldsymbol{\omega}}\right)\boldsymbol{J}\right] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{9.18}$$

derived in Appendix C. All matrices are $3$ by $3$ matrices. The system is linearized around the current estimate, and is used in the discrete equivalent, denoted $\boldsymbol{\Phi}_k\left(\hat{\boldsymbol{x}}_k\right)$. This matrix is used in the covariance propagation equation in Equation 8.38, and the correction of $\boldsymbol{P}$ is done using Equation 8.40. No changes apply, due the use of quaternions, except for the reduction from ten to nine states, causing $\boldsymbol{P}$ to be a $9$ by $9$ matrix.

## 9.3 Linear Steady-State Kalman Filter

The steady-state filter is a linear filter, hence the system is linearized around a nominal state. In the extended filter, the nominal state is the current estimate, which varies in time. In the steady-state filter, the nominal value is the reference of the controller, and is constant.

When using a linear filter, the estimation of the gyro biases is left out, since the algebraic Riccati equation has no solution. Instead, the estimated biases from the extended filter in coarse mode may be applied in the linear filter. This solution only holds if the true bias is close to the estimated bias, which is held constant in the linear filter, which must be shown trough simulation.

Using a constant nominal state, yields the following linearized system matrix

$$\boldsymbol{F}\left(\bar{\boldsymbol{x}}\right) = \begin{bmatrix} -\mathbf{S}\left(\bar{\boldsymbol{\omega}}\right) & \frac{1}{2}\mathbf{1} \\ \mathbf{0} & \boldsymbol{J}^{-1}\left[\mathbf{S}\left(\boldsymbol{J}\bar{\boldsymbol{\omega}}\right) - \mathbf{S}\left(\bar{\boldsymbol{\omega}}\right)\boldsymbol{J}\right] \end{bmatrix} \tag{9.19}$$

where $\bar{\boldsymbol{x}}$ is the nominal state. Since $\boldsymbol{F}$ is constant, the covariance matrix $\boldsymbol{P}$ will converge to a steady-state covariance, which can be calculated in the initialization of the filter, by solving the algebraic Riccati equation in Equation 8.17. Given the solution $\boldsymbol{P}_\infty$, the steady state Kalman gain $\boldsymbol{K}_\infty$ is calculated using Equation 8.18.

Using the equations of the extended Kalman filter with the steady-state Kalman gain and a reduced state vector, where the gyro biases are left out, yields the steady-state filter used in fine mode. The equations of the covariance are left out in the steady-state filter, since the steady-state covariance is the covariance of the state, after the filter has converged.

The estimators given so far, do not handle the delay of the quaternion measurement by the star trackers, and updates the attitude and angular rate estimates simultaneously. The filters can be improved by updating the rate estimate faster than the limit of the attitude update. These issues will be discussed in the following.

## 9.4 Multi-rate Filter

The filters designed in the previous sections make use of measurements from the star trackers and rate gyros. As mentioned in Chapter 2, the CCD readout of the CHU takes 332ms. After the CCD readout, the attitude estimate is calculated, which delays the measurement further. This means that the update of the filter can run at maximum 2Hz, assuming that the attitude calculation takes a minimum of 170ms.

The gyros however can be sampled at a higher rate. This can be accomplished by a multi-rate filter, where the ordinary filter corrector is split into two parts. One for the rate update, and another for the attitude update. Hence the rate estimate can be updated at higher frequency, and when an attitude measurement is available, the attitude corrector is used.

Instead of implementing two correctors, the standard filter can be used. When an attitude estimate is unavailable, the measurement residual $\tilde{\boldsymbol{z}}$ of the attitude is simply set to zero. This will result that the correction is solely based on the angular rate measurement residual. The calculation of the correction term of Equation 9.12 becomes

$$\rho\hat{\tilde{\boldsymbol{x}}}_k = \boldsymbol{K}_k \begin{bmatrix} \mathbf{0}_{3\times1} \\ \tilde{\boldsymbol{\omega}} \end{bmatrix} \tag{9.20}$$

The above can also be interpreted as using the estimate of the attitude as the measurement, when a true measurement is unavailable, which will result in a measurement residual of zero. By using this approach, all the equations of the filter are the same, with the only difference that the measurement vector $\boldsymbol{z}_k$ contains the estimated attitude when an attitude measurement is unavailable.

Since the attitude measurement is not used, because it is unavailable, the correction of the covariance is

$$\boldsymbol{P}_k = \boldsymbol{P}_k^- - \boldsymbol{K}\boldsymbol{H}_\mathrm{g}\boldsymbol{P}_k^- \tag{9.21}$$

where

$$\boldsymbol{H}_\mathrm{g} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix} \tag{9.22}$$

In an implementation, the calculations where the attitude measurement residual is set to zero, only the rate estimate equations of the corrector are calculated, i.e. implementing two correctors. However, describing the multi-rate filter in a single equation, is an advantage when handling the attitude measurement delay, discussed later in this chapter.

A different approach is to use a low pass filter on the gyro measurements, when provided to the Kalman filter. The bandwidth of the low pass filter, should be lower the bandwidth of the system, including controller. This will minimize the noise on the gyro measurement.

The choice of solution depends on the handling of the sensing delay of the star trackers.

## 9.5 Sensing Delay

The delayed measurement of the attitude requires causes the update of the corrector to be suboptimal, since the estimated output is different in time than the measurement. Four solutions are considered in this section

- Measurement propagation
- Parallel re-calculation
- Delayed correction
- On demand estimation

The four algorithms are presented in the following.

### Measurement Propagation

This method is based on a propagation of the measurement, to the time of the next update. Figure 9.2 illustrates the time line of the events of the system. The controller requires an estimate at time $t_k$ and $t_{k+1}$ etc., which are separated in time by $T$ seconds. At time $t_d$, the filter receives a delayed attitude measurement representing the attitude $\boldsymbol{q}_m$ of the star tracker at time $t_m$.

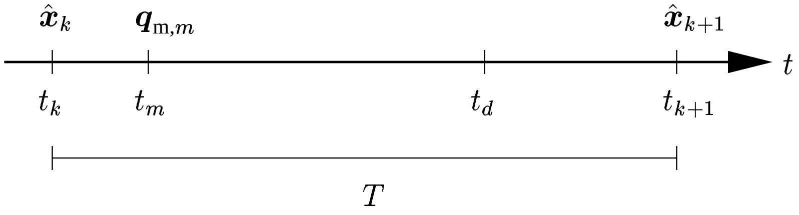


**Figure 9.2**: *Time line of a delayed measurement within a single sample period of the controller.*

The delayed measurement can update the state in two different ways:

1. The state $\hat{\boldsymbol{x}}_k$ is propagated $t_m - t_k$ seconds to $\hat{\boldsymbol{x}}_m^-$, where it is updated with the measurement to yield the a posteriori estimate $\hat{\boldsymbol{x}}_m$. The a posteriori is then propagated $t_{k+1} - t_m$ seconds to time $t_{k+1}$, resulting in the estimate $\hat{\boldsymbol{x}}_{k+1}$.
2. The measurement $\boldsymbol{q}_{\mathrm{m},m}$ is propagated $t_{k+1} - t_m$ seconds to time $t_{k+1}$, which is used to update the a priori estimate $\hat{\boldsymbol{x}}_{k+1}^-$ to the a posteriori estimate $\hat{\boldsymbol{x}}_{k+1}$.

Of the two methods, the second method has proven to produce the best results in [Azor et al., 2001]. If the measurement is delayed beyond $t_{k+1}$, the measurement must be propagated to the following update time after the time the delayed measurement is received at time $t_d$. The measurement should be propagated using the estimated angular rate, at each sampling time of the controller. This means that the propagation of $\boldsymbol{q}_{\mathrm{m},m}$ to $\boldsymbol{q}_{\mathrm{m},k\,+\,1}$ is done using $\hat{\boldsymbol{\omega}}_k$. If $t_d$ is beyond $t_{k+1}$ then the measurement is propagated to $\boldsymbol{q}_{\mathrm{m},k\,+\,2}$ using $\hat{\boldsymbol{\omega}}_{k+1}$. In other words, the best estimate of $\boldsymbol{\omega}$ is always used to propagate $\boldsymbol{q}$.

Note that this algorithm does not handle the update of $\hat{\boldsymbol{x}}^-_{k+1}$ in the absence of $\boldsymbol{q}_{\mathrm{m}}$ in case it is delayed beyond the sampling time $t_{k+1}$. This can be accommodated using the multi-rate filter discussed in the previous section.

## Parallel Re-calculation

The idea of the parallel re-calculation algorithm is to re-calculate the filter estimates, from the time where the delayed measurement should have been incorporated. Since this is a time consuming process, it is done in a parallel process, until the re-calculated estimate is ready, at which point it is used in the filter. The algorithm is designed for the multi-rate filter, where the rate estimate is updated faster that the attitude estimate.

Figure 9.3 illustrates the time line of the events in the algorithm.

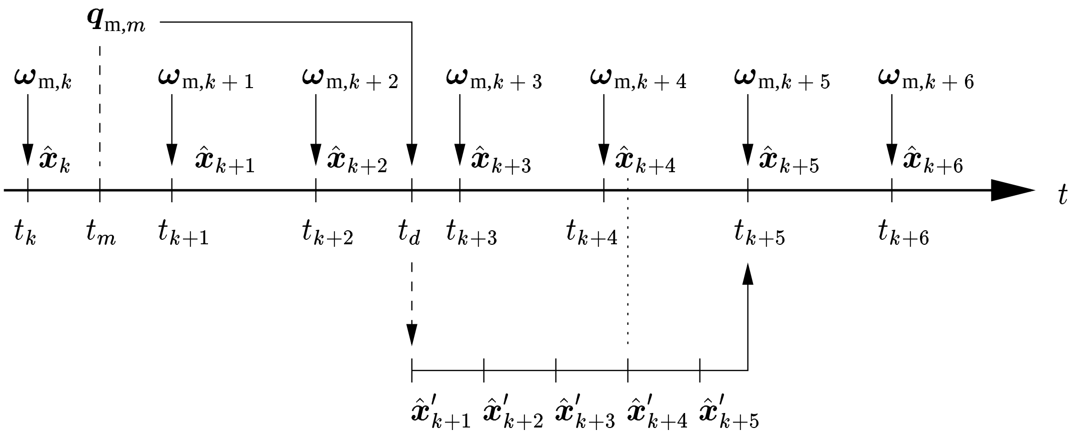

$\boldsymbol{q}_{\mathrm{m},m}$
$\boldsymbol{\omega}_{\mathrm{m},k}$
$\boldsymbol{\omega}_{\mathrm{m},k\,+\,1}$
$\boldsymbol{\omega}_{\mathrm{m},k\,+\,2}$
$\boldsymbol{\omega}_{\mathrm{m},k\,+\,3}$
$\boldsymbol{\omega}_{\mathrm{m},k\,+\,4}$
$\boldsymbol{\omega}_{\mathrm{m},k\,+\,5}$
$\boldsymbol{\omega}_{\mathrm{m},k\,+\,6}$
$\hat{\boldsymbol{x}}_k$
$\hat{\boldsymbol{x}}_{k+1}$
$\hat{\boldsymbol{x}}_{k+2}$
$\hat{\boldsymbol{x}}_{k+3}$
$\hat{\boldsymbol{x}}_{k+4}$
$\hat{\boldsymbol{x}}_{k+5}$
$\hat{\boldsymbol{x}}_{k+6}$
$t$
$t_k$
$t_m$
$t_{k+1}$
$t_{k+2}$
$t_d$
$t_{k+3}$
$t_{k+4}$
$t_{k+5}$
$t_{k+6}$
$\hat{\boldsymbol{x}}'_{k+1}$
$\hat{\boldsymbol{x}}'_{k+2}$
$\hat{\boldsymbol{x}}'_{k+3}$
$\hat{\boldsymbol{x}}'_{k+4}$
$\hat{\boldsymbol{x}}'_{k+5}$

**Figure 9.3**: *Illustration of the parallel re-calculation algorithm time line.*

It is seen in Figure 9.3 that at each sampling point $t_{k+i},\ i = 0..6$, the gyro measurement $\boldsymbol{\omega}_{\mathrm{m},k\,+\,i}$ is used to update the a priori estimate, to produce the a posteriori estimate $\hat{\boldsymbol{x}}_{k+i}$. At time $t_d$ a delayed attitude measurement is received by the filter, which was measured at time $t_m$, denoted $\boldsymbol{q}_{\mathrm{m},m}$.

The filter has already updated the estimate at time $t_{k+1}$ and $t_{k+2}$. A parallel process is initiated, in which the updates are recalculated, from the time $t_{k+1}$ where the measurement would have been incorporated, if no delay had occured. The parallel process calculates estimates up to $\hat{\boldsymbol{x}}'_{k+5}$ at which time the estimate is passed, and used in the primary filter. Note that the parallel process does not need to run realtime, hence it catches up with the primary filter.

The update at time $t_k + 1$ is accomplished using the measurement propagation algorithm.

The algorithm suffers from computational overload, because two filters are run parallel in some time intervals. An alternative to executing two filters, is using update correction.

## Update Correction

The update correction algorithm updates the estimate following the delayed measurement, by correcting for the error in the estimate, caused by the delayed update. The algorithm spawns from [Alexander, 1991]. Figure 9.4 illustrates the time line of a delayed and an undelayed estimate. The delay time has been minimized in order to simplify the equations.

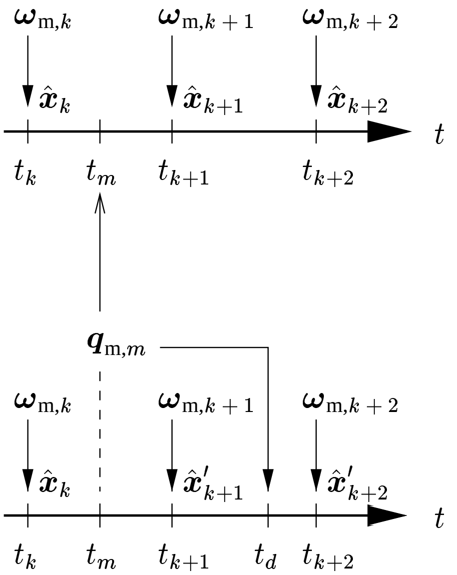

$\boldsymbol{\omega}_{\mathrm{m},k}$
$\boldsymbol{\omega}_{\mathrm{m},k+1}$
$\boldsymbol{\omega}_{\mathrm{m},k+2}$
$\hat{\boldsymbol{x}}_k$
$\hat{\boldsymbol{x}}_{k+1}$
$\hat{\boldsymbol{x}}_{k+2}$
$t$
$t_k$
$t_m$
$t_{k+1}$
$t_{k+2}$
$\boldsymbol{q}_{\mathrm{m},m}$
$\boldsymbol{\omega}_{\mathrm{m},k}$
$\boldsymbol{\omega}_{\mathrm{m},k+1}$
$\boldsymbol{\omega}_{\mathrm{m},k+2}$
$\hat{\boldsymbol{x}}_k$
$\hat{\boldsymbol{x}}'_{k+1}$
$\hat{\boldsymbol{x}}'_{k+2}$
$t$
$t_k$
$t_m$
$t_{k+1}$
$t_d$
$t_{k+2}$

**Figure 9.4**: *Illustration of the update correction algorithm time line (below), compared to an undelayed timeline (above).*

It is seen from Figure 9.4 that the measurement $\boldsymbol{q}_{\mathrm{m},m}$ is received after time $t_{k+1}$. The parallel re-calculation algorithm would restart the calculation of the estimates, starting from $\hat{\boldsymbol{x}}_{k+1}$. The estimate $\hat{\boldsymbol{x}}_{k+2}$ is the correct estimate to be used at time $t_{k+2}$. The idea of the update correction algorithm is to calculate the estimate $\hat{\boldsymbol{x}}_{k+2}$ using the measurement $\boldsymbol{q}_{\mathrm{m},m}$ at time $t_{k+2}$, and then correcting the update with $\hat{\boldsymbol{x}}_{k+2} - \hat{\boldsymbol{x}}'_{k+2}$, giving

$$\hat{\boldsymbol{x}}_{k+2} = \hat{\boldsymbol{x}}'_{k+2} + \left[\hat{\boldsymbol{x}}_{k+2} - \hat{\boldsymbol{x}}'_{k+2}\right] \tag{9.23}$$

The measurement $\boldsymbol{q}_{\mathrm{m},m}$ is propagated to time $t_{k+1}$, yielding $\boldsymbol{q}_{k+1}$ regardless of the delay time. If the measurement was incorporated in the estimate $\hat{\boldsymbol{x}}_{k+1}$, the estimate $\hat{\boldsymbol{x}}_{k+2}$ would have been given by

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k+2} =& \hat{\boldsymbol{x}}^{-}_{k+2} + \boldsymbol{K}_{k+2}\left[\boldsymbol{z}_{k+2} - \boldsymbol{H}\hat{\boldsymbol{x}}^{-}_{k+2}\right] \\
=& \left[\mathbf{1} - \boldsymbol{K}_{k+2}\boldsymbol{H}\right]\hat{\boldsymbol{x}}^{-}_{k+2} + \boldsymbol{K}_{k+2}\boldsymbol{z}_{k+2} \\
=& \left[\mathbf{1} - \boldsymbol{K}_{k+2}\boldsymbol{H}\right]\left[\boldsymbol{\Phi}\hat{\boldsymbol{x}}_{k+1} + \boldsymbol{\Gamma}\boldsymbol{u}_{k+1}\right] + \boldsymbol{K}_{k+2}\boldsymbol{z}_{k+2} \\
=& \left[\mathbf{1} - \boldsymbol{K}_{k+2}\boldsymbol{H}\right]\left[\boldsymbol{\Phi}\left[\mathbf{1} - \boldsymbol{K}_{k+1}\boldsymbol{H}\right]\hat{\boldsymbol{x}}^{-}_{k+1} + \boldsymbol{\Phi}\boldsymbol{K}_{k+1}\boldsymbol{z}'_{k+1} + \boldsymbol{\Gamma}\boldsymbol{u}_{k+1}\right] \\
& + \boldsymbol{K}_{k+2}\boldsymbol{z}_{k+2}
\end{aligned} \tag{9.24}$$

Note that $\boldsymbol{z}'_{k+1}$ is the full measurement

$$\boldsymbol{z}'_{k+1} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{m},k\,+\,1} \\ \boldsymbol{\omega}_{\mathrm{m},k\,+\,1} \end{bmatrix} \tag{9.25}$$

since attitude measurement was assumed available, and propagated to time $t_{k+1}$. The measurement $\boldsymbol{z}_{k+2}$ contains the estimated attitude, as it is explained in the multirate filter, giving

$$\boldsymbol{z}_{k+2} = \begin{bmatrix} \hat{\boldsymbol{q}}_{k+1} \\ \boldsymbol{\omega}_{\mathrm{m},k\,+\,1} \end{bmatrix} \tag{9.26}$$

which will result in the residual of Equation 9.20. Note that the state reduction and expansion in Equation 9.24 has been disregarded in order to simplify notation. This will persist throughout the section.

