МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(национальный исследовательский университет)

ФИЗТЕХ-ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра математического моделирования и прикладной математики

Квалификационная работа на соискание степени магистра
по направлению 03.04.01 «Прикладные математика и физика»,
магистерская программа «Управление динамическими системами»

Определение углового движения по видеоизображению

Выполнил:

студент группы М03-835б

Хулио Сесар Молина Саки

____

Научный руководитель:

к.ф.-м.н., доцент

Ткачев Степан Сергеевич

____

Москва, 2020

Аннотация

Работа посвящена задаче оценки ориентации объекта и его угловой скорости с помощью обработки изображений. Для решения данной задачи рассмотрен подход, состоящий из двух частей: определение матрицы поворота через модель измерения, адаптированной для использования кватернионов и реализация расширенного фильтра Калмана для оценки угловой скорости.

Модель измерения - это функция обеспечивающая преобразование координат точки в пространстве в координаты этой точки на ПЗС-матрице камеры. Она зависит от внутренних неизменных параметров камеры, а также от кватерниона ориентации и расстояния между камерой и объектом. Для определения внутренних параметров камеры проводится калибровка, которая также позволяет получить кватернион и расстояние.

Результаты экспериментальных исследований показали, что при калибровке можно определить кватернион ориентации с хорошей точностью, при это точность оценки угловой скорости, полученной с помощью численного дифференцирования, оказалась неудовлетворительной. Для повышения точности измерения угловой скорости был использован расширенный фильтр Калмана.

Contents

Introduction

The measurement of angular movement is of great importance because it allows to know and predict the orientation of the bodies with respect to a reference system, such information is vital for missions where maneuvers and interactions of two bodies or more are performed.

Researches on attitude and angular velocity estimation of objects have been performed with the help of different sensors such as photelectric encoders, tachometers, inertial sensors, and even laser. However, their implementation can be expensive. The use of digital images as low-cost sources of information for evaluating the angular motion is actively used in the field of robotics, control system, augmented reality, and are also widely used in the field of satellite systems.

Over the last four decades, a variety researches have been done on measuring motion parameters of objects using cameras, where a considerable importance had the develop of methods for camera calibration, which consist in the determination of internal parameters of the camera. In [4-7] calibration methods with analytical solutions are presented, where in addition to determine internal parameters of the camera, the 3-D object attitude in space are determined as part of the calibration process. Tsai [6] used Euler angles, while Zhang [5] used Rodrigues' rotation formula.

Researches have been proposed to investigate the measurement of object pose estimation. M. Dhome [8] proposed method to find the analytical solutions to the problem of the determination of the 3-D object attitude in space from a single perspective image. H. Kim [9] proposed a simple and fast stereo matching algorithm for real-time robotic applications using 3D information of vertexes on the outline of an object in image plane. Z. Zhong [10] presents a feature point pairs based technique for object pose estimation and 3D structure recovery from a single view, where it is defined strategies for small rotational and large rotational motion, X. Zhang [11] present algorithms for recovering the camera pose and the 3D-to-2D line correspondences simultaneously.

Measurement angular velocity by image processing is furthermore studied. Zhang [14] by means blurred images processing, proposed the estimation of motion parameters by measuring and comparing global geometric properties. Shigang [13] proposed parameter measurement of rotation through analyzing the information of visual rotation motion blur based on a single blurred image. By using event cameras, which have independent pixels that respond asynchronously to brightness changes, G. Gallego and D.Scaramuzza [11] proposed algorithm to estimate the angular velocity of the camera by analyzing the spatio-temporal coordinates of the brightness change.

Several years ago, it is increased the interest in parameters movement estimation by image processing for space applications. A.A.Boguslavsky [20] presented a software package that by means of video signal received from the TV-camera, mounted on the spacecraft board, allows the automatic visual monitoring of a spacecraft "Progress" docking to International Space Station. D. Ivanov [21], proposed a satellite relative position and orientation determination algorithm by performing image processing of the Sunlit spacecraft. This algorithm was used to determine the relative movement of the ChibisM microsatellite developed by the IKI RAS.