Equation 9.24 gives the estimate at time $t_{k+2}$ assuming the measurement $\boldsymbol{q}_{\mathrm{m},k\,+\,1}$ is used at time $t_{k+1}$. Due to the delay of the measurement it cannot be used at time $t_{k+1}$. The estimate $\hat{\boldsymbol{x}}'_{k+2}$ of time $t_{k+2}$, disregarding the measurement update at time $t_{k+1}$, and instead using the attitude estimate at both times $t_{k+1}$ and $t_{k+2}$, yields

$$\begin{aligned}
\hat{\boldsymbol{x}}'_{k+2} &= \left(\hat{\boldsymbol{x}}'_{k+2}\right)^{-} + \boldsymbol{K}'_{k+2}\left[\boldsymbol{z}_{k+2} - \boldsymbol{H}\left(\hat{\boldsymbol{x}}'_{k+2}\right)^{-}\right] \\
&= \left[\boldsymbol{1} - \boldsymbol{K}'_{k+2}\boldsymbol{H}\right]\left(\hat{\boldsymbol{x}}'_{k+2}\right)^{-} + \boldsymbol{K}'_{k+2}\boldsymbol{z}_{k+2} \\
&= \left[\boldsymbol{1} - \boldsymbol{K}'_{k+2}\boldsymbol{H}\right]\left[\boldsymbol{\Phi}\hat{\boldsymbol{x}}'_{k+1} + \boldsymbol{\Gamma}\boldsymbol{u}_{k+1}\right] + \boldsymbol{K}'_{k+2}\boldsymbol{z}_{k+2} \\
&= \left[\boldsymbol{1} - \boldsymbol{K}'_{k+2}\boldsymbol{H}\right]\left[\boldsymbol{\Phi}\left[\boldsymbol{1} - \boldsymbol{K}'_{k+1}\boldsymbol{H}\right]\left(\hat{\boldsymbol{x}}'_{k+1}\right)^{-} + \boldsymbol{\Phi}\boldsymbol{K}'_{k+1}\boldsymbol{z}_{k+1} + \boldsymbol{\Gamma}\boldsymbol{u}_{k+1}\right] \\
&\quad + \boldsymbol{K}'_{k+2}\boldsymbol{z}_{k+2}
\end{aligned} \tag{9.27}$$

The Kalman gains in Equation 9.24 and 9.27 differ, due the update of the covariance $\boldsymbol{P}$ at time $t_{k+1}$, when the attitude measurement is assumed available. However if the covariance matrix is updated at time $t_{k+1}$, as if an attitude measurement was available, the Kalman gains $\boldsymbol{K}'$ are equal to $\boldsymbol{K}$. The difference between the two equations, which is the effect of not updating with a measurement at time $t_{k+1}$, becomes

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k+2} - \hat{\boldsymbol{x}}'_{k+2} &= \left[\boldsymbol{1} - \boldsymbol{K}_{k+2}\boldsymbol{H}\right]\boldsymbol{\Phi}\boldsymbol{K}_{k+1}\left[\boldsymbol{z}'_{k+1} - \boldsymbol{z}_{k+1}\right] \\
&= \left[\boldsymbol{1} - \boldsymbol{K}_{k+2}\boldsymbol{H}\right]\boldsymbol{\Phi}\boldsymbol{K}_{k+1}\begin{bmatrix}\boldsymbol{q}_{\mathrm{m},k+1} - \hat{\boldsymbol{q}}_{k+1} \\ \boldsymbol{0}\end{bmatrix}
\end{aligned} \tag{9.28}$$

realizing that

$$\left(\hat{\boldsymbol{x}}'_{k+1}\right)^{-} = \hat{\boldsymbol{x}}^{-}_{k+1} \tag{9.29}$$

since the a priori estimates are a propagation of $\hat{\boldsymbol{x}}_k$, which is not influenced by the measurement at time $t_m$.

This update correction can be calculated if the estimate at time $t_{k+1}$, and the following Kalman gains are saved. The time of measurement $t_m$ is known, hence the filter knows when to start saving the estimate and Kalman gains.

The disadvantage of the algorithm is, that the covariance of the state estimate is wrong at time $t_{k+1}$, since the covariance must be updated as if an attitude measurement is available. When the measurement is used at time $t_{k+2}$, the covariance is true. Since the covariance of the estimate is used to ensure the pointing performance, this yields a problem. If the filter is to be used in fine mode, in case the steady-state filter is inapplicable, the covariance is used to change from fine to coarse mode, when the estimate cavariance increases above a thershold.

However, since the covariance of the estimate is independent of the actual measurement, and only of the time of measurement update, the maximum delay time allowed, can be found through simulation or pre-calculation.

If simulations show that a maximum delay time of $t_{\mathrm{md}}$ is allowed, in order to keep the covariance of the estimate below the threshold needed to stay in fine mode, change to coarse mode can be based on the observed delay time of the measurement instead of the covariance. The delay time from the measurement is executed $t_m$ to the measurement is incorporated in the filter $t_d$ must be below $t_{\mathrm{md}}$, or else the mode is changed to coarse.

The general equation of the update correction term $\tilde{\boldsymbol{x}}_{k+d}$, given an arbitrary delay time $d$, is given by

$$\tilde{\boldsymbol{x}}_{k+d} = \prod_{i=2}^{d} \left( \mathbf{1} - \boldsymbol{K}_{k+i} \boldsymbol{H} \right) \boldsymbol{\Phi}^{d-1} \boldsymbol{K}_{k+1} \begin{bmatrix} \boldsymbol{q}_{\mathrm{m},k+1} - \hat{\boldsymbol{q}}_{k+1} \\ \mathbf{0} \end{bmatrix} \tag{9.30}$$

assuming that the measurement was executed between time $t_k$ and $t_{k+1}$, and received in time for incorporation at time $t_{k+d}$.

## On Demand Estimation

The on demand estimation algorithm, simplifies the problem of sensing delay from a design point of view. As a consequence the implementation of the algorithm requires more design analysis.

The algorithm saves all measurement available between two controller sample points, and then calculates the estimate in time before the next sample point of the controller. Figure 9.5 illustrates the time line of events.

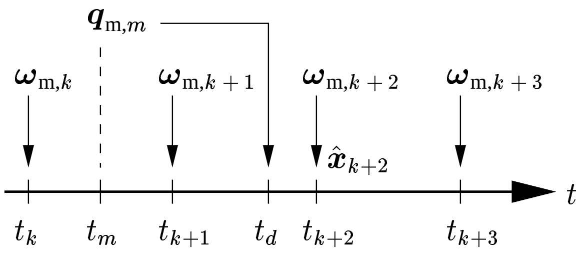

$\boldsymbol{q}_{\mathrm{m},m}$
$\boldsymbol{\omega}_{\mathrm{m},k}$
$\boldsymbol{\omega}_{\mathrm{m},k+1}$
$\boldsymbol{\omega}_{\mathrm{m},k+2}$
$\boldsymbol{\omega}_{\mathrm{m},k+3}$
$\hat{\boldsymbol{x}}_{k+2}$
$t$
$t_k$
$t_m$
$t_{k+1}$
$t_d$
$t_{k+2}$
$t_{k+3}$

**Figure 9.5**: *Illustration of the on demand estimation algorithm time line, with a short measurement delay.*

The measurements received by the filter at times $t_k$, $t_{k+1}$, and the delayed measurement at time $t_d$, are saved in the filter. Note that no estimates are calculated at times $t_k$ and $t_{k+1}$. If the controller requires an estimate at time $t_{k+2}$, the estimator initiates the calculation of the estimate just before time $t_{k+2}$, using the measurements received. In this way, the measurement $\boldsymbol{q}_{\mathrm{m},m}$ can be incorporated as if no delay occurred. The disadvantage of the algorithm is, that the implementation requires the execution time to be known, in order to initiate calculation in time for completion at time $t_{k+2}$.

The example given in Figure 9.5, illustrates a delay, which does not influence the estimate, because the measurement can be incorporated at the time the controller requires an estimate. If the delay time increases, this cannot be done. Consider the extended delay time in Figure 9.6. The measurement is delayed beyond the time $t_{k+2}$, where the controller requires an estimate. Hence the measurement cannot be used in the estimate of $\hat{\boldsymbol{x}}_{k+2}$.

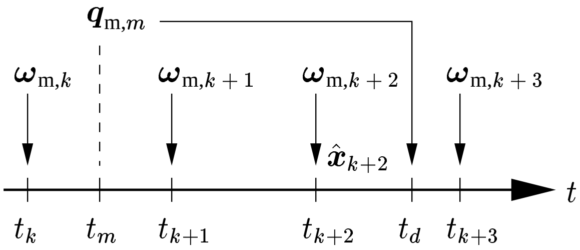

$\boldsymbol{q}_{\mathrm{m},m}$
$\boldsymbol{\omega}_{\mathrm{m},k}$
$\boldsymbol{\omega}_{\mathrm{m},k+1}$
$\boldsymbol{\omega}_{\mathrm{m},k+2}$
$\boldsymbol{\omega}_{\mathrm{m},k+3}$
$\hat{\boldsymbol{x}}_{k+2}$
$t$
$t_k$
$t_m$
$t_{k+1}$
$t_{k+2}$
$t_d$
$t_{k+3}$

**Figure 9.6**: *Illustration of the on demand estimation algorithm time line, with an extended measurement delay.*

This problem can be solved by using measurement propagation, as discussed in the above. The measurement at time $t_m$ is propagated to time $t_{k+3}$, where it will be incorporated when calculating the next estimate for the controller.

## Comparison of Algorithms

Four algorithms were discussed in the previous sections. This section summarizes the advantages and disadvantages of the algorithms, and based on design criteria, the preferred solution is derived.

The design criteria of the estimator, is a trade of between accuracy and computational load. If accuracy requirements are met, a less complex filter can be considered in order to free computation time for other tasks on the on-board computer. Table 9.1 shows the algorithms evaluated wrt. these criteria. In addition the implementation possibilities are stated. Multi-rate / single-rate implies that the algorithm could be implemented in a single rate filter, possibly using a low pass filter on the rate estimate, as discussed in the multi-rate filter section.

Obviously the algorithm using most computation time is the parallel re-calculation algorithm. In return the estimate suffers no loss in estimate accuracy due to the measurement delay. The measurement propagation algorithm uses a minimum computation time. The on demand algorithm uses the same amount, but in large chunks at a lower frequency, since it starts calculation just before an estimate is needed. If the delayed measurement arrives before the estimate calculation, the on demand algorithm estimate has no accuracy loss. If the measurement is delayed beyond the point of calculation, the measurement is propagated, suffering from the same loss of accuracy as the measurement

| **Algorithm** | **Implementation** | **Computation** | **Accuracy** |
|---|---|---|---|
| Measurement Propagation | Multi-rate / Single-rate | Low | Low |
| Parallel Re-calculation | Multi-rate | High | High |
| Update Correction | Multi-rate | Medium | High |
| On Demand | Multi-rate / Single-rate | Low | Medium |

**Table 9.1**: *Filter algorithms evaluated with respect to design criteria.*

propagation algorithm. Using measurement propagation clearly decreases the accuracy of the estimate, since the measurement is influenced by the error of the rate estimate, which is used for the propagation. Based on this, the on demand algorithm is preferred to the measurement propagation, since the on demand algorithm only propagates when it is necessary.

A compromise between the parallel re-calculation and the on demand algorithm, is the update correction, which corrects for the delayed update, at the cost of an extra computational load. This load, however, is significantly smaller than the parallel re-calculation, since no propagation is needed. The estimate is not influenced by the delayed measurement, instead the covariance is at some points out of synchronization. This happens between the time the measurement is carried out, and the time it is received.

The update correction is preferred to the parallel re-calculation, since the update correction algorithm requires less computation, at no cost of the estimate accuracy. This leaves the update correction and the on demand algorithm. The choice of algorithm should be based on the performance of the estimator. If the accuracy of the estimate is far better than the requirements, the on demand algorithm should be evaluated, possibly with a low pass filter on the rate, instead of using a multi-rate filter. If requirements are not met, the update correction algorithm is preferred.

# Part IV

# Simulation

Chapter 10

# Implementation

This chapter gives an overview of the SIMULINK implementation of the multi-rate and steady state estimators.

## 10.1 Multi-rate Kalman Filter

The main system of the multi-rate estimator block is shown in Figure 10.1.

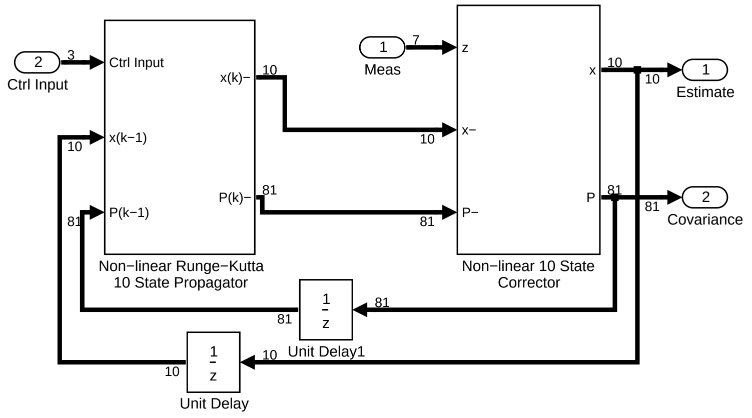


**Figure 10.1**: *Multi-rate Kalman filter subsystem block.*

The algebraic loop of the system, is broken in the unit delay blocks. The initial values of the estimate $\hat{\boldsymbol{x}}_0$ and covariance $\boldsymbol{P}_0$ are output of the delay blocks when the simulation starts.

The propagator uses the the current estimate and the control input to propagate the estimate $\hat{\boldsymbol{x}}_k$ and covariance $\boldsymbol{P}_k$ to the a priori values $\hat{\boldsymbol{x}}_{k+1}^-$ and $\boldsymbol{P}_{k+1}^-$. The propagator block is shown in Figure 10.2.

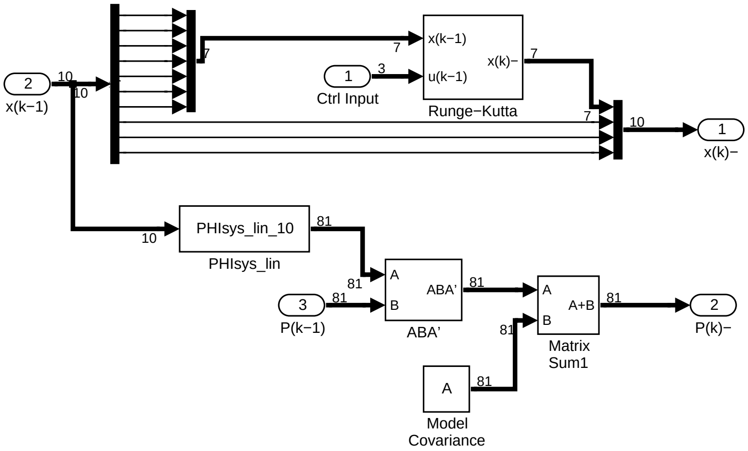


**Figure 10.2**: *Multi-rate non-linear Runge-Kutta 10 state propagator subsystem block.*

The integrator block of SIMULINK is not applicable in the extended Kalman filter, due to its behavior upon reset, hence a Runge-Kutta propagator has been implemented. The non-linear differential function $\boldsymbol{f}\left(\hat{\boldsymbol{x}}_k\right)$ is linearized and made discrete in the *PHIsys_lin* block, which gives the transition matrix $\boldsymbol{\Phi}\left(\hat{\boldsymbol{x}}_k\right)$. The transition matrix is used to propagate the covariance. The bias is simply passed through the propagator block, since it has zero dynamics.

The a priori values are used in the corrector block, together with the measurement vector $\boldsymbol{z}_k$, which contains the noise simulated by the star tracker and gyro models. The corrector block is shown in Figure 10.3.

The *select* block, seen in Figure 10.3, is used in the multi-rate corrector to change the measurement of $\boldsymbol{q}$ from the estimate $\hat{\boldsymbol{q}}$ and the actual measurement. Recall that when the attitude measurement is unavailable, the estimate is used to produce a measurement residual of zero. The *select1* block outputs the $\boldsymbol{H}_{\mathrm{g}}$ when attitude measurement is unavailable, and $\boldsymbol{H}$ when available.

Note also that the state is reduced in the corrector by the *large2small 10 state* block. After the correction, the state is expanded in the *small2large 10 state* block.

## 10.2 Steady-state Kalman Filter

The steady-state Kalman filter is shown in Figure 10.4. The layout is the same as the multi-rate filter, except that the covariance is left out of the loop, since it is constant.

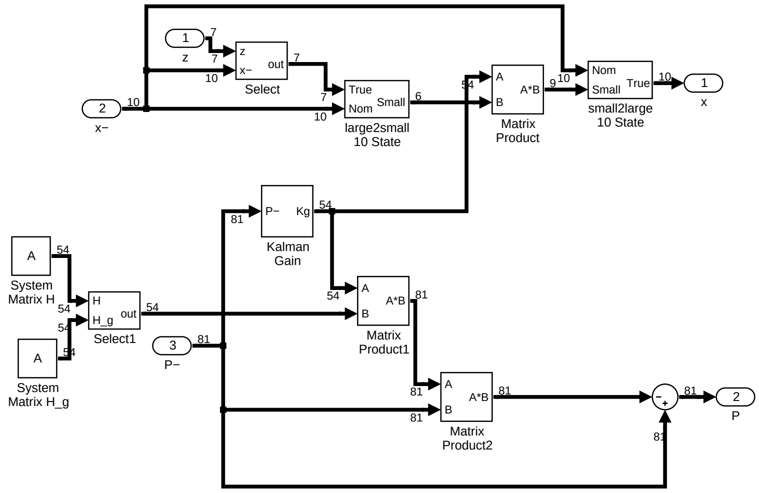


**Figure 10.3**: *Multi-rate quaternion 10 state corrector subsystem block.*

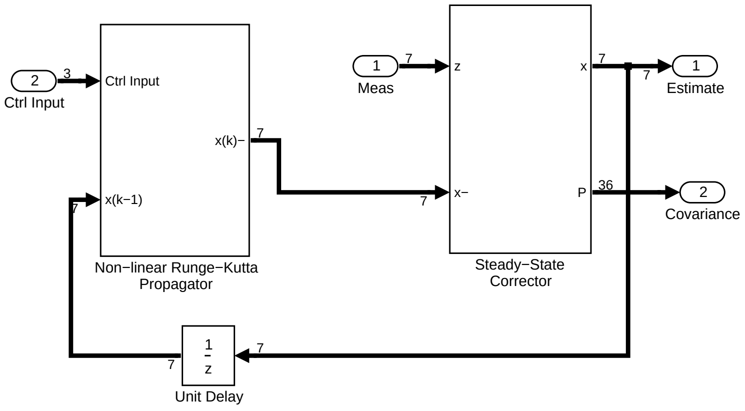


**Figure 10.4**: *Steady-state Kalman filter subsystem block.*

The propagator is shown in Figure 10.5.

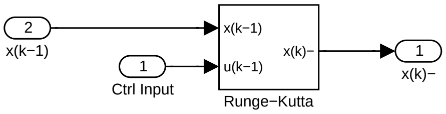


**Figure 10.5**: *Steady-state non-linear propagator subsystem block.*

Using $\mathbf{\Phi}\left(\bar{\boldsymbol{x}}\right)$ for the propagation of $\hat{\boldsymbol{x}}$ proved inapplicable. Instead the Runge-Kutta propagator block from the multi-rate filter is used. This requires more computation, but an advantage is, that the filter is not limited to states close to the nominal value $\bar{\boldsymbol{x}}$.

Note that the propagation of the covariance is not necessary, since the steady-state value is calculated in the corrector block. The corrector block is shown in Figure 10.6.

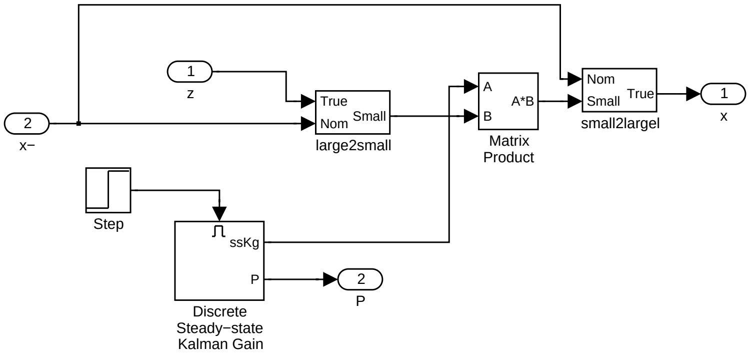


**Figure 10.6**: *Steady-state corrector subsystem block.*

The *Discrete Steady-state Kalman Gain* block calculates a steady-state Kalman gain, based on the steady-state covariance, which is a solution to the discrete algebraic Riccati equation in Equation 8.17. The block is only enabled in the first sample of the simulation, hence the computation of the Kalman gain is executed once, and used throughout the simulation.

Note that the measurement vector $\boldsymbol{z}$, which is input to the block, is subtracted by the estimated biases from the multi-rate filter. The steady-state filter does not estimate the bias, hence the multi-rate filter has to do this, before the steady-state filter is used. This requires that fine mode is only entered from coarse mode. Recall Figure 3.2, where the pre-defined modes and transitions of Rømer are illustrated. It is seen that fine mode can be entered from standby mode. Either bias estimation must be done in standby mode, or transition three should be removed, such that fine mode is only entered from coarse mode. In this way, the bias estimates will always be available for the steady-state filter in fine mode.

## 10.3 Model and Measurement Noise Strength Matrices

The Kalman filters rely on the knowledge of the model noise strength $\boldsymbol{Q}$ and measurement noise strength $\boldsymbol{R}$. The performance of the Kalman filters are only optimal under the assumption that $\boldsymbol{Q}$ and $\boldsymbol{R}$ reflect the true behavior of the noises.

The measurement noise strength is found by simulating the noise models of the star trackers and gyros. The output noise strengths are inserted as the diagonal elements of $\boldsymbol{R}$, resulting in

$$\boldsymbol{R} = \begin{bmatrix} 1.2 \times 10^{-5} \times \mathbf{1} & \mathbf{0} \\ \mathbf{0} & 0.02 \times \mathbf{1} \end{bmatrix} \tag{10.1}$$

where all matrices are $3$ by $3$.

The Kalman filters assume that the measurement noise is zero-mean white noise. This is not the case of the gyro noise, since the measurement is biased. Instead the strength of the gyro measurement noise is set to a value which includes the bias value.

The model noise strength is more difficult to estimate, since the effect of disturbances has not been analyzed. The kinematic equation of the attitude is not influenced by error, hence the modeling noise of the quaternion is set to zero. The bias is modeled as zero dynamics. The noise model changes the measurement noise with the drift rate, which has a deviation of $0.0002\mathrm{rad/s}^2$. The model noise strength of the rate is assumed to be in the order of $10^{-6}$, and the error is modeled as such, by adding white noise to the true state of the system. These considerations result in a modeling noise strength of

$$\boldsymbol{Q} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 10^{-6} \times \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2 \times 10^{-5} \times \mathbf{1} \end{bmatrix} \tag{10.2}$$

where all matrices are $3$ by $3$.

Adding white noise to the true state of the system, and using the observed noise strengths from the noise models in $\boldsymbol{Q}$ and $\boldsymbol{R}$ improves the simulated performance of the estimators, which should be considered when evaluating the results.

## 10.4 Sensing Delay

The update correction algorithm has not been implemented in this project. The performance of the estimator may still be evaluated, since the update algorithm ensures that the estimate is unaffected, once the measurement is received and incorporated in the estimate.

The effect of the delay time is analyzed by changing the interval of which the attitude update is done in the estimators.

The performance of the estimators is presented and discussed in the following chapter.

# Chapter 11

# Results

In this chapter the results of the SIMULINK simulations are presented and discussed. First the implementations are verified by showing their ability to converge from an initial estimate, which is different from the true state of the system. The multi-rate and steady-state filters have been tested in a static and dynamic environment, in order to evaluate their performance. The results are evaluated wrt. the system requirements in Chapter 4.

All simulations have been conducted in open loop, i.e. no controller is implemented. The behavior of the satellite is done by setting an initial attitude and angular rate of the satellite model. The model in the Kalman filters likewise need initial values. The initial conditions for the true state simulation are denoted $\boldsymbol{q}_0$ and $\boldsymbol{\omega}_0$, and $\hat{\boldsymbol{q}}_0$ and $\hat{\boldsymbol{\omega}}_0$ for the filters. The equivalent state vectors are $\boldsymbol{x}_0$ and $\hat{\boldsymbol{x}}_0$, respectively, where the bias estimate is denoted $\hat{\boldsymbol{\beta}}_0$.

## 11.1 Verification

The purpose of the verification is to see if the estimation error converges from an initial value, which is non-zero. The filters have been simulated in equal environments, where the satellite is simulated as static. The initial estimates are chosen to be significantly different from the true state. The simulation parameters are stated in Table 11.1.

The result of the multi-rate Kalman filter is shown in Figure 11.1, and the result of the steady-state Kalman filter in Figure 11.2. The estimation errors of $\boldsymbol{q}$ and $\boldsymbol{\omega}$ are denoted $\hat{\boldsymbol{q}}_\epsilon$ and $\hat{\boldsymbol{\omega}}_\epsilon$, respectively.

The results of the verification simulations show, that both filters are able to estimate the state of the system, from initial values which are different from the true state. Both filters have converged after approximately 100s. The multi-rate filter is faster to correct

| Symbol | Value |
|---|---|
| $\boldsymbol{q}_0$ [rad] | $[0\ 0\ 0\ 1]^{\mathrm{T}}$ |
| $\boldsymbol{\omega}_0$ [°/s] | $[0\ 0\ 0]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{q}}_0$ [rad] | $[0.4082\ 0.4082\ 0.4082\ 0.7071]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{\omega}}_0$ [°/s] | $[0.5\ 0.5\ 0.5]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{\beta}}_0$ [°/s] | $[0\ 0\ 0]^{\mathrm{T}}$ |

**Table 11.1**: *Simulation parameters of the verification of multi-rate and steady-state filters.*

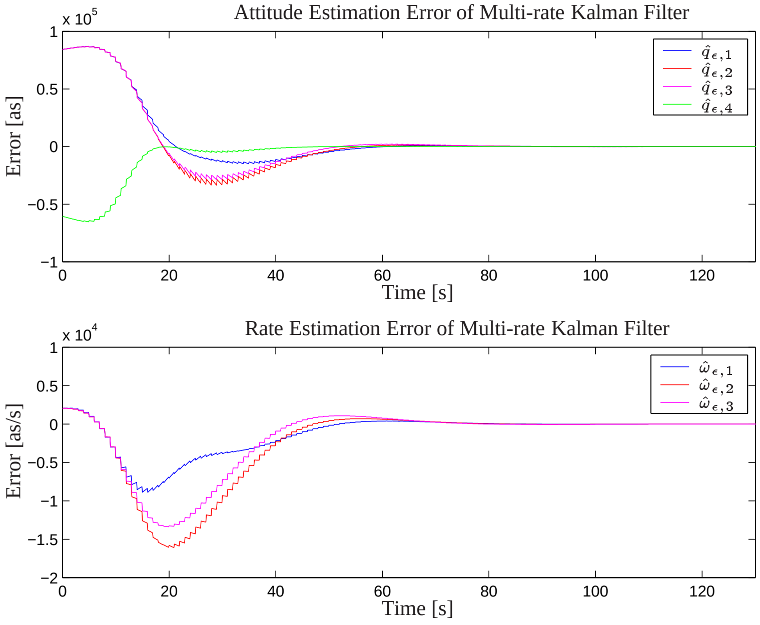


**Figure 11.1**: *Verification result of the multi-rate Kalman filter.*

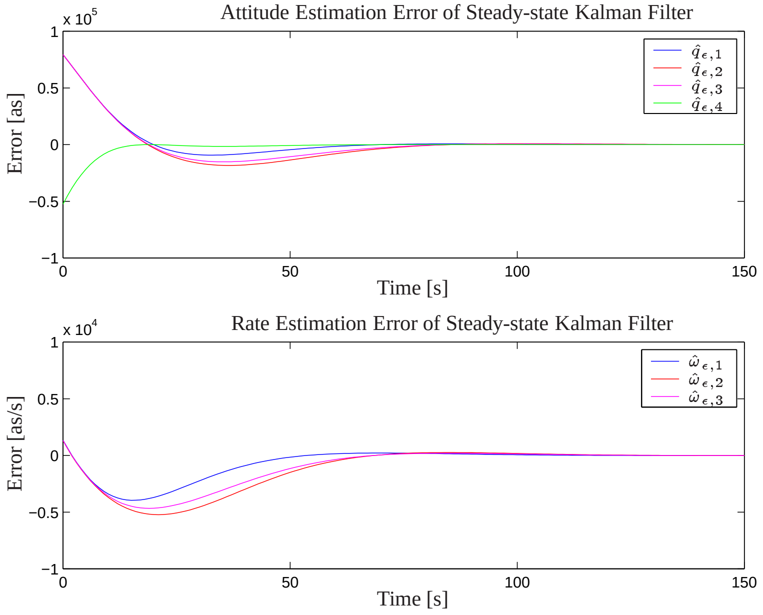


**Figure 11.2**: *Verification result of the steady-state Kalman filter.*

the estimate than the steady-state filter. This is an advantage of estimating the covariance on-line.

## 11.2 Multi-rate Kalman Filter

The performance of the filters in static and dynamic environments are investigated. The results are presented in the following.

### Static Simulation

The parameters of the static simulation are listed in Table 11.2.

The result of the static simulation is shown in Figure 11.3. The result is plotted in terms of the Euler angles, in order to compare them with the system requirements.

The mean and RMS values of the Euler angle errors are listed in Table 11.3. The values are calculated from time $100$ and out, to ensure that the filter is in steady-state. This is consistent for all the following calculations of mean, RMS, and RSS values.

The mean and RSS values for the attitude and rate estimation errors are listed in Table 11.4, together with the equivalent values of the measurement vector.

It is seen from the results, that the attitude estimation RSS error is improved from $2.4$as

| Symbol | Value |
|---|---|
| $\boldsymbol{q}_0$ [rad] | $[0\ 0\ 0\ 1]^{\mathrm{T}}$ |
| $\boldsymbol{\omega}_0$ [°/s] | $[0\ 0\ 0]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{q}}_0$ [rad] | $[0\ 0\ 0\ 0]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{\omega}}_0$ [°/s] | $[0\ 0\ 0]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{\beta}}_0$ [°/s] | $[0\ 0\ 0]^{\mathrm{T}}$ |

**Table 11.2**: *Parameters of the static simulation for the multi-rate Kalman filter.*

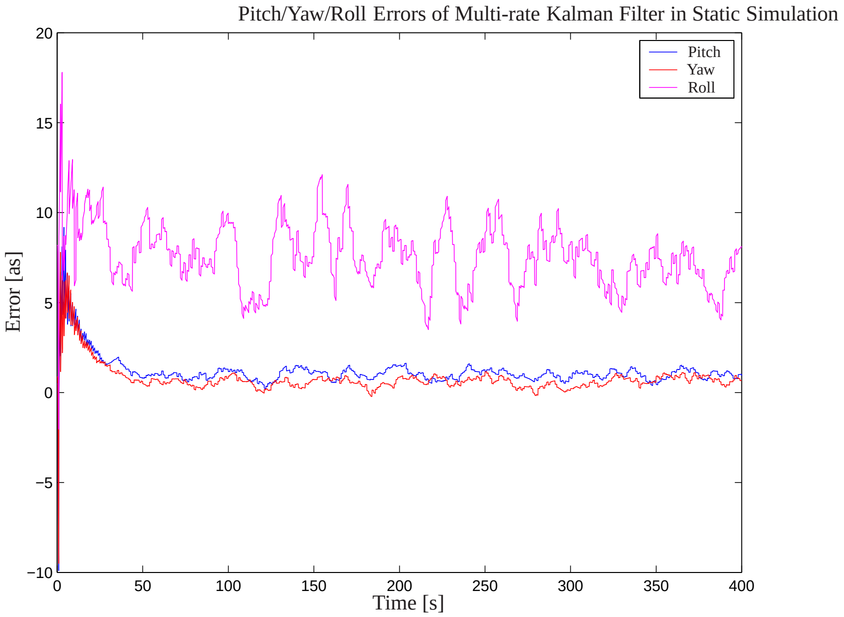


**Figure 11.3**: *Results of the static simulation of the multi-rate Kalman filter.*

| **Euler Angle** | **Mean** [as] | **RMS** [as] |
|---|---|---|
| Pitch | 1.0 | 0.27 |
| Yaw | 0.59 | 0.27 |
| Roll | 7.3 | 1.7 |

**Table 11.3**: *Mean and RMS values of the Euler angle error from static simulation with multi-rate Kalman filter.*

| **Vector** | **Mean** | **RSS** |
|---|---|---|
| | Estimates | |
| $\hat{\boldsymbol{q}}$ | $-1.1$as | $0.85$as |
| $\hat{\boldsymbol{\omega}}$ | $0.003$as/s | $0.13$as/s |
| | Measurements | |
| $\boldsymbol{q}_{\mathrm{m}}$ | $-1.1$as | $2.4$as |
| $\boldsymbol{\omega}_{\mathrm{m}}$ | $1800$as/s | $29$as/s |

**Table 11.4**: *Mean and RSS values of the estimate and measurement errors from static simulation with multi-rate Kalman filter.*

to $0.85$as. The bias in the attitude estimate is caused by the bias of the measurement. It should be considered to estimate this bias, if improved performance is required. This has not been included in this project.

The rate RSS error is improved from $29$as/s to $0.13$as/s. The bias in the measurement is estimated. As a result the mean value of the estimate is almost zero, compared to the measurement mean.

The covariance estimate is also investigated. Figure 11.4, shows a plot of the error in $\hat{q}_1$ and $\hat{\omega}_1$ with a 2-sigma confidence contour. The sigma value of $\hat{q}_{\epsilon,1}$ is calculated from the covariance $\boldsymbol{P}$, by

$$\sigma_{\hat{q}_\epsilon,1} = \sqrt{P_{1,1}} \tag{11.1}$$

where $P_{1,1}$ is the element at the 1st row and 1st column of $\boldsymbol{P}$.