Koptev M.D. [22] proposed a method for the translational and rotational motion determination of mock-ups suspended on an aerodynamic testbed. The algorithm was based on the detection of installed special marks on the model’s body to evaluate the location of the model’s center of mass, angular position and angular velocity in the coordinate system associated with the aerodynamic testbed.

The difference between the determination algorithm developed and the one described above is that it does not require the installation of an additional special objective or photodiodes on the satellite to shoot; it is enough to know the geometry of the object being shot. The algorithm does not require the transfer of any data from the satellite being taken, therefore a piece of space debris can act as the second device. Therefore, the algorithm is suitable for the tasks of removing space debris from orbit: the satellite companion flies towards the debris, determines its movement, captures it and takes it to dense layers of the atmosphere.

Most of researches on measuring angular motion parameters mentioned above are focused on parameter 3-D object attitude determination or measurement angular velocity, but not both, if 3-D object attitude determination and angular velocity are considered to be estimated at the same time, usually it is considered to add one more sensor in addition to camera.

While in [5] and [6] used Rodrigues' rotation formula and Euler angles respectively, The purpose of this thesis work is the estimation of the 3-D object attitude by using quaternions , and in addition the angular velocity estimation at the same time by mean of a conventional low-cost camera without any additional sensor. The section 1 elaborates the problem statement of this research, and it is explained the importance of measurement model determination, which is the mathematical model for the camera. In the section 2 it is explained step by step the mathematical expression for the measurement model and its gradient. In the section 3 is the linear measurement model which allows to determine some parameters of the measurement model by linear methods. Due to the fact that the measurement model h is nonlinear and depends on unknown parameters of the camera, a calibration process is needed to be performed; In the section 4 the algorithm for calibration is explained and applied for 3-D object attitude determination.

Because the estimation of the angular velocity from consecutives rotation matrix has low precision, in the section 5 is shown the modeling system and the application of the Extended Kalman Filter to improve 3-D object attitude accuracy and for angular velocity estimation. And Finally, in section 6 the results for 3-D object attitude determination angular velocity estimation are shown.

1 Problem statement

The problem of the angular motion determination by the image processing is considered. The source of measurements is the camera which captures the object's movement by taking photographs at a certain frequency (see Figure 1)**,** These pictures are processed to estimate the angular motion of the object using the following information:

- $X_i(x_i, y_i, z_i)$: coordinates of the object points relative to the body-fixed frame OXYZ
- $X_i'(x_i', y_i')$: coordinates of the same points X_i, which are visualized and located in the image coordinate system (image)