It is seen that in spite the $0.85$as RSS value of $\hat{\boldsymbol{q}}_\epsilon$, the estimated 2-sigma value is at approximately $600$as. This implies an error in the estimation of the state covariance, which has not been solved by the deadline of this project. The error is also found in the steady-state filter. The covariance of the steady-state filter is however smaller, which results in a better performance of the steady-state filter. The problem could be caused by the gyro bias. In order to include this noise source in the filter, the measurement noise strength of the gyro is increased to include the bias. This actually represents a zero mean Gaussian noise with a large deviation, which does not reflect the behavior of the gyro measurement noise.

In the steady-state filter, the bias is subtracted before the measurement is used in the filter, hence the measurement noise strength is set to the deviation of the noise, excluding bias.

The covariance analysis is left out in the remaining simulations, due to this problem.

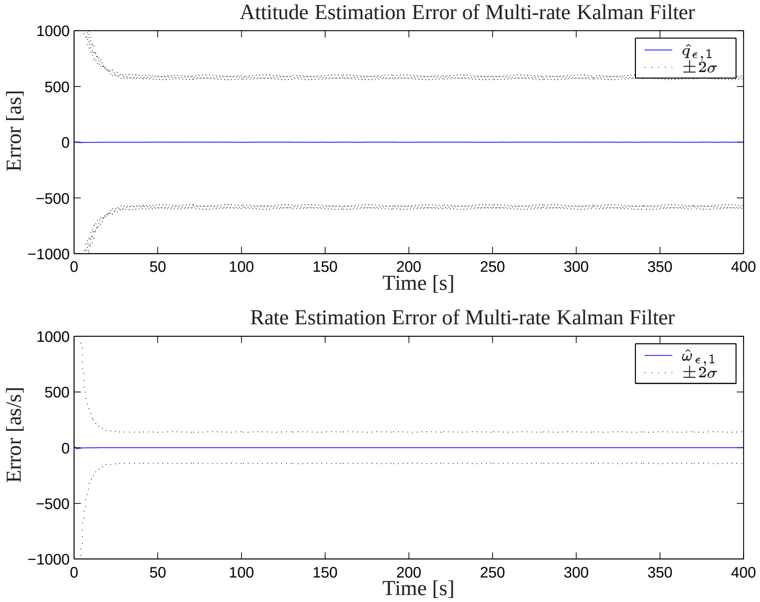


**Figure 11.4**: *Estimation error of $\hat{q}_1$ and $\hat{\omega}_1$ with 2-sigma confidence contour.*

## Dynamic Simulation

In order to investigate the performance of the estimator in a worst case situation wrt. measurement noise, a dynamic simulation is performed, where the satellite is rotated at constant angular velocity. The parameters of the dynamic simulation are listed in Table 11.5.

| **Symbol** | **Value** |
|---|---|
| $\boldsymbol{q}_0$ [rad] | $[0\ 0\ 0\ 1]^{\mathrm{T}}$ |
| $\boldsymbol{\omega}_0$ $[^\circ/\mathrm{s}]$ | $[5\ 5\ 5]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{q}}_0$ [rad] | $[0\ 0\ 0\ 0]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{\omega}}_0$ $[^\circ/\mathrm{s}]$ | $[5\ 5\ 5]^{\mathrm{T}}$ |
| $\hat{\boldsymbol{\beta}}_0$ $[^\circ/\mathrm{s}]$ | $[0\ 0\ 0]^{\mathrm{T}}$ |

**Table 11.5**: *Parameters of the dynamic simulation for the multi-rate Kalman filter.*

The results of the simulation are shown in Figure 11.5, and the RMS values of the Euler angle errors are listed in Table 11.6.

The mean and RSS values of the estimates and measurements are listed in Table 11.7.

The accuracy of the estimate is decreased, which is expected. This is due to the increased

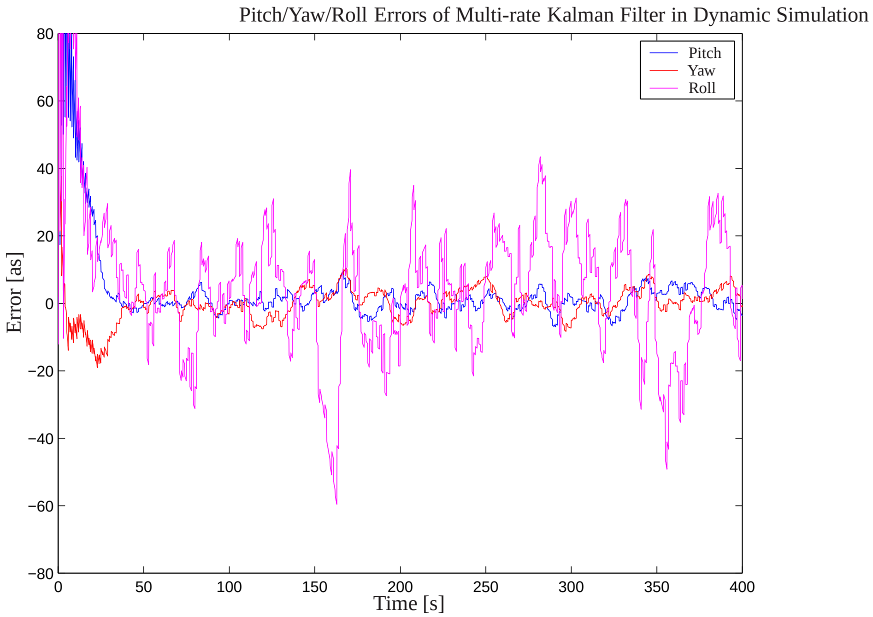


**Figure 11.5**: *Results of the dynamic simulation of the multi-rate Kalman filter.*

| **Euler Angle** | **Mean** [as] | **RMS** [as] |
|---|---|---|
| Pitch | 0.89 | 3.0 |
| Yaw | 0.81 | 3.8 |
| Roll | 2.4 | 18 |

**Table 11.6**: *Mean and RMS values of the Euler angle error from dynamic simulation with multi-rate Kalman filter.*

| **Vector** | **Mean** | **RSS** |
|---|---|---|
| Estimates | | |
| $\hat{\boldsymbol{q}}$ | $-0.56\text{as}$ | $8.0\text{as}$ |
| $\hat{\boldsymbol{\omega}}$ | $0.29\text{as}/\text{s}$ | $2.28\text{as}/\text{s}$ |
| Measurements | | |
| $\boldsymbol{q}_{\text{m}}$ | $-0.25\text{as}$ | $21\text{as}$ |
| $\boldsymbol{\omega}_{\text{m}}$ | $1976\text{as}/\text{s}$ | $755\text{as}/\text{s}$ |

**Table 11.7**: *Mean and RSS values of the estimate and measurement errors from dynamic simulation with multi-rate Kalman filter.*

measurement noise. The RSS error of the attitude is improved from $21$as to $8$as, and the rate RSS from $755$as/s to $0.29$as/s.

## 11.3 Steady-state Kalman Filter

The static and dynamic simulations are repeated with the steady-state Kalman filter, and the results are presented in the following. The measurement of $\boldsymbol{\omega}$ is subtracted by the estimated bias $\hat{\boldsymbol{\beta}}$ of the multi-rate Kalman filter, which is

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} 0.00851 \\ 0.00850 \\ 0.00850 \end{bmatrix} \text{as/s} \tag{11.2}$$

The plots of the simulation results are left out, as the behavior of the errors reflect the previous cases.

### Static Simulation

The initial parameters of the static simulation are the same as in the multi-rate simulation, and listed in Table 11.2. The mean and RMS values of the Euler angle errors are listed in Table 11.8. The mean and RSS values of the estimates and measurements are listed in Table 11.9.

| **Euler Angle** | **Mean** [as] | **RMS** [as] |
|---|---|---|
| Pitch | $0.14$ | $0.17$ |
| Yaw | $-0.34$ | $0.28$ |
| Roll | $6.6$ | $1.1$ |

**Table 11.8**: *Mean and RMS values of the Euler angle error from static simulation with steady-state Kalman filter.*

The RSS of the attitude error is improved from $2.4$as to $0.58$as, and the RSS of the rate error from $85$as/s to $2.4$as/s.

### Dynamic Simulation

The initial parameters of the dynamic simulation are given in Table 11.5. The results of the simulation are listed in Tables 11.10 and 11.11.

| **Vector** | **Mean** | **RSS** |
|---|---:|---:|
| Estimates | | |
| $\hat{\boldsymbol{q}}$ | $-0.80\text{as}$ | $0.58\text{as}$ |
| $\hat{\boldsymbol{\omega}}$ | $0.04\text{as/s}$ | $0.054\text{as/s}$ |
| Measurements | | |
| $\boldsymbol{q}$ | $-1.1\text{as}$ | $2.4\text{as}$ |
| $\boldsymbol{\omega}$ | $1802\text{as/s}$ | $85\text{as/s}$ |

**Table 11.9**: *Mean and RSS values of the estimate and measurement errors from static simulation with steady-state Kalman filter.*

| **Euler Angle** | **Mean** [as] | **RMS** [as] |
|---|---:|---:|
| Pitch | $293$ | $759$ |
| Yaw | $-287$ | $628$ |
| Roll | $-276$ | $1561$ |

**Table 11.10**: *Mean and RMS values of the Euler angle error from dynamic simulation with steady-state Kalman filter.*

| **Vector** | **Mean** | **RSS** |
|---|---:|---:|
| Estimates | | |
| $\hat{\boldsymbol{q}}$ | $17\text{as}$ | $818\text{as}$ |
| $\hat{\boldsymbol{\omega}}$ | $-43\text{as/s}$ | $-0.39\text{as/s}$ |
| Measurements | | |
| $\boldsymbol{q}$ | $-43\text{as}$ | $19\text{as}$ |
| $\boldsymbol{\omega}$ | $1965\text{as/s}$ | $313\text{as/s}$ |

**Table 11.11**: *Mean and RSS values of the estimate and measurement errors from dynamic simulation with steady-state Kalman filter.*

It is seen that the steady-state filter in a dynamic environment, is unable to estimate the state of the system, with an acceptable accuracy. The accuracy of the raw measurement data is better than the estimate. This is of no consequence, as the filter is only to be used in fine mode, where the satellite is in a near static environment.

## 11.4 Relative Pointing

The relative pointing error requirements are shown in Figure 4.1. In order to verify the steady-state estimator, the PSD of the pitch and yaw errors have been plotted, shown in Figure 11.6.

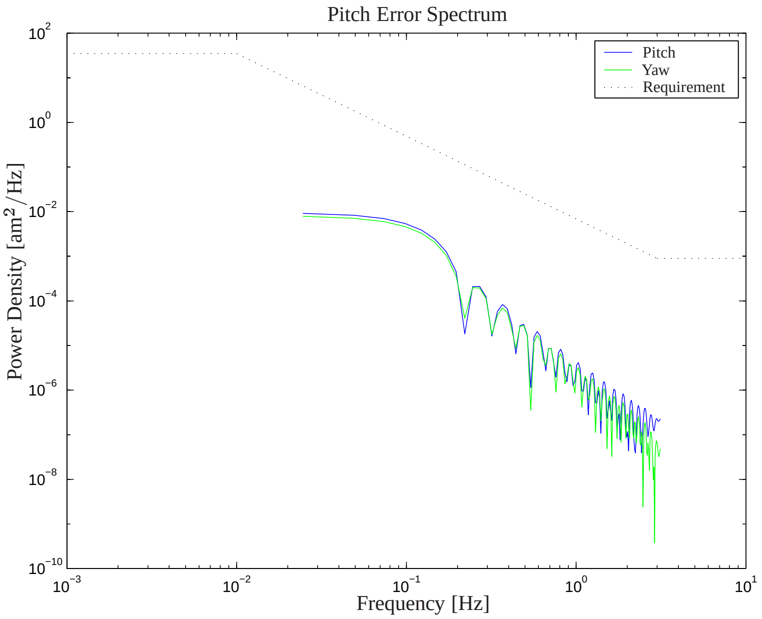


**Figure 11.6**: *PSD of the pitch and yaw errors.*

It is seen that the relative pointing error of the filter, is well within the requirements.

## 11.5 Sensing Delay

The effect of delaying the measurement of the attitude, is investigated, by sampling the steady-state Kalman filter at different rates. The steady-state filter only estimates the attitude and rate, when the attitude measurement is available.

The previous simulations used an attitude measurement every second. The time between each attitude measurement, called the measurement period, is set to $2$s, $3$s, $5$s, and

10s. Table 11.12 shows the results of the pitch/yaw and roll errors for different . The simulation was executed in a static environment.

| **Measurement Period** [s] | **Pitch/Yaw RSS** [as] | **Roll RMS** [as] |
|---|---|---|
| 2 | 1.4 | 3.2 |
| 3 | 3.0 | 3.5 |
| 5 | 5.1 | 4.3 |
| 10 | 8.6 | 7.9 |

**Table 11.12**: *Pitch/yaw error RSS and roll error RMS from the varying measurement period simulations.*

The pitch of the relative pointing is investigated. Figure 11.7 shows the results from the simulations.

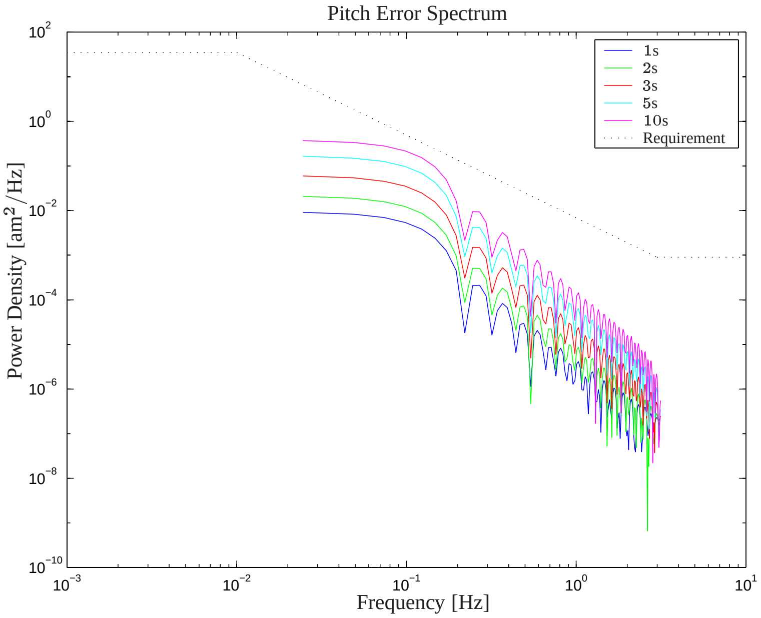


**Figure 11.7**: *PSD of the pitch error in variable measurement period simulations.*

It is seen from Figure 11.7 that the relative pointing requirements are met with smaller margin, when the time between updates increases. With a measurement period of 10s the relative pointing requirement is met with a small margin. As mentioned earlier, the model and measurement noise matrices reflect the true noises of the simulation. Since this will not be the case in the real-life implementation, the results simulated results are better, than is expected in the real-life. Hence some margin to the requirements is wanted.

The star tracker measurements consume CPU time. The measurement period should be adjusted in the controller design, in order to free CPU time, while still meeting the ACS requirements. It is suggested that the measurement period should be no higher than approximately $5$s.

The simulation results of the two filters in static and dynamic environments are summarized and discussed in the following section.

## 11.6 Discussion of Results

The results of the simulations must be compared with the system requirements in Chapter 4. The attitude results of the multi-rate Kalman filter in dynamic environment, and the steady-state Kalman filter in static environment are listed in Table 11.6. The RSS values are specified by a pitch/yaw RSS value and a roll RMS value.

| **Filter** | **Multi-rate** | | **Steady-state** | |
|---|---|---|---|---|
| | Pitch/Yaw | Roll | Pitch/Yaw | Roll |
| **Mean** [as] | $0.85$ | $2.4$ | $-0.10$ | $6.6$ |
| **RMS/RSS** [as] | $4.8$ | $18$ | $0.33$ | $1.1$ |
| **System Requirements** | $15°$ | $15°$ | $2$am | $60$am |

**Table 11.13**: *Mean and RMS/RSS values of the Euler angle estimation error from the dynamic multi-rate filter simulation, and the static steady-state filter simulation.*

It is seen that the that the estimated errors are well within the requirements of the system. The multi-rate filter has $4.8$as in pitch/yaw and $18$as in roll, where a coarse pointing of $15°$ is required. The steady-state filter has $0.33$as in pitch/yaw and $1.1$as. where the requirements state $2$am for pitch/yaw and $60$am for roll. The mean values should be added to the RMS/RSS values, however these are negligible compared to the requirements.

The delay in measurement cause the accuracy of the estimate to decrease. Due to the relative pointing requirements, the measurement period of the attitude should be no higher than approximately $5$s.

Figure 11.6 shows that the steady-state filter fulfills the requirements of relative pointing. Hence it can be concluded that the estimators are applicable for the Rømer ACS.

The requirements in coarse mode of $15°$ implies that the use of a multi-rate Kalman filter is unnecessary. However the multi-rate estimator is necessary in order to estimate the gyro bias, and to ensure that the initial estimate for the steady-state filter is close to the true state of the system.

# Part V

# Conclusion and Bibliography

Chapter 12

# Conclusion

It was the main goal of this project to design an attitude estimator for the Rømer satellite. The estimator is to be incorporated in the ACS system designed at Aalborg University in cooperation with Danish Technical University. The estimator should be implemented in SIMULINK , in order to be used with the ACS simulation model.

In order to test the simulation, noise models of the sensors were developed. A detailed star tracker noise model was derived and implemented in SIMULINK . The kinematic and dynamic equations of satellite motion were derived, and used to simulate the rotation of Rømer. From the modeled satellite rotation, the gyro and star tracker noise models, calculated measurement noise contributes, which were input to the estimators.

The noise models of the hardware were tested, and produced reasonable results. The star tracker noise model, produces an attitude measurement which is better than the night sky test performed by Terma. The number of stars in the night sky test was not observed, hence the results are not fully comparable. Further real-life tests should be conducted to fit the star tracker noise model, and to verify the simulations results.

The results of the gyro noise model, are in agreement with the hardware specifications. No real-life test results were available for comparison. These test should be conducted, in order to verify the noise model.

The purpose of the estimator is to filter noise of the measurements. It was chosen to focus on Kalman filters, as they include noise information to produce optimal estimation results. Preliminary considerations suggested that a multi-rate Kalman filter should be used in coarse mode, when computation time was not limited by science calculations. A steady-state filter was proposed for the fine mode, in order to minimize computation. This would free the on-board computer for parallel science observations.

An algorithm for handling attitude measurement delays were derived. A number of algorithms were presented, and the update correction algorithm was suggested. The algorithm was not implemented in the time frame of this project. The performance of the estimators could still be investigated with liability, since the update correction algorithm cancels the effect of delays, once the measurement is received. Instead the effect of measurement delays were investigated.

The simulation results proved that the suggested filters were well within the requirements of the ACS system. The coarse mode estimator produced result which were far better than the results, implying that a simpler algorithm could be applied. However, the results of the fine mode estimator relies on the accurate estimation of the gyro biases, which are achieved by the multi-rate estimator in coarse mode.

The multi-rate filter was tested in a dynamic environment, where the satellite was tumbling with $5^\circ/$s about all three axes, and the RSS value of the pitch and yaw errors were $4.8$as and $18$as for roll.

The results of the covariance estimates were very high compared to the variance of the estimate error, implying an implementation error in the covariance estimate, which was not solved within the time frame of this project. It was suggested that the cause was due to the non-zero mean of the gyro measurement noise, which is inconsistent with the assumptions of the filter design.

The steady-state Kalman filter was tested in a static environment, where the rate was zero, and the RSS of the pitch and yaw errors were $0.33$as and $1.1$as for the roll. The power spectrum was plotted to verify the relative pointing requirement. The result was within the ACS requirement. In addition it was shown that the measurement period of the attitude should be kept below $5$s, in order to fulfill the relative pointing requirements.

It is the conclusion of this project that the preliminary goals of the project have been met. The proposed estimators have proven to be well within the requirements of the ACS controller. The SIMULINK implementations of sensor models and Kalman filters may contribute to the ACS simulation, used for the verification of the Rømer ACS.

# Bibliography

(2001). *Roemer Pre-Review Meeting*, Aalborg University.

Alexander, H. L. (1991). State estimation for distributed systems with sensing delay. *SPIE, Data Structures and Target Classification*, Vol. 1470.

Azor, R., Bar-Itzhack, I. Y., Deutschmann, J. K., and Harman, R. R. (2001). Angular-rate estimation using delayed quaternion measurements. *Journal of Guidance, Control, and Dynamics*, Vol. 24(No. 3).

Bak, T., Quottrup, M. M., Blanke, M., and Niemann, H. (2001). Rømer system definition phase attitude control subsystem algorithms. Technical Report roemer/iesp/sus/tn/0001(3), Aalborg University.

Bayard, D. S. (1996). Star tracker simulation model for pointing control studies with application to sirtf. Technical report, Jet Propulsion Laboratory.

Bhanderi, D. (2001). Laser metrology system for smart-2 with spherical mirror as formation reference. Technical report, European Space Agency.

Bhanderi, D., Andersen, T. B., Andersen, R., Poulsen, K., and Schroeder, C. (1999). Balancerende pind. Technical report, Aalborg University.

Bhanderi, D., Andersen, T. B., Michelsen, S. G., Schmidt, M. J., Schroeder, C., and Tietze, A. (2000). High precision attitude determination using gps. Technical report, Aalborg University.

DSRI (2001). Homepage of danish space research institute. Technical report, DSRI. www.dsri.dk.

Franklin, G. F., Powell, J. D., and Emami-Naeini, A. (1994). *Feedback Control of Dynamic Systems*. Adison-Wesley Publishing Company.

Grewal, M. S. (1990). Application of kalman filtering to the calibration and alignment of inertial navigation systems. Technical report, California State University.

Grewal, M. S. and Andrews, A. P. (1993). *Kalman Filtering Theory and Practice*. Prentice Hall.

Lefferts, E. J., Markely, F. L., and Shuster, M. D. (1982). Kalman filtering for spacecraft attitude estimation.

Maybeck, P. S. (1982). *Stochastic models, estimation, and control*, volume Vol. 2. Academic Press Limited.

Noteborn, R. (2001). Rømer attitude control system user requirements document. Technical report, Terma.

Paulsen, T. E. and Maresi, L. (2000). Calibration and verification of the terma star tracker for the nemo satellite. Technical report, Terma A/S.

Shuster, M. D. and Oh, S. (1981). Three-axis attitude determinnation from vector observations.

SYSTRON (2001). Homepage of systron donner intertial division. Technical report, Systron Donner Inertial Division. www.systron.com.

Wertz, J. R., editor (1978). *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers.

# Part VI

# Appendices

# Appendix A

# Quaternions

This chapter is a brief description of the quaternions and their algebra. It is based on [Wertz, 1978], and modified to the notation used in this project.

Quaternions is one of many ways to represent attitude. The quaternion has the advantage of being without singularities for all attitudes.

A quaternion $\boldsymbol{q}$ is defined by its four vector elements $q_1$, $q_2$, $q_3$ and $q_4$, as

$$\boldsymbol{q} = q_4 + iq_1 + jq_2 + kq_3 = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{A.1}$$

where $i$, $j$ and $k$ are hyper imaginary numbers satisfying

$$i^2 = j^2 = k^2 = -1 \tag{A.2}$$
$$ij = -ji = k \tag{A.3}$$
$$jk = -kj = i \tag{A.4}$$
$$ki = -ik = j \tag{A.5}$$

The four parameters of a quaternion are subject to the constraint that

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{A.6}$$

which means that the quaternion has three degrees of freedom, corresponding to the minimum set of parameters needed for attitude representation [Wertz, 1978].

The last element $q_4$ of the quaternion $\boldsymbol{q}$ is named the scalar element, and the 1st, 2nd, and 3rd the complex elements. The complex part of the quaternion is written $\boldsymbol{q}_{1:3}$, hence a quaternion may be written

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_{1:3} \\ q_4 \end{bmatrix} \tag{A.7}$$

A rotation $\Delta\phi$ around a unit vector $\boldsymbol{u}$, is represented by the quaternion

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{u}\sin\left(\frac{\Delta\phi}{2}\right) \\ \cos\left(\frac{\Delta\phi}{2}\right) \end{bmatrix} \tag{A.8}$$

The complex conjugate of the quaternion is defined as

$$\boldsymbol{q}^* = q_4 - iq_1 - jq_2 - kq_3 = \begin{bmatrix} -\boldsymbol{q}_{1:3} \\ q_4 \end{bmatrix} \tag{A.9}$$

Note that

$$\boldsymbol{q}\boldsymbol{q}^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{A.10}$$

which is the unit quaternion representing the zero rotation, i.e. no rotation.

The product of two quaternions $\boldsymbol{q}$ and $\boldsymbol{q}'$ is defined in matrix form as

$$\boldsymbol{q}'' = \boldsymbol{q}'\boldsymbol{q} = \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{A.11}$$

Note that the multiplication of quaternions is not commutative, which is also the case for attitude matrices.

Representing the attitude of $\boldsymbol{q}$ by the attitude matrix $\boldsymbol{A}\left(\boldsymbol{q}\right)$, the rotation sequence of Equation A.11, can be written in terms of the associated attitude matrices, as

$$\boldsymbol{A}\left(\boldsymbol{q}''\right) = \boldsymbol{A}\left(\boldsymbol{q}'\right)\boldsymbol{A}\left(\boldsymbol{q}\right) \tag{A.12}$$

where $\boldsymbol{A}\left(\boldsymbol{q}\right)$ is given by

$$\begin{aligned}
\boldsymbol{A}\left(\boldsymbol{q}\right) &= \left(q_4^2 - \left|\boldsymbol{q}_{1:3}\right|^2\right)\mathbf{1}_{3\times3} + 2\boldsymbol{q}_{1:3}\boldsymbol{q}_{1:3}^{\mathrm{T}} - 2q_4\mathbf{S}\left(\boldsymbol{q}_{1:3}\right) && \text{(A.13)}\\
&= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2\left[q_1q_2 + q_3q_4\right] & 2\left[q_1q_3 - q_2q_4\right] \\ 2\left[q_1q_2 - q_3q_4\right] & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2\left[q_2q_3 + q_1q_4\right] \\ 2\left[q_1q_3 + q_2q_4\right] & 2\left[q_3q_3 - q_1q_4\right] & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} && \text{(A.14)}
\end{aligned}$$

## A.1 Conversion to Euler Angles

The 1-2-3 Euler sequence of pitch $\theta$, yaw $\psi$, and roll $\phi$, is given by the attitude matrix

$$\boldsymbol{A}\left(\theta,\psi,\phi\right) = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\phi & \mathrm{c}\theta\mathrm{s}\phi\mathrm{s}\psi + \mathrm{s}\theta\mathrm{c}\psi & -\mathrm{c}\theta\mathrm{s}\phi\mathrm{c}\psi + \mathrm{s}\theta\mathrm{s}\psi \\ -\mathrm{s}\theta\mathrm{c}\phi & -\mathrm{s}\theta\mathrm{s}\phi\mathrm{s}\psi + \mathrm{c}\theta\mathrm{c}\psi & \mathrm{s}\theta\mathrm{s}\phi\mathrm{c}\psi + \mathrm{c}\theta\mathrm{s}\psi \\ \mathrm{s}\phi & -\mathrm{c}\theta\mathrm{s}\psi & \mathrm{c}\phi\mathrm{c}\psi \end{bmatrix} \tag{A.15}$$

From the above matrix, the Euler angles are found to be

$$\begin{aligned}
\theta &= \text{atan2}\left(\frac{-\boldsymbol{A}_{2,1}\left(\theta,\psi,\phi\right)}{\cos\left(\phi\right)}, \frac{\boldsymbol{A}_{1,1}\left(\theta,\psi,\phi\right)}{\cos\left(\phi\right)}\right) && \text{(A.16)}\\
\psi &= \text{atan2}\left(\frac{-\boldsymbol{A}_{3,2}\left(\theta,\psi,\phi\right)}{\cos\left(\phi\right)}, \frac{\boldsymbol{A}_{3,3}\left(\theta,\psi,\phi\right)}{\cos\left(\phi\right)}\right) && \text{(A.17)}\\
\phi &= \text{atan2}\left(\boldsymbol{A}_{3,1}\left(\theta,\psi,\phi\right), \cos\left(\phi\right)\right) && \text{(A.18)}
\end{aligned}$$

where

$$\cos\left(\phi\right) = \sqrt{\boldsymbol{A}_{1,1}\left(\theta,\psi,\phi\right)^2 + \boldsymbol{A}_{2,1}\left(\theta,\psi,\phi\right)^2} \tag{A.19}$$

Inserting the values of the quaternion attitude matrix representation $\boldsymbol{A}\left(\boldsymbol{q}\right)$ of Equation A.13, yields the conversion from quaternion the 1-2-3 Euler angle sequence, given by

$$\theta = \text{atan2}\left(\frac{-2\left[q_1 q_2 - q_3 q_4\right]}{\cos\left(\phi\right)}, \frac{q_1^2 - q_2^2 - q_3^2 + q_4^2}{\cos\left(\phi\right)}\right) \tag{A.20}$$

$$\psi = \text{atan2}\left(\frac{-2\left[q_3 q_3 - q_1 q_4\right]}{\cos\left(\phi\right)}, \frac{-q_1^2 - q_2^2 + q_3^2 + q_4^2}{\cos\left(\phi\right)}\right) \tag{A.21}$$

$$\phi = \text{atan2}\left(2\left[q_1 q_3 + q_2 q_4\right], \cos\left(\phi\right)\right) \tag{A.22}$$

where

$$\cos\left(\phi\right) = \sqrt{\left[q_1^2 - q_2^2 - q_3^2 + q_4^2\right]^2 + 2^2\left[q_1 q_2 - q_3 q_4\right]^2} \tag{A.23}$$

Appendix B

# The q-Method

This appendix describes the q-Method used for solving Wahba's problem. The appendix has been previously published in [Bhanderi et al., 2000].

The q-Method for attitude determination was developed by Davenport in 1968 and is based on earlier work done by Wahba [Wertz, 1978]. The algorithm is very popular due to its speed and efficiency, when given a set of $n \geq 2$ vector observations. The vector observation $\boldsymbol{e}_j^{\mathrm{B}}$, is the LOS vector given in body system. The following loss function is minimized

$$L\left(\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\right) = \frac{1}{2}\sum_{j=1}^{k} w_j |\boldsymbol{e}_j^{\mathrm{B}} - \boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\boldsymbol{e}_j^{\mathrm{R}}|^2 \tag{B.1}$$

where $w_j$ is the weight of the $j$'th vector observation, $\boldsymbol{e}_j^{\mathrm{R}}$ is the LOS vector in the reference frame and $\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}$ is the orthonormal rotation matrix, representing the rotation from reference to body frame, which is sought. This makes the loss function a weighted sum squared of the difference between the measured and the transformed LOS vectors. To simplify Wahba's problem it is necessary to expand it to

$$L\left(\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\right) = \frac{1}{2}\sum_{j=1}^{k} w_j | \left(\boldsymbol{e}_j^{\mathrm{B}}\right)^{\mathrm{T}}\boldsymbol{e}_j^{\mathrm{B}} + \left(\boldsymbol{e}_j^{\mathrm{R}}\right)^{\mathrm{T}}\left(\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\right)^{\mathrm{T}}\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\boldsymbol{e}_j^{\mathrm{R}} - 2\left(\boldsymbol{e}_j^{\mathrm{B}}\right)^{\mathrm{T}}\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\boldsymbol{e}_j^{\mathrm{R}}|$$

where $\left(\boldsymbol{e}_j^{\mathrm{B}}\right)^{\mathrm{T}}\boldsymbol{e}_j^{\mathrm{B}}$ is constant and $\left(\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\right)^{\mathrm{T}}\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}$ is the identity matrix. The loss function, $L\left(\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\right)$, may then be written as

$$L\left(\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\right) = -\sum_{j=1}^{k}\boldsymbol{t}_j^{\mathrm{T}}\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\boldsymbol{s}_j + \text{constant terms} \tag{B.2}$$

where the unnormalized vectors $\boldsymbol{t}_j$ and $\boldsymbol{s}_j$ are defined as

$$\begin{aligned}\boldsymbol{t}_j &= \sqrt{w_j}\boldsymbol{e}_j^{\mathrm{B}} \\ \boldsymbol{s}_j &= \sqrt{w_j}\boldsymbol{e}_j^{\mathrm{R}}\end{aligned}$$

The loss function is at a minimum when the following expression is at a maximum.

$$L'\left(\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\right) = \sum_{j=1}^{k} \boldsymbol{t}_j^{\mathrm{T}} \boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}} \boldsymbol{s}_j = \mathrm{tr}\left(\boldsymbol{T}^{\mathrm{T}} \boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}} \boldsymbol{S}\right) \tag{B.3}$$

where the matrices $\boldsymbol{T}$ and $\boldsymbol{S}$ are defined as

$$\begin{aligned}\boldsymbol{T} &= \left[\boldsymbol{t}_1 \;|\boldsymbol{t}_2 \;|\cdots \;|\boldsymbol{t}_k\right] \\ \boldsymbol{S} &= \left[\boldsymbol{s}_1 \;|\boldsymbol{s}_2 \;|\cdots \;|\boldsymbol{s}_k\right]\end{aligned}$$

The maximization of $L'\left(\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}\right)$ is complicated since the nine elements of $\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}$ are subject to six constraints [Shuster and Oh, 1981]. It is therefore convenient to express $\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}$ in terms of a quaternion $\boldsymbol{q}$.

$$\boldsymbol{q} = \begin{bmatrix}\boldsymbol{q}_{1:3} \\ q_4\end{bmatrix} \tag{B.4}$$

According to Appendix A the rotation matrix $\boldsymbol{A}_{\mathrm{R}}^{\mathrm{B}}$ is defined in terms of quaternions as

$$\boldsymbol{A}\left(\boldsymbol{q}\right) = \left(q_4^2 - |\boldsymbol{q}_{1:3}|^2\right) \mathbf{1}_{3\times 3} + 2\boldsymbol{q}_{1:3}\boldsymbol{q}_{1:3}^{\mathrm{T}} - 2q_4 \mathbf{S}\left(\boldsymbol{q}_{1:3}\right) \tag{B.5}$$

Substituting Equation B.5 into B.3 gives the modified expression for the loss function [Wertz, 1978, p. 428]

$$L'\left(\boldsymbol{q}\right) = \boldsymbol{q}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{q} \tag{B.6}$$

where $\boldsymbol{K}$ is defined as

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{U} - \sigma\boldsymbol{I} & \vdots & \boldsymbol{Z} \\ \cdots\cdots & \cdots & \cdots \\ \boldsymbol{Z}^{\mathrm{T}} & \vdots & \sigma \end{bmatrix} \tag{B.7}$$

The contained matrices are defined as

$$\begin{aligned}\boldsymbol{U} &= \boldsymbol{B}^{\mathrm{T}} + \boldsymbol{B} \\ \boldsymbol{Z} &= \left[B_{2,3} - B_{3,2} | B_{3,1} - B_{1,3} |\; B_{1,2} - B_{2,1}\right]^{\mathrm{T}} \\ \sigma &= \mathrm{tr}\left(\boldsymbol{B}\right)\end{aligned}$$

where

$$\boldsymbol{B} = \boldsymbol{T}\boldsymbol{S}^{\mathrm{T}} \tag{B.8}$$

The problem of determining the optimal attitude has been reduced to finding the quaternion that maximizes equation B.6. The normalization constraint, $\boldsymbol{q}^{\mathrm{T}}\boldsymbol{q} = 1$, can be taken into account by using Lagrange multipliers [Wertz, 1978]. A new function $g(\boldsymbol{q})$ is defined

$$g(\boldsymbol{q}) = \boldsymbol{q}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{q} - \xi\boldsymbol{q}^{\mathrm{T}}\boldsymbol{q} \tag{B.9}$$

where $\xi$ is the Lagrange multiplier, $g(\boldsymbol{q})$ is maximized without constraints and $\xi$ is chosen to satisfy the normalization constraint.