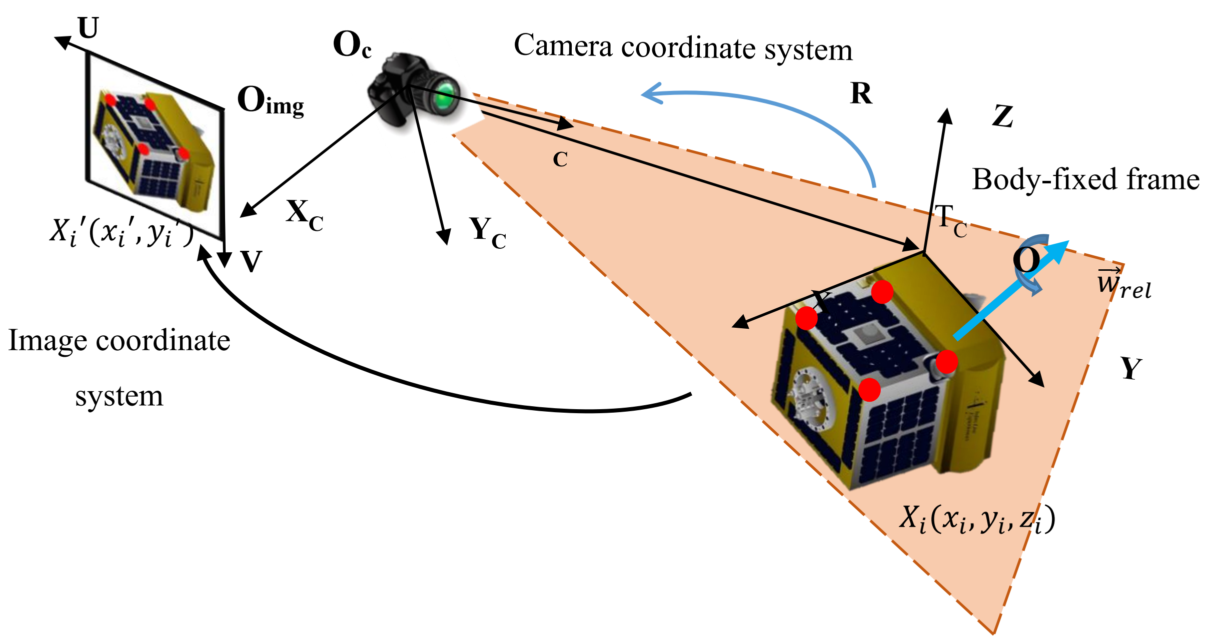


Figure 1. Diagram of the problem statement

The basic structure of a camera is shown in the Figure 2, where two main components are involved:

- Lens: Has the function of gathering and focus the light reflected from an object or scene. As the reflected light rays enter the camera lens, they are directed to the image sensor.
- Image sensor: it is a rectangular plane into where the points are projected, representing in that way the image of the object.

The image sensor is located parallel to the lens in the focal plane of the lens. The distance between the lens and the focal plane is called focal length f.

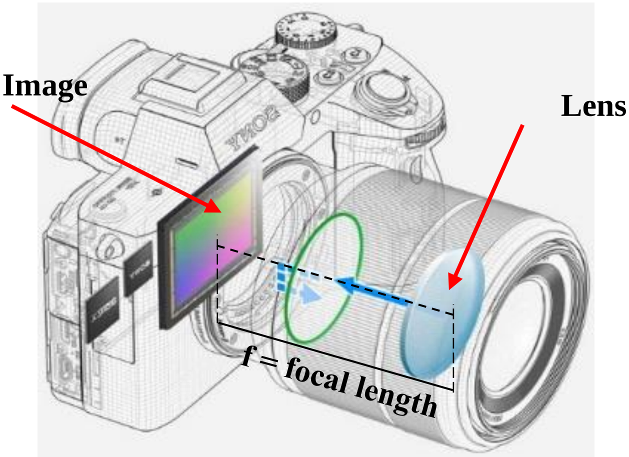


Figure 2. Basic structure of a camera

The locations of specific points of an object in a photograph varies according to the rotation matrix $\boldsymbol{R}$, and its translation vector $\boldsymbol{T_c}$ with respect to the camera. Thus, the estimation of the rotation matrix of a rigid body is possible when function $h(\boldsymbol{R}, \boldsymbol{T_c})$, called measurement model, that performs the projection of point X_i into the Image coordinate system is found. In addition, an average angular velocity can be calculated from two consecutive rotation matrices, thus, the first stage on this work is focused on the rotation matrix determination.

In the Figure 3 are shown the cartesian coordinates system used in this work:

- $OXYZ -$ Body-fixed frame (BF). This coordinate system is placed on the object.
- $O_cX_cY_cZ_c-$ Camera Coordinate System (CCS). It is based on the pinhole model, where its origin O_c is located at camera center (center of the lens), O_cZ_c is defined by the line from the camera center perpendicular to the image sensor, O_cX_c is parallel to the horizontal side of the image sensor, O_cY_c is parallel to the vertical side of the image sensor.
- $O_{img}UV$ – Image Coordinate System (ICS), also known as Image plane, is located in the plane defined by $Z_c = f$, Its origin O_{img} location depends on

the size of the image, $O_{img}U$ is parallel to the axis O_cX_c, $O_{img}V$ is parallel to the axis O_cY_c. $P = (u_0, v_0)$ is the principal point, which is located with respect to ICS. The principal point is formed by the intersection point between the O_cZ_c-axis and the sensor plane.