By differentiating Equation B.9 with respect to $\boldsymbol{q}^{\mathrm{T}}$ and setting the result equal to zero, an eigenvector equation is obtained [Shuster and Oh, 1981]

$$\boldsymbol{K}\boldsymbol{q} = \xi\boldsymbol{q} \tag{B.10}$$

The optimal quaternion which determines the optimal attitude matrix, in accordance with Equation B.5, is an eigenvector of $\boldsymbol{K}$. Substitution of equation B.10 into equation B.6 gives

$$L'(\boldsymbol{q}) = \boldsymbol{q}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{q} = \boldsymbol{q}^{\mathrm{T}}\xi\boldsymbol{q} = \xi \tag{B.11}$$

Thus, $L'(\boldsymbol{q})$ will be maximized if the optimal quaternion is chosen to be the eigenvector of $\boldsymbol{K}$ belonging to the largest eigenvalue.

# Appendix C

# Linearization of Attitude Equations

In this appendix the attitude dynamic and kinematic equations are linearized around a working point of the states, previously published in [Bhanderi, 2001]. As a consequence of linearization, the system description will contain the small signals of the states.

The non-linear system matrix $\boldsymbol{f}\left(\boldsymbol{x}, t\right)$ describing the satellite's rotation, is found by combining Equations 7.9 and 7.27 to be

$$\boldsymbol{f}\left(\boldsymbol{x}, t\right) = \begin{bmatrix} \frac{1}{2}\boldsymbol{\Omega}\boldsymbol{q} \\ \boldsymbol{J}^{-1}\left(-\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}\right) \end{bmatrix} \tag{C.1}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{bmatrix} \tag{C.2}$$

An expression in the linearized form $\dot{\boldsymbol{x}} = \boldsymbol{F}\left(t\right)\boldsymbol{x}$ is sought. The kinematic and dynamic equations are linearized separately in the following.

## C.1 Kinematic Equation

For the linearization of the kinematic equation, the attitude quaternion of the satellite $\boldsymbol{q}$ is written, in terms of a working point $\bar{\boldsymbol{q}}$ and small signal $\tilde{\boldsymbol{q}}$, as

$$\begin{aligned} \boldsymbol{q} &= \tilde{\boldsymbol{q}}\bar{\boldsymbol{q}} \\ &\Updownarrow \\ \tilde{\boldsymbol{q}} &= \boldsymbol{q}\bar{\boldsymbol{q}}^* \end{aligned} \tag{C.3}$$

where $\bar{\boldsymbol{q}}^*$ is the complex conjugate of $\bar{\boldsymbol{q}}$. Recalling the definition of $\boldsymbol{\Omega}$ in Equation 7.6 on page 72 and the definition of quaternion products, the kinematic equation can be written in terms of a quaternion product, as

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q}_{\boldsymbol{\omega}}\boldsymbol{q} \tag{C.4}$$

where the quaternion $\boldsymbol{q}_{\boldsymbol{\omega}}$ is defined by

$$\boldsymbol{q}_{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \tag{C.5}$$

Using the chain rule and Equation C.4, the derivative of the small signal attitude quaternion of Equation C.3 can be expressed as

$$\begin{aligned} \dot{\tilde{\boldsymbol{q}}} &= \boldsymbol{q}\dot{\bar{\boldsymbol{q}}}^* + \dot{\boldsymbol{q}}\bar{\boldsymbol{q}}^* \\ &= \frac{1}{2}\left[\boldsymbol{q}\left(\boldsymbol{q}_{\bar{\boldsymbol{\omega}}}\bar{\boldsymbol{q}}\right)^* + \boldsymbol{q}_{\boldsymbol{\omega}}\boldsymbol{q}\bar{\boldsymbol{q}}^*\right] \\ &= \frac{1}{2}\left[-\boldsymbol{q}\bar{\boldsymbol{q}}^*\boldsymbol{q}_{\bar{\boldsymbol{\omega}}} + \boldsymbol{q}_{\boldsymbol{\omega}}\boldsymbol{q}\bar{\boldsymbol{q}}^*\right] \\ &= \frac{1}{2}\left[-\tilde{\boldsymbol{q}}\boldsymbol{q}_{\bar{\boldsymbol{\omega}}} + \boldsymbol{q}_{\boldsymbol{\omega}}\tilde{\boldsymbol{q}}\right] \end{aligned} \tag{C.6}$$

where $\boldsymbol{q}_{\bar{\boldsymbol{\omega}}}$ is defined as in Equation C.5, where the working point of the angular velocity $\bar{\boldsymbol{\omega}}$ is inserted. Note that the complex conjugate of a quaternion represents the opposite rotation. Hence the reverse rotation defined by a sequence of rotations, is the reverse sequence of each rotation complex conjugated. The complex conjugate of $\boldsymbol{q}_{\bar{\boldsymbol{\omega}}}$ is simply $-\boldsymbol{q}_{\bar{\boldsymbol{\omega}}}$, since the scalar part is zero.

The angular velocity is defined, in terms of a working point $\bar{\boldsymbol{\omega}}$ and small signal $\tilde{\boldsymbol{\omega}}$, as

$$\omega = \bar{\omega} + \tilde{\omega} \tag{C.7}$$

hence $\boldsymbol{q}_{\omega}$ can be written

$$\begin{aligned} \boldsymbol{q}_{\omega} &= \begin{bmatrix} \bar{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \\ &= \boldsymbol{q}_{\bar{\omega}} + \boldsymbol{q}_{\tilde{\omega}} \end{aligned} \tag{C.8}$$

Inserting Equation C.8 in Equation C.6, and recognizing that the associative rule applies for quaternions, yields

$$\dot{\tilde{\boldsymbol{q}}} = \frac{1}{2}\left[-\tilde{\boldsymbol{q}}\boldsymbol{q}_{\bar{\omega}} + \boldsymbol{q}_{\bar{\omega}}\tilde{\boldsymbol{q}}\right] + \frac{1}{2}\boldsymbol{q}_{\tilde{\omega}}\tilde{\boldsymbol{q}} \tag{C.9}$$

A quaternion can be expressed, in terms of a scalar part and a vector part, as

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_{1:3} \\ q_4 \end{bmatrix} \tag{C.10}$$

where $\boldsymbol{q}_{1:3}$ is a vector containing the 1st, 2nd, and 3rd element of $\boldsymbol{q}$, as described in Appendix A. If $\tilde{\boldsymbol{q}}$ is a small rotation, $\tilde{q}_4$ approaches one and $\tilde{\boldsymbol{q}}_{1:3}$ approaches zero. Hence the quaternion product of the last term in Equation C.9 can be approximated by

$$\begin{aligned} \boldsymbol{q}_{\tilde{\omega}}\tilde{\boldsymbol{q}} &= \begin{bmatrix} -\mathbf{S}\left(\tilde{\boldsymbol{\omega}}\right) & \tilde{\boldsymbol{\omega}} \\ -\tilde{\boldsymbol{\omega}}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{q}}_{1:3} \\ q_4 \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{S}\left(\tilde{\boldsymbol{\omega}}\right)\tilde{\boldsymbol{q}}_{1:3} + \tilde{\boldsymbol{\omega}} q_4 \\ -\tilde{\boldsymbol{\omega}}^{\mathrm{T}}\tilde{\boldsymbol{q}}_{1:3} \end{bmatrix} \\ &\approx \boldsymbol{q}_{\tilde{\omega}} \end{aligned} \tag{C.11}$$

where $\mathbf{S}\left(\tilde{\boldsymbol{\omega}}\right)$ is the cross product matrix function, defined in Equation 7.18 on page 75, applied on $\tilde{\boldsymbol{\omega}}$. The terms $\tilde{\boldsymbol{q}}\boldsymbol{q}_{\bar{\omega}}$ and $\boldsymbol{q}_{\bar{\omega}}\tilde{\boldsymbol{q}}$ of Equation C.9, can be expressed as

$$\begin{aligned} \tilde{\boldsymbol{q}}\boldsymbol{q}_{\bar{\omega}} &= \begin{bmatrix} -\mathbf{S}\left(\tilde{\boldsymbol{q}}_{1:3}\right) + \tilde{q}_4\mathbf{1}_{3\times 3} & \tilde{\boldsymbol{q}}_{1:3} \\ -\tilde{\boldsymbol{q}}_{1:3}^{\mathrm{T}} & \tilde{q}_4 \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{S}\left(\tilde{\boldsymbol{q}}_{1:3}\right)\bar{\boldsymbol{\omega}} + \tilde{q}_4\bar{\boldsymbol{\omega}} \\ -\tilde{\boldsymbol{q}}_{1:3}^{\mathrm{T}}\bar{\boldsymbol{\omega}} \end{bmatrix} \end{aligned} \tag{C.12}$$

and

$$\begin{aligned}
\boldsymbol{q}_{\tilde{\boldsymbol{\omega}}}\tilde{\boldsymbol{q}} &= \begin{bmatrix} -\mathbf{S}\left(\bar{\boldsymbol{\omega}}\right) & \bar{\boldsymbol{\omega}} \\ -\bar{\boldsymbol{\omega}}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{q}}_{1:3} \\ q_4 \end{bmatrix} \\
&= \begin{bmatrix} -\mathbf{S}\left(\bar{\boldsymbol{\omega}}\right)\tilde{\boldsymbol{q}}_{1:3} + \bar{\boldsymbol{\omega}} q_4 \\ -\bar{\boldsymbol{\omega}}^{\mathrm{T}}\tilde{\boldsymbol{q}}_{1:3} \end{bmatrix} \qquad (\mathrm{C.13}) \\
& \qquad (\mathrm{C.14})
\end{aligned}$$

Inserting the approximation of Equation C.11 and the results of Equations C.13 and C.12 in Equation C.9, yields

$$\dot{\tilde{\boldsymbol{q}}} = \begin{bmatrix} -\mathbf{S}\left(\bar{\boldsymbol{\omega}}\right) \\ 0 \end{bmatrix} \tilde{\boldsymbol{q}} + \frac{1}{2}\boldsymbol{q}_{\tilde{\boldsymbol{\omega}}} \tag{C.15}$$

which is the sought linearized kinematic equation.

## C.2 Dynamic Equation

The dynamic equation is linearized using first order Taylor expansion around the working point $\bar{\boldsymbol{\omega}}$. The control torques are disregarded in the linearization. They are considered when using controller input, and may be added by the term $\boldsymbol{J}^{-1}\boldsymbol{n}_{\mathrm{ctrl}}$, which is already a linear term.

The derivative of the small signal angular velocity describes the linearized dynamics of the system, and can be expressed as

$$\begin{aligned}
\dot{\tilde{\boldsymbol{\omega}}} &\approx -\boldsymbol{J}^{-1} \left.\frac{\mathrm{d}\left(\boldsymbol{\omega}\times\boldsymbol{J}\boldsymbol{\omega}\right)}{\mathrm{d}\boldsymbol{\omega}}\right|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}} \tilde{\boldsymbol{\omega}} \\
&= -\boldsymbol{J}^{-1} \left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\omega}}\mathbf{S}\left(\boldsymbol{\omega}_{\mathrm{s}}\right)\boldsymbol{J}\boldsymbol{\omega}_{\mathrm{s}}\right|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}} \tilde{\boldsymbol{\omega}} \\
&= -\boldsymbol{J}^{-1} \left[ \left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\omega}}\mathbf{S}\left(\boldsymbol{\omega}\right)\right|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}} \boldsymbol{J}\bar{\boldsymbol{\omega}} + \mathbf{S}\left(\bar{\boldsymbol{\omega}}\right)\boldsymbol{J} \left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\omega}}\boldsymbol{\omega}\right|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}} \right] \tilde{\boldsymbol{\omega}} \\
&= -\boldsymbol{J}^{-1} \left[ \left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\omega}}\mathbf{S}\left(\boldsymbol{\omega}\right)\boldsymbol{J}\bar{\boldsymbol{\omega}}\right|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}} + \left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\omega}}\mathbf{S}\left(\bar{\boldsymbol{\omega}}\right)\boldsymbol{J}\boldsymbol{\omega}\right|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}} \right] \tilde{\boldsymbol{\omega}} \\
&= \boldsymbol{J}^{-1} \left[ \left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\omega}}\mathbf{S}\left(\boldsymbol{J}\bar{\boldsymbol{\omega}}\right)\boldsymbol{\omega}\right|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}} - \left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\omega}}\mathbf{S}\left(\bar{\boldsymbol{\omega}}\right)\boldsymbol{J}\boldsymbol{\omega}\right|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}} \right] \tilde{\boldsymbol{\omega}} \\
&= \boldsymbol{J}^{-1}\left[\mathbf{S}\left(\boldsymbol{J}\bar{\boldsymbol{\omega}}\right) - \mathbf{S}\left(\bar{\boldsymbol{\omega}}\right)\boldsymbol{J}\right]\tilde{\boldsymbol{\omega}}
\end{aligned} \tag{C.16}$$

## C.3 Linear Attitude Model

The non-linear system can now be expressed in a linear state space model. However, observing Equation C.15, it is seen that the scalar element $q_4$ has zero dynamics. This is expected, since it can be approximated to one for small angles. The state vector is reduced to six states being the small signals of the states in the non-linear model, which is written

$$\boldsymbol{x} = \begin{bmatrix} \tilde{\boldsymbol{q}}_{1:3} \\ \tilde{\boldsymbol{\omega}} \end{bmatrix} \tag{C.17}$$

When obtaining the quaternion from the reduced state vector, the scalar element should be set to one and the result normalized, since this is not ensured by the linearized kinematic equation.

The linearized attitude model can now be expressed as

$$\begin{bmatrix} \dot{\tilde{\boldsymbol{q}}}_{1:3} \\ \dot{\tilde{\boldsymbol{\omega}}} \end{bmatrix} = \begin{bmatrix} -\mathbf{S}\left(\bar{\boldsymbol{\omega}}\right) & \frac{1}{2}\mathbf{1}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \boldsymbol{J}^{-1}\left[\mathbf{S}\left(\boldsymbol{J}\bar{\boldsymbol{\omega}}\right) - \mathbf{S}\left(\bar{\boldsymbol{\omega}}\right)\boldsymbol{J}\right] \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{q}}_{1:3} \\ \tilde{\boldsymbol{\omega}} \end{bmatrix} \tag{C.18}$$

# Part VII

# Annexes

Annex I

# Physical Properties of Rømer

Figure I.1 shows the exploded view of Rømer.

Figure I.2 shows the geometry of Rømer.

Figure I.3 shows the geometry of the Sun Protection Lid.

Figure I.4 illustrates the electrical layout of the hardware components on Rømer.

Table I.1 lists the assumed mass properties of Rømer.

| **Property** | **Value** |
|---|---|
| Spacecraft Mass | $m_{\text{s}} = 83.9\text{kg}$ |
| CoG wrt. origo (CoM) | $\boldsymbol{r}_{\text{CoG}} = \begin{bmatrix} 0.007 \\ -0.002 \\ 0.275 \end{bmatrix}$ m |
| Moments of Inertia wrt. CoM | $\boldsymbol{J}^{\text{SCB}} = \begin{bmatrix} 18.09 & 0 & 0 \\ 0 & 18.64 & 0 \\ 0 & 0 & 4.28 \end{bmatrix}$ kgm$^2$ |

**Table I.1**: *Assumed mass properties of Rømer, [Bak et al., 2001].*

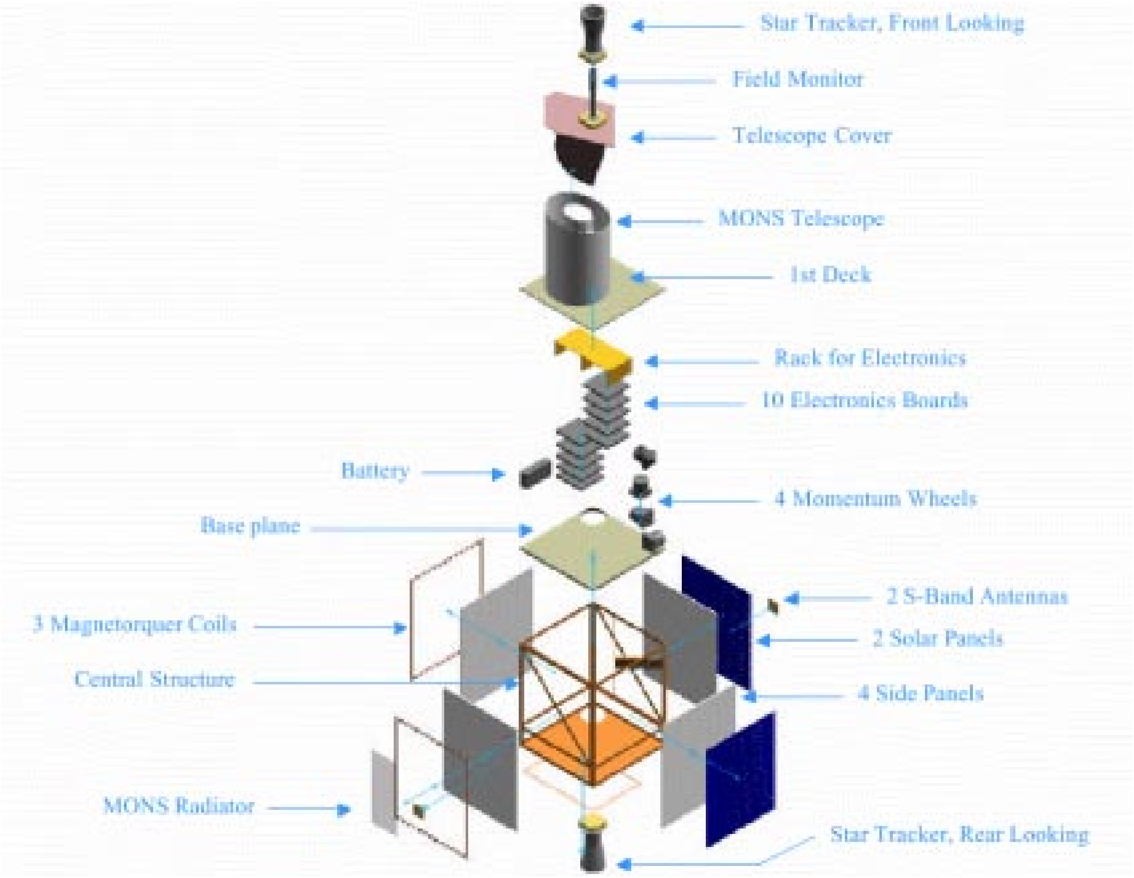


**Figure I.1**: *Exploded view of Rømer, [DSRI, 2001].*

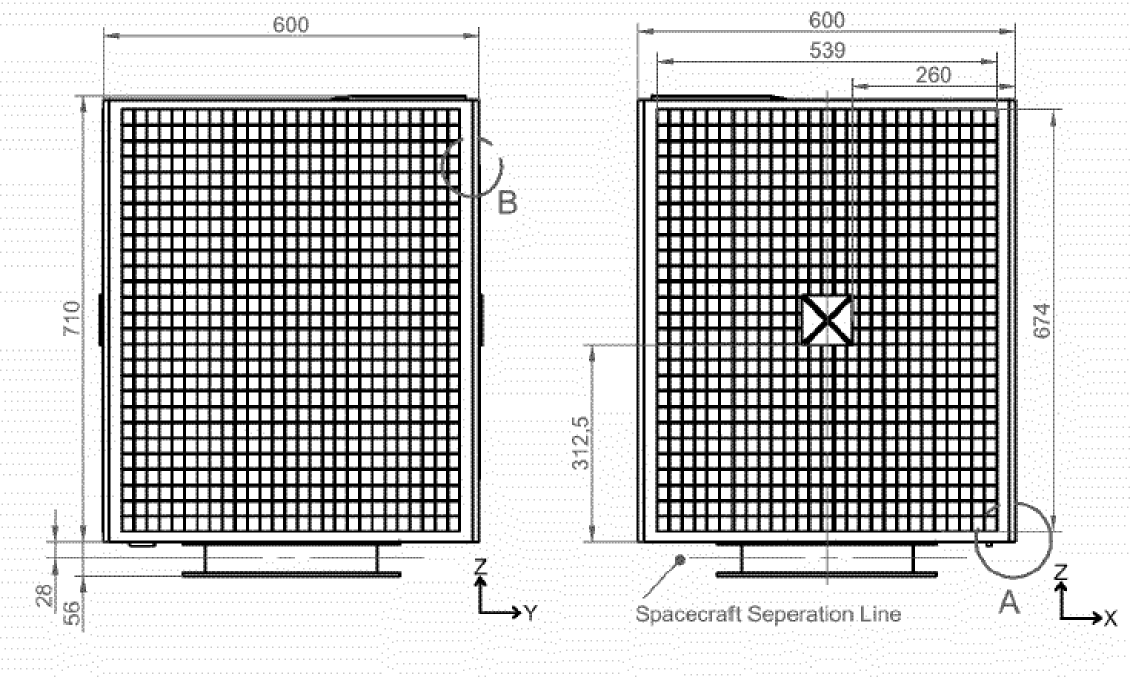


**Figure I.2**: *Basic geometry of Rømer, [Bak et al., 2001].*

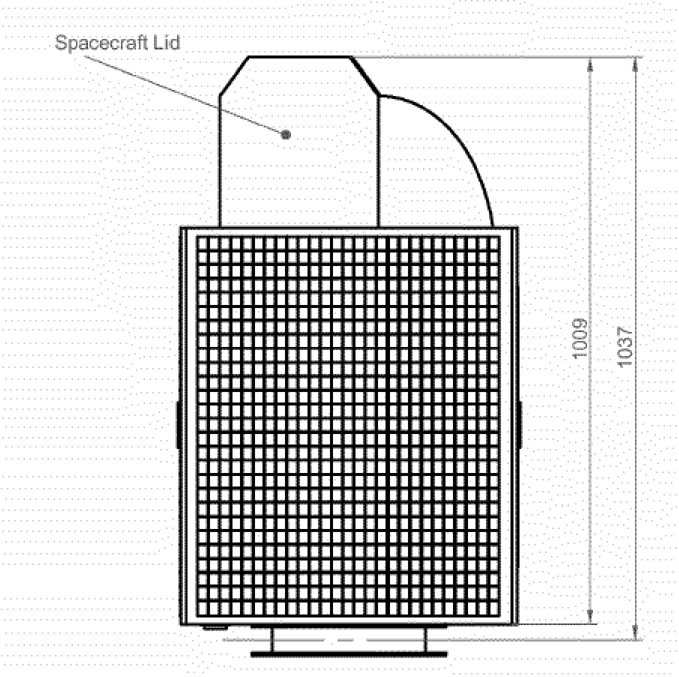


**Figure I.3**: *Geometry of the Sun Protection Lid on Rømer, [Bak et al., 2001].*

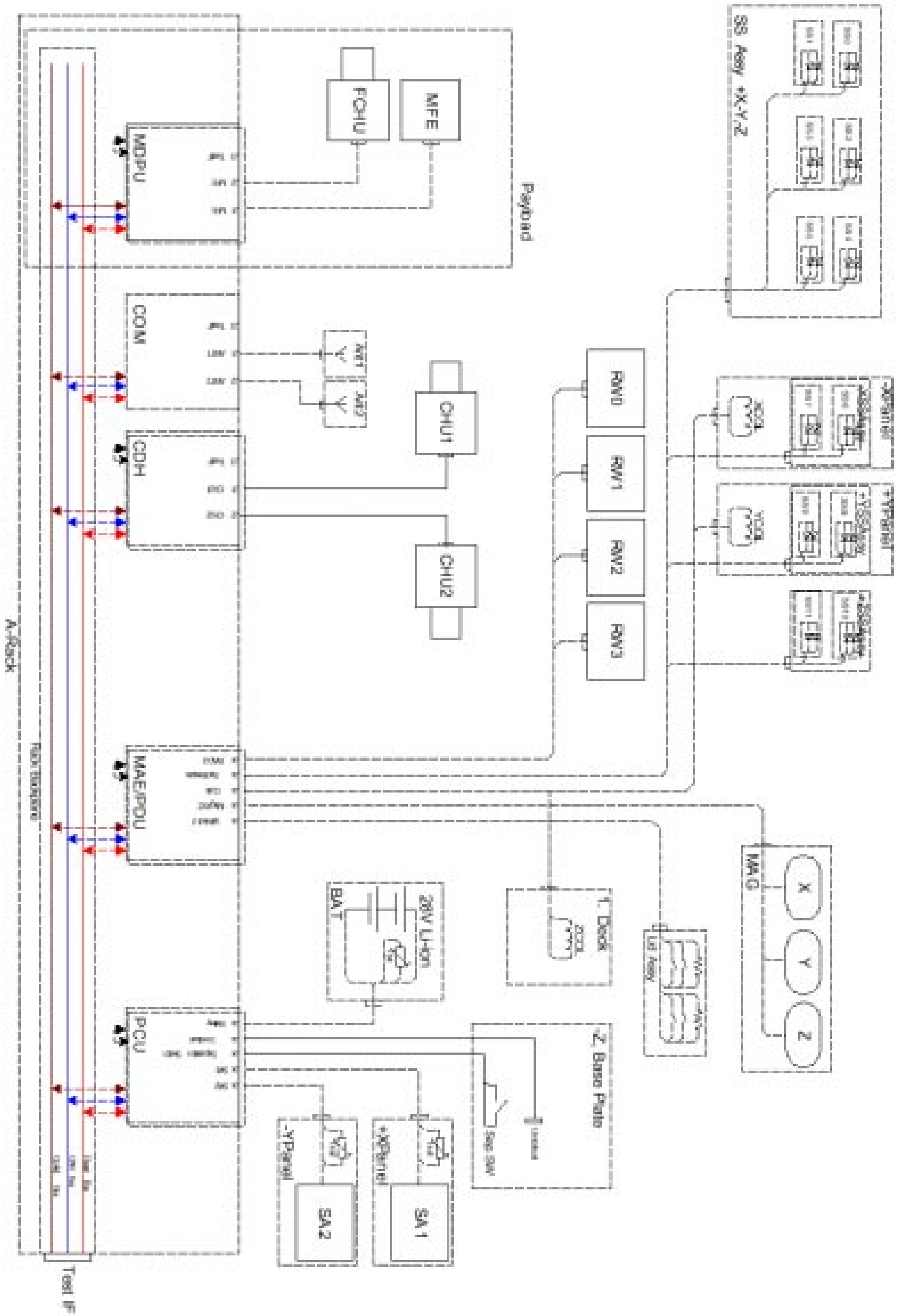
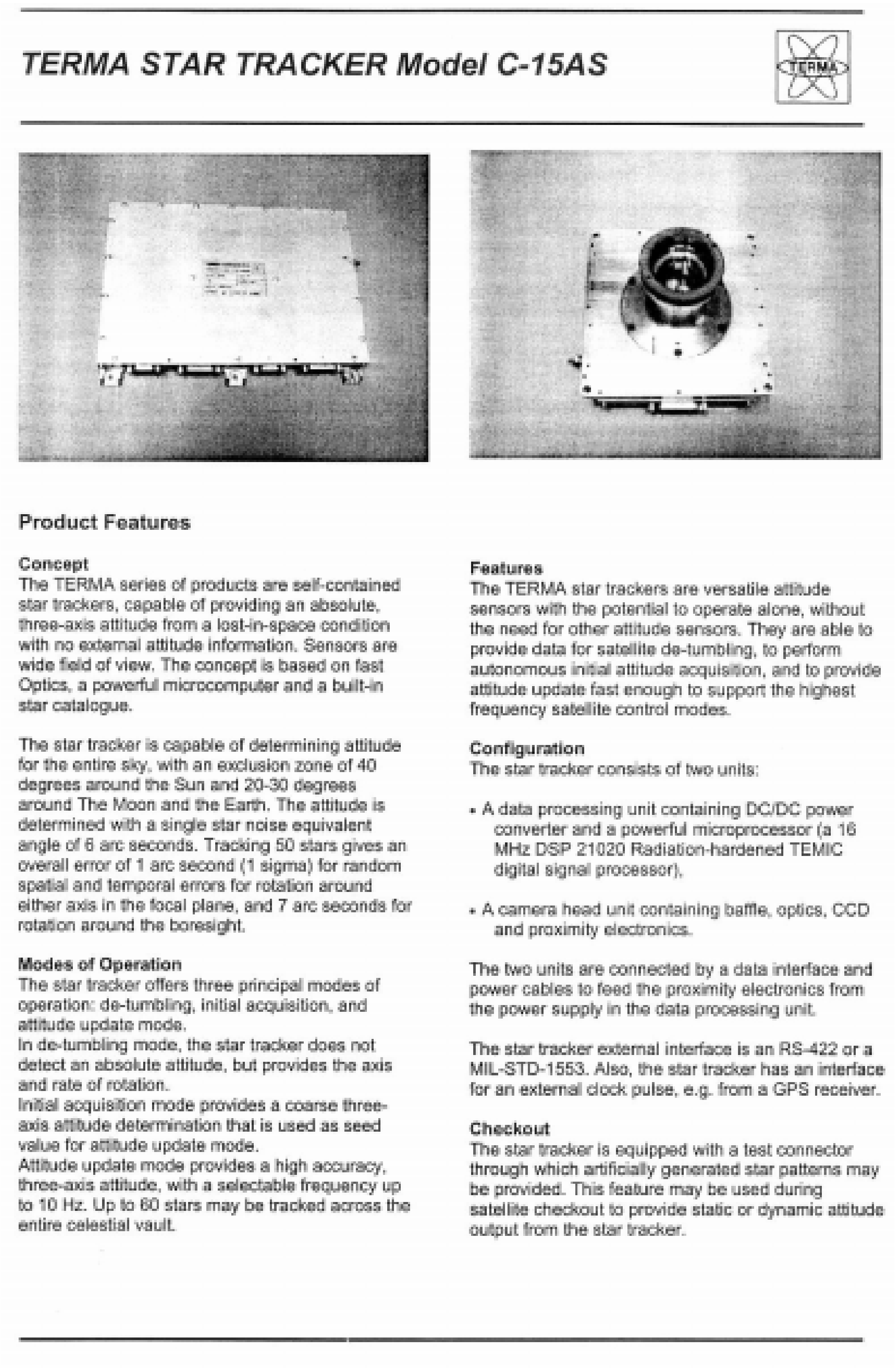


**Figure I.4**: *Electrical layout of the hardware components on Rømer.*

# Annex II

# NEMO Star Tracker Data Sheet



# TERMA STAR TRACKER Model C-15AS

## Product Features

### Concept

The TERMA series of products are self-contained star trackers, capable of providing an absolute, three-axis attitude from a lost-in-space condition with no external attitude information. Sensors are wide field of view. The concept is based on fast Optics, a powerful microcomputer and a built-in star catalogue.

The star tracker is capable of determining attitude for the entire sky, with an exclusion zone of 40 degrees around the Sun and 20-30 degrees around The Moon and the Earth. The attitude is determined with a single star noise equivalent angle of 6 arc seconds. Tracking 50 stars gives an overall error of 1 arc second (1 sigma) for random spatial and temporal errors for rotation around either axis in the focal plane, and 7 arc seconds for rotation around the boresight.

### Modes of Operation

The star tracker offers three principal modes of operation: de-tumbling, initial acquisition, and attitude update mode.
In de-tumbling mode, the star tracker does not detect an absolute attitude, but provides the axis and rate of rotation.
Initial acquisition mode provides a coarse three-axis attitude determination that is used as seed value for attitude update mode.
Attitude update mode provides a high accuracy, three-axis attitude, with a selectable frequency up to 10 Hz. Up to 60 stars may be tracked across the entire celestial vault.

### Features

The TERMA star trackers are versatile attitude sensors with the potential to operate alone, without the need for other attitude sensors. They are able to provide data for satellite de-tumbling, to perform autonomous initial attitude acquisition, and to provide attitude update fast enough to support the highest frequency satellite control modes.

### Configuration

The star tracker consists of two units:

- A data processing unit containing DC/DC power converter and a powerful microprocessor (a 16 MHz DSP 21020 Radiation-hardened TEMIC digital signal processor).

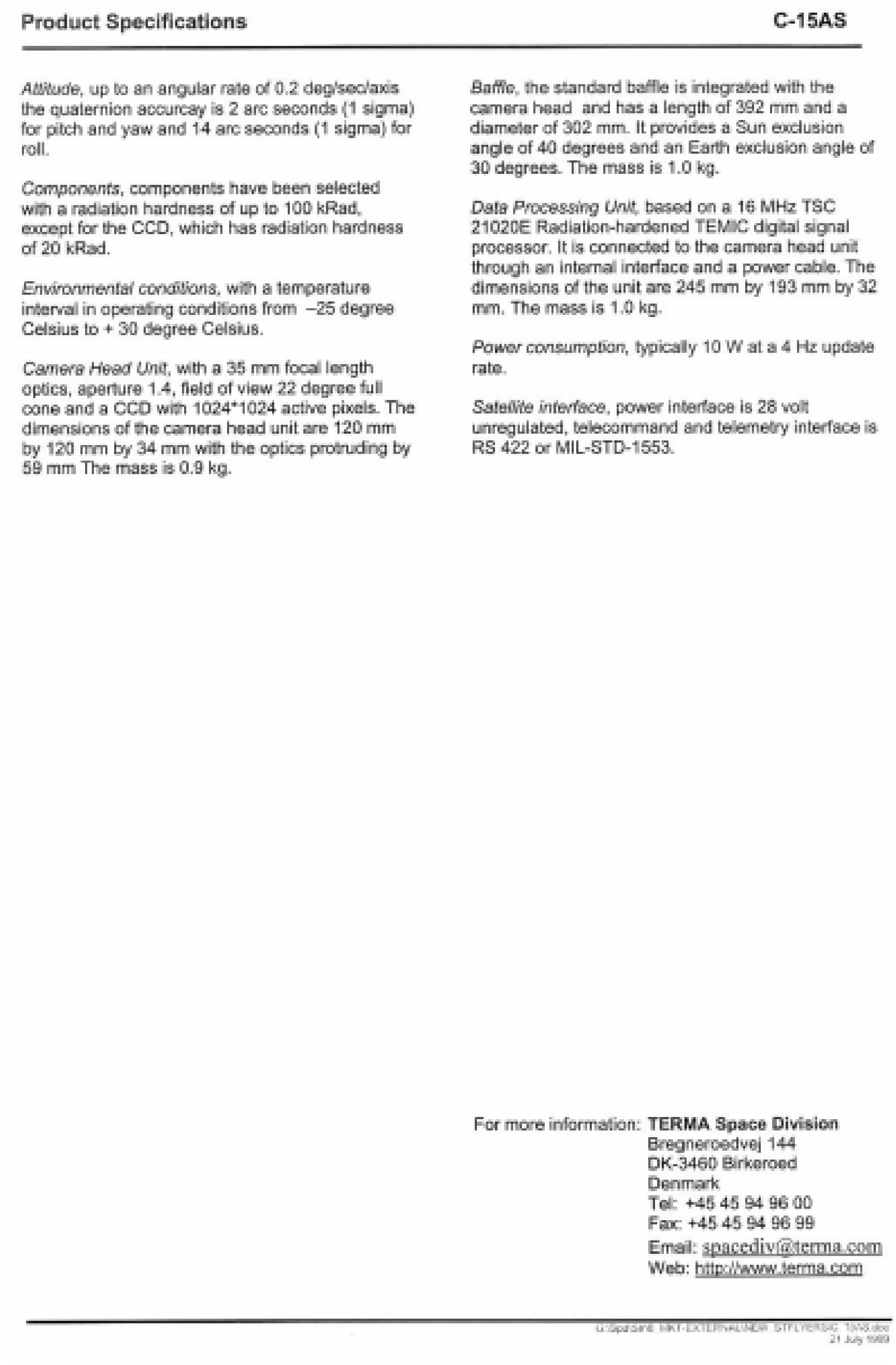
- A camera head unit containing baffle, optics, CCD and proximity electronics.