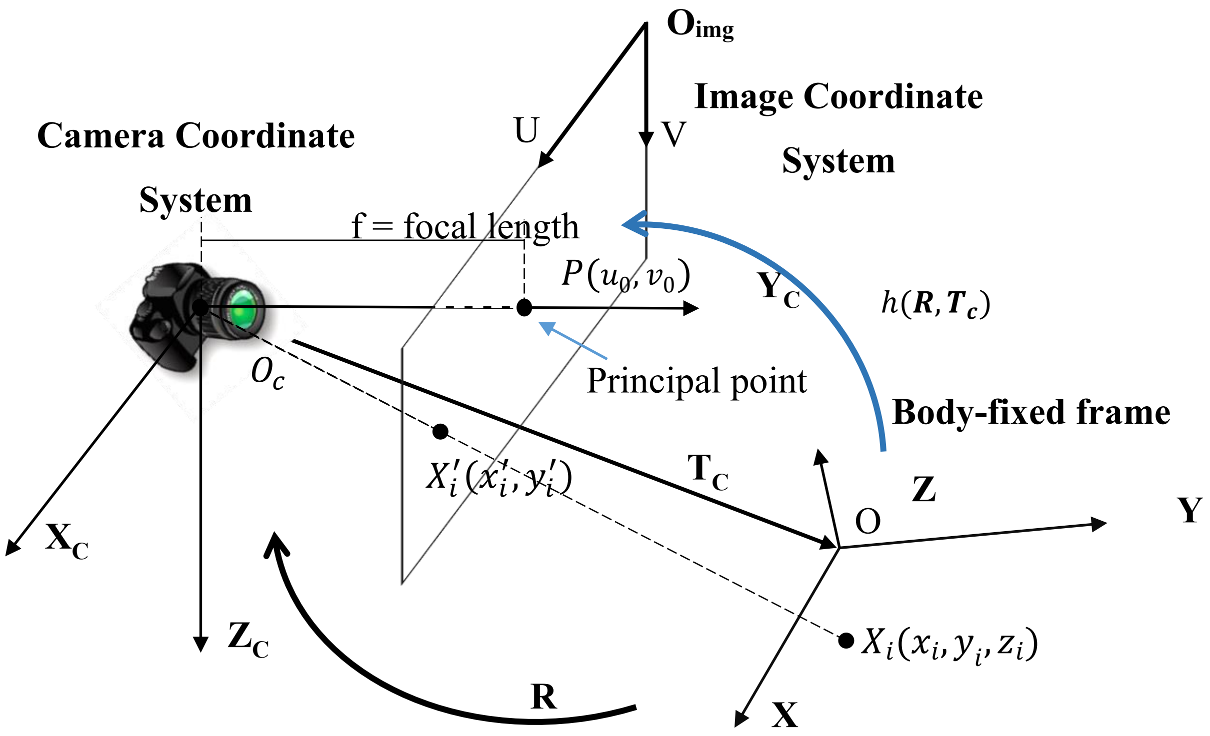


Figure 3. Pinhole camera model

The following notation of points in different coordinate systems is used:

- $X_i(x_i, y_i, z_i)$ – i-th point with respect to BF.
- $X_{c_i}(x_{c_i}, y_{c_i}, z_{c_i})$ – i-th point with respect to CCS.
- $X_i'(x_i', y_i') -$ i-th point with respect to ICS.

With regard to X_i , it is important to notice this point remain fixed with respect to the BF.

In order to implement the above explained, the next are considered:

- In section 2 the measurement model $h(\boldsymbol{R}, \boldsymbol{T_c})$ is defined. It shows the interconnection between parameters $\boldsymbol{R}$ and $\boldsymbol{T_c}$, that must be defined and points X_i' and X_i coordinates which can be measured.

- In section 3 the linearization of the measurement model is performed. This is required in order to obtain initial values for $\boldsymbol{R}$ and $\boldsymbol{T_c}$, which will be used during the calibration.
- In section 4 the calibration algorithm of the of the camera is presented.
- In section 5 the Kalman filter for angular velocity estimation is derived.
- Finally, in section 6 the results for 3-D object attitude determination angular velocity estimation are shown.

Most of the elements mentioned in this section are considered for measurement model $h(\boldsymbol{R}, \boldsymbol{T_c})$ determination because of its importance and relevance in the success of this work, for that reason is given in details the process to determine the measurement model.

2 Measurement model

In order to determine the measurement model h, which is a function that performs the projection of point X_i into the ICS from BF, it is required to consider the following:

- Transformation from BF to CCS
- Projection of the points from CCS into the sensor plane
- Distortion lens
- Transformation from sensor plane to ICS

which going to be explained in detail in this section.

2.1 Transformation from BF to CCS

Let $X_i = [x_i, y_i, z_i]^T$ be any point in the BF, where its transformation to the CCS is defined as follow:

$$X_{c_i} = \begin{bmatrix} x_{c_i} \\ y_{c_i} \\ z_{c_i} \end{bmatrix} = R \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \boldsymbol{T_c} \tag{2.1}$$

Where the rotation matrix R can be expressed as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{31} \end{bmatrix} \tag{2.2}$$

and T_c is translation vector with respect to CCS:

$$\boldsymbol{T_c} = [t_{xc} \quad t_{yc} \quad t_{zc}]^T \tag{2.3}$$

From equation (2.1), (2.2) and (2.3) the next expression

$$X_{c_i} = \begin{bmatrix} x_{c_i} \\ y_{c_i} \\ z_{c_i} \end{bmatrix} = \begin{bmatrix} r_{11}x_i + r_{12}y_i + r_{13}z_i + t_{xc} \\ r_{21}x_i + r_{22}y_i + r_{23}z_i + t_{yc} \\ r_{31}x_i + r_{32}y_i + r_{33}z_i + t_{zc} \end{bmatrix} \tag{2.4}$$

performs the transition from points from BF to the CCS.

2.2 Projection of the points from CCS into the image plane

The point X_{cp} represents the projection into the image plane of the points from CCS into the image plane, and it is expressed as

$$X_{cp_i} = \begin{bmatrix} x_{cp_i} \\ y_{cp_i} \end{bmatrix} = \begin{bmatrix} x_{c_i}/z_{c_i} \\ y_{c_i}/z_{c_i} \end{bmatrix} \quad (2.5)$$

Where

$$x_{cp_i} = \frac{r_{11}x_i + r_{12}y_i + r_{13}z_i + t_{xc}}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_{zc}} \quad (2.6)$$

$$y_{cp_i} = \frac{r_{21}x_i + r_{22}y_i + r_{23}z_i + t_{yc}}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_{zc}} \quad (2.7)$$

It is important to mention that X_{cp} is still located in the CCS.

2.3 Lens distortion

It is necessary to take into account that the image is distorted because of the lens distortion during the projection of the point X_c into the sensor plane. The usual types of distortion are radial distortion and tangential distortion. Radial distortion can be defined as a function which depends on the distance from the principal point (center image), and tangential distortion is caused by a not perfect parallel alignment between the lens and the image sensor [3]. These distortions can be defined by the following equation

$$X_{d_i} = \begin{bmatrix} x_{d_i} \\ y_{d_i} \end{bmatrix} = \begin{bmatrix} \underbrace{x_{cp_i}(1 + k_1 r_i^2 + k_2 r_i^4 + k_3 r_i^6)}_{\text{Radial distortion}} + \underbrace{2p_1 x_{cp_i} y_{cp_i} + p_2(r_i^2 + 2x_{cp_i}^2)}_{\text{Tangential}} \\ y_{cp_i}(1 + k_1 r_i^2 + k_2 r_i^4 + k_3 r_i^6) + 2p_2 x_{cp_i} y_{cp_i} + p_1(r_i^2 + 2y_{cp_i}^2) \end{bmatrix} \quad (2.8)$$

Where $r_i^2 = x_{cp_i}^2 + y_{cp_i}^2$ and k_1, k_2, k_3, p_1, p_2 are the distortion coefficients, X_d is the point coordinates when the lens distortion is taken into account. Incase when there is no lens distortion (ideal lens), X_{d_i} and X_{cp_i} are equal.

2.4 Transformation from sensor plane to ICS

Due to the fact that the ICS and sensor plane are parallel, and both located at the same plane, this transformation is based on scaling and translation of the points located in the sensor plane as follow:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h\,(f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2, R, T_c, X_i) = \begin{bmatrix} (x_{d_i} + s y_{d_i}) f_x + u_0 \\ y_{d_i} f_y + v_0 \end{bmatrix} \quad (2.9)$$

Where:

$P = (u_0, v_0)$: Principal point.

$f_x = \alpha_x f$: focal length axis-x (pixel).

$f_y = \alpha_y f$: focal length axis-y (pixel).

α_x, α_y: number of pixel per unit distance.

s: skew coefficient, which usually is equal to zero.

X_{p_i}: mapped point in the ICS from the BF.

X_i: Point in the BF.

The equation (2.9) rewritten as follow:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2, R, T_c) \qquad (2.10)$$

where h_{X_i} is called measurement model, which performs the projection of a point X_i into the ICS from the BF. However, taking into account that the parameters $f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2$ are fixed values and specific for each camera, the equation (2.10) can be simplified to $h_{X_i}(\boldsymbol{R}, \boldsymbol{T_c})$ once those parameters are determined.

2.5 Measurement model based on Rodrigues' rotation formula

Due to the fact that the rotation matrix has 9 scalar elements, it is convenient to express the rotation matrix with less scalar elements, it can be done by mean of Rodrigues's rotation formula as follow:

$$R(\alpha, \boldsymbol{u}) = I_3 \cos\alpha + (1 - \cos\alpha)\boldsymbol{u}\boldsymbol{u}^T + [\boldsymbol{u}]_x \sin\alpha \qquad (2.11)$$

Where $\boldsymbol{u} = [u_1, u_2, u_3]^T$ is the axis-rotation, α is the angle rotation, and $[\boldsymbol{u}]_x$ is skew-symmetric matrix of the vector $\boldsymbol{u}$.

$$[\boldsymbol{u}]_x = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \qquad (2.12)$$

In addition, as in [2], let the vector $\boldsymbol{v} = [v_1, v_2, v_3]^T$ be defined as follow:

$$\boldsymbol{v} = \alpha\boldsymbol{u} \qquad (2.13)$$

where α and $\vec{u}$ are given as a function of $\vec{v}$:

$$\alpha(\boldsymbol{v}) = |\boldsymbol{v}| = \sqrt{(\boldsymbol{v},\boldsymbol{v})} \tag{2.14}$$

$$\boldsymbol{u}(\boldsymbol{v}) = \frac{\boldsymbol{v}}{|\boldsymbol{v}|} \tag{2.15}$$

After substitutions, the rotation matrix R can be written as a function of the vector $\boldsymbol{v}$.

$$R(\boldsymbol{v}) = R(\boldsymbol{u}(\boldsymbol{v}), \alpha(\boldsymbol{v})) = I_3 \cos\alpha + (1-\cos\alpha)\boldsymbol{u}\boldsymbol{u}^T + [\boldsymbol{u}]_x \sin\alpha \tag{2.16}$$

As it can be noticed from the previous equations, the rotation matrix, which originally depends on 9 scalar values, now can be expressed by using a vector of three scalar values, hence the equation (2.10) can be rewritten as follow:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2, \boldsymbol{v}, \boldsymbol{T_c}) \tag{2.17}$$

It is convenient for an optimization problem, because in that way the number of parameters that have to be determined is decreased.

2.6 Measurement model based on Quaternions

Taking into account basic quaternion theory, let the multiplication of quaternions $\boldsymbol{Q} = [q_0, q_1, q_2, q_3]^T$and $\boldsymbol{P} = [p_0, p_1, p_2, p_3]^T$ be defined as follow:

$$\boldsymbol{P} \circ \boldsymbol{Q} = \begin{bmatrix} p_0 \\ \boldsymbol{p} \end{bmatrix} \circ \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} p_0 q_0 - \boldsymbol{p}\boldsymbol{q} \\ p_0\boldsymbol{q} + q_0\boldsymbol{p} + \boldsymbol{p}\ x\ \boldsymbol{q} \end{bmatrix} \tag{2.18}$$

Where $\boldsymbol{q} = [q_1, q_2, q_3]^T$ and $\boldsymbol{p} = [p_1, p_2, p_3]^T$ are vector parts of the quaternions $\boldsymbol{Q}$ and $\boldsymbol{P}$ respectively. The previous equation also can be rewritten in a matrix form:

$$\boldsymbol{P} \circ \boldsymbol{Q} = \begin{bmatrix} p_0 \\ \boldsymbol{p} \end{bmatrix} \circ \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} p_0 & -\boldsymbol{p}^T \\ \boldsymbol{p} & p_0 I_3 + [\boldsymbol{p}]_x \end{bmatrix} \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} q_0 & -\boldsymbol{q}^T \\ \boldsymbol{q} & q_0 I_3 - [\boldsymbol{q}]_x \end{bmatrix} \begin{bmatrix} p_0 \\ \boldsymbol{p} \end{bmatrix} \tag{2.19}$$

$$\boldsymbol{P} \circ \boldsymbol{Q} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ \boldsymbol{p} \end{bmatrix} \tag{2.20}$$

The rotation of points by means of quaternions is defined as follow:

$$\begin{bmatrix} 0 \\ X_c \end{bmatrix} = \Lambda \circ \begin{bmatrix} 0 \\ X \end{bmatrix} \circ \widetilde{\Lambda} \tag{2.21}$$

Where $X = [x, y, z]^T$is a tri dimensional point, and $\Lambda = [\lambda_0, \lambda_1, \lambda_2, \lambda_3]^T$ is an unit quaternion , which modulus $|\Lambda|$ is defined as follow:

$$|\Lambda| = \sqrt{{\lambda_0}^2 + {\lambda_1}^2 + {\lambda_2}^2 + {\lambda_3}^2} = 1 \tag{2.22}$$

And the conjugated of Λ is represented by $\widetilde{\Lambda}$.

$$\widetilde{\Lambda} = \begin{bmatrix} \lambda_0 \\ -\boldsymbol{\lambda} \end{bmatrix} \tag{2.23}$$

From the equation (2.21)

$$\begin{bmatrix} 0 \\ X_c \end{bmatrix} = \begin{bmatrix} \lambda_0 \\ \boldsymbol{\lambda} \end{bmatrix} \circ \begin{bmatrix} 0 \\ X \end{bmatrix} \circ \begin{bmatrix} \lambda_0 \\ -\boldsymbol{\lambda} \end{bmatrix} \tag{2.24}$$

$$\begin{bmatrix} 0 \\ X_c \end{bmatrix} = \begin{bmatrix} \lambda_0 & -\boldsymbol{\lambda}^T \\ \boldsymbol{\lambda} & \lambda_0 I_3 + [\boldsymbol{\lambda}]_x \end{bmatrix} \begin{bmatrix} \lambda_0 & -(-\boldsymbol{\lambda})^T \\ -\boldsymbol{\lambda} & \lambda_0 I_3 - [-\boldsymbol{\lambda}]_x \end{bmatrix} \begin{bmatrix} 0 \\ X \end{bmatrix} \tag{2.25}$$

After mathematics, it is obtained in the next matrix form:

$$\begin{bmatrix} 0 \\ X_c \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \boldsymbol{R}(\Lambda) \end{bmatrix} \begin{bmatrix} 0 \\ X \end{bmatrix} \tag{2.26}$$

Where $\boldsymbol{R}(\Lambda)$ represents the rotation matrix as a function of the quaternion Λ.

$$\boldsymbol{R}(\Lambda) = \begin{bmatrix} {\lambda_0}^2 + {\lambda_1}^2 - {\lambda_2}^2 - {\lambda_3}^2 & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & {\lambda_0}^2 - {\lambda_1}^2 + {\lambda_2}^2 - {\lambda_3}^2 & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & {\lambda_0}^2 - {\lambda_1}^2 - {\lambda_2}^2 + {\lambda_3}^2 \end{bmatrix} \tag{2.27}$$

As it has been shown in the equation (2.27), the rotation matrix can be expressed as a function of a quaternion Λ, which in general is composed of four elements; however, due to the fact that Λ is a unit quaternion, λ_0 in itself depends on their three elements.

$$\lambda_0(\lambda_1, \lambda_2, \lambda_3) = \lambda_0(\boldsymbol{\lambda}) = \sqrt{1 - ({\lambda_1}^2 + {\lambda_2}^2 + {\lambda_3}^2)} \tag{2.28}$$

The rotation matrix $\boldsymbol{R}(\Lambda)$, can be expressed