The two units are connected by a data interface and power cables to feed the proximity electronics from the power supply in the data processing unit.

The star tracker external interface is an RS-422 or a MIL-STD-1553. Also, the star tracker has an interface for an external clock pulse, e.g. from a GPS receiver.

### Checkout

The star tracker is equipped with a test connector through which artificially generated star patterns may be provided. This feature may be used during satellite checkout to provide static or dynamic attitude output from the star tracker.



**Product Specifications** **C-15AS**

*Attitude*, up to an angular rate of 0.2 deg/sec/axis the quaternion accurcay is 2 arc seconds (1 sigma) for pitch and yaw and 14 arc seconds (1 sigma) for roll.

*Components*, components have been selected with a radiation hardness of up to 100 kRad, except for the CCD, which has radiation hardness of 20 kRad.

*Environmental conditions*, with a temperature interval in operating conditions from –25 degree Celsius to + 30 degree Celsius.

*Camera Head Unit*, with a 35 mm focal length optics, aperture 1.4, field of view 22 degree full cone and a CCD with 1024*1024 active pixels. The dimensions of the camera head unit are 120 mm by 120 mm by 34 mm with the optics protruding by 59 mm The mass is 0.9 kg.

*Baffle*, the standard baffle is integrated with the camera head and has a length of 392 mm and a diameter of 302 mm. It provides a Sun exclusion angle of 40 degrees and an Earth exclusion angle of 30 degrees. The mass is 1.0 kg.

*Data Processing Unit*, based on a 16 MHz TSC 21020E Radiation-hardened TEMIC digital signal processor. It is connected to the camera head unit through an internal interface and a power cable. The dimensions of the unit are 245 mm by 193 mm by 32 mm. The mass is 1.0 kg.

*Power consumption*, typically 10 W at a 4 Hz update rate.

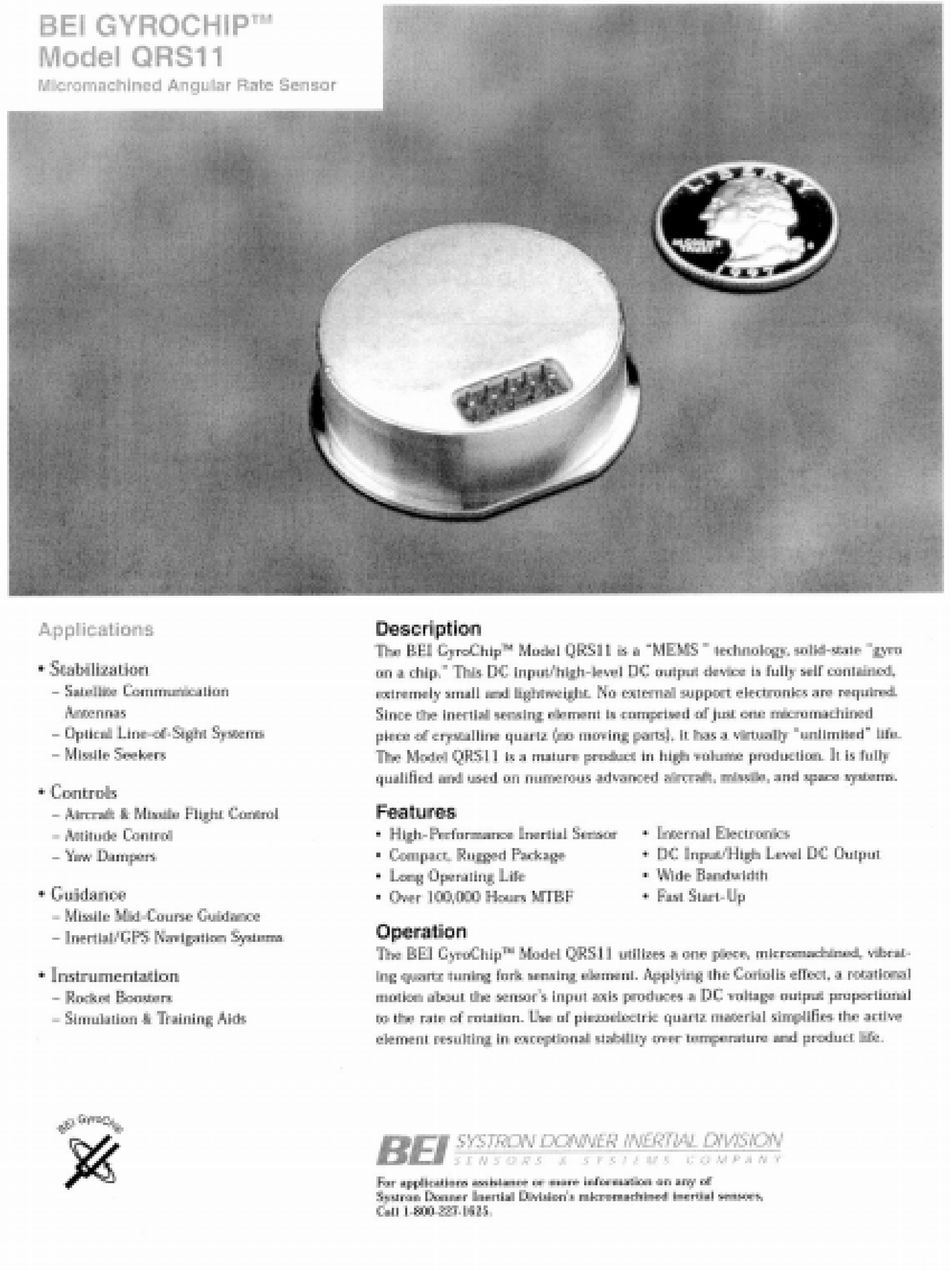
*Satellite interface*, power interface is 28 volt unregulated, telecommand and telemetry interface is RS 422 or MIL-STD-1553.

For more information: **TERMA Space Division**
Bregneroedvej 144
DK-3460 Birkeroed
Denmark
Tel: +45 45 94 96 00
Fax: +45 45 94 96 99
Email: spacediv@terma.com
Web: http://www.terma.com

[illegible]
21 July 1999

# Annex III

# QRS11Pro Rate Gyro Data Sheet



## BEI GYROCHIP™ Model QRS11

Micromachined Angular Rate Sensor

### Applications

- Stabilization
  - Satellite Communication Antennas
  - Optical Line-of-Sight Systems
  - Missile Seekers
- Controls
  - Aircraft & Missile Flight Control
  - Attitude Control
  - Yaw Dampers
- Guidance
  - Missile Mid-Course Guidance
  - Inertial/GPS Navigation Systems
- Instrumentation
  - Rocket Boosters
  - Simulation & Training Aids

### Description

The BEI GyroChip™ Model QRS11 is a "MEMS " technology, solid-state "gyro on a chip." This DC input/high-level DC output device is fully self contained, extremely small and lightweight. No external support electronics are required. Since the inertial sensing element is comprised of just one micromachined piece of crystalline quartz (no moving parts), it has a virtually "unlimited" life. The Model QRS11 is a mature product in high volume production. It is fully qualified and used on numerous advanced aircraft, missile, and space systems.

### Features

- High-Performance Inertial Sensor
- Compact, Rugged Package
- Long Operating Life
- Over 100,000 Hours MTBF
- Internal Electronics
- DC Input/High Level DC Output
- Wide Bandwidth
- Fast Start-Up

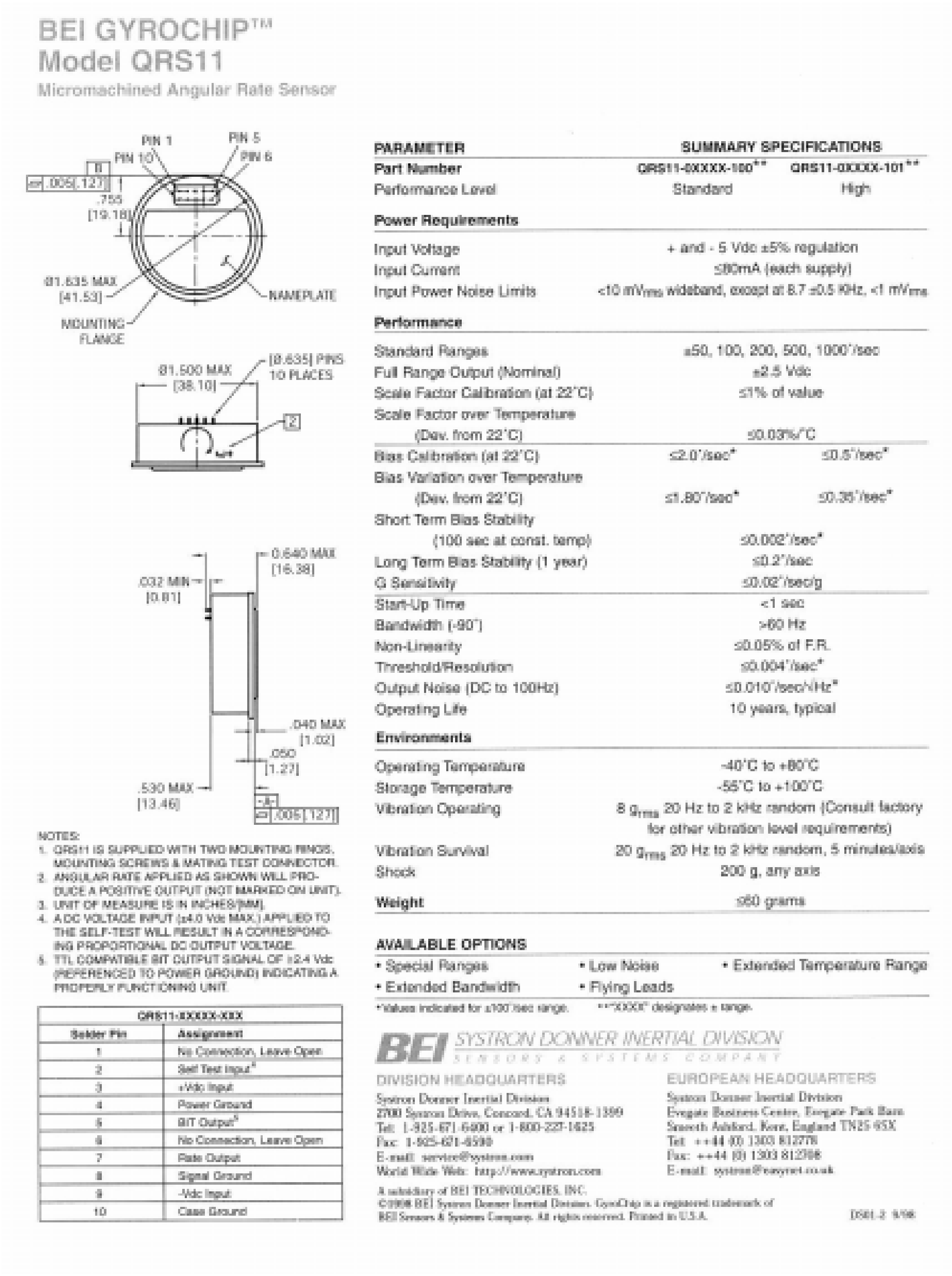
### Operation

The BEI GyroChip™ Model QRS11 utilizes a one piece, micromachined, vibrating quartz tuning fork sensing element. Applying the Coriolis effect, a rotational motion about the sensor's input axis produces a DC voltage output proportional to the rate of rotation. Use of piezoelectric quartz material simplifies the active element resulting in exceptional stability over temperature and product life.

BEI GyroChip

BEI SYSTRON DONNER INERTIAL DIVISION
SENSORS & SYSTEMS COMPANY

For applications assistance or more information on any of
Systron Donner Inertial Division's micromachined inertial sensors,
Call 1-800-227-1625.



PIN 1
PIN 5
PIN 10
PIN 6
B
.005[.127]
.755
[19.18]
Ø1.635 MAX
[41.53]
NAMEPLATE
MOUNTING
FLANGE
Ø1.500 MAX
[38.10]
[Ø.635] PINS
10 PLACES
2
0.640 MAX
[16.38]
.032 MIN
[0.81]
.040 MAX
[1.02]
.050
[1.27]
.530 MAX
[13.46]
-A-
.005[.127]

# BEI GYROCHIP™ Model QRS11

Micromachined Angular Rate Sensor

NOTES:

1. QRS11 IS SUPPLIED WITH TWO MOUNTING RINGS, MOUNTING SCREWS & MATING TEST CONNECTOR.
2. ANGULAR RATE APPLIED AS SHOWN WILL PRODUCE A POSITIVE OUTPUT (NOT MARKED ON UNIT).
3. UNIT OF MEASURE IS IN INCHES/[MM].
4. A DC VOLTAGE INPUT (±4.0 Vdc MAX.) APPLIED TO THE SELF-TEST WILL RESULT IN A CORRESPONDING PROPORTIONAL DC OUTPUT VOLTAGE.
5. TTL COMPATIBLE BIT OUTPUT SIGNAL OF ≥2.4 Vdc (REFERENCED TO POWER GROUND) INDICATING A PROPERLY FUNCTIONING UNIT.

| QRS11-XXXXX-XXX | |
|---|---|
| **Solder Pin** | **Assignment** |
| 1 | No Connection, Leave Open |
| 2 | Self Test Input$^{4}$ |
| 3 | +Vdc Input |
| 4 | Power Ground |
| 5 | BIT Output$^{5}$ |
| 6 | No Connection, Leave Open |
| 7 | Rate Output |
| 8 | Signal Ground |
| 9 | -Vdc Input |
| 10 | Case Ground |

| PARAMETER | SUMMARY SPECIFICATIONS | |
|---|---|---|
| **Part Number** | QRS11-0XXXX-100** | QRS11-0XXXX-101** |
| Performance Level | Standard | High |
| **Power Requirements** | | |
| Input Voltage | + and - 5 Vdc ±5% regulation | |
| Input Current | ≤80mA (each supply) | |
| Input Power Noise Limits | <10 $mV_{rms}$ wideband, except at 8.7 ±0.5 KHz, <1 $mV_{rms}$ | |
| **Performance** | | |
| Standard Ranges | ±50, 100, 200, 500, 1000°/sec | |
| Full Range Output (Nominal) | ±2.5 Vdc | |
| Scale Factor Calibration (at 22°C) | ≤1% of value | |
| Scale Factor over Temperature (Dev. from 22°C) | ≤0.03%/°C | |
| Bias Calibration (at 22°C) | ≤2.0°/sec* | ≤0.5°/sec* |
| Bias Variation over Temperature (Dev. from 22°C) | ≤1.80°/sec* | ≤0.35°/sec* |
| Short Term Bias Stability (100 sec at const. temp) | ≤0.002°/sec* | |
| Long Term Bias Stability (1 year) | ≤0.2°/sec | |
| G Sensitivity | ≤0.02°/sec/g | |
| Start-Up Time | <1 sec | |
| Bandwidth (-90°) | >60 Hz | |
| Non-Linearity | ≤0.05% of F.R. | |
| Threshold/Resolution | ≤0.004°/sec* | |
| Output Noise (DC to 100Hz) | ≤0.010°/sec/√Hz* | |
| Operating Life | 10 years, typical | |
| **Environments** | | |
| Operating Temperature | -40°C to +80°C | |
| Storage Temperature | -55°C to +100°C | |
| Vibration Operating | 8 $g_{rms}$ 20 Hz to 2 kHz random (Consult factory for other vibration level requirements) | |
| Vibration Survival | 20 $g_{rms}$ 20 Hz to 2 kHz random, 5 minutes/axis | |
| Shock | 200 g, any axis | |
| **Weight** | ≤60 grams | |

## AVAILABLE OPTIONS

- Special Ranges
- Low Noise
- Extended Temperature Range
- Extended Bandwidth
- Flying Leads

*Values indicated for ±100°/sec range. **"XXXX" designates ± range.

BEI SYSTRON DONNER INERTIAL DIVISION
SENSORS & SYSTEMS COMPANY

DIVISION HEADQUARTERS

Systron Donner Inertial Division
2700 Systron Drive, Concord, CA 94518-1399
Tel: 1-925-671-6400 or 1-800-227-1625
Fax: 1-925-671-6590
E-mail: service@systron.com
World Wide Web: http://www.systron.com

EUROPEAN HEADQUARTERS

Systron Donner Inertial Division
Evegate Business Centre, Evegate Park Barn
Smeeth Ashford, Kent, England TN25 6SX
Tel: ++44 (0) 1303 812778
Fax: ++44 (0) 1303 812708
E-mail: systron@easynet.co.uk

A subsidiary of BEI TECHNOLOGIES, INC.

DS01-2 9/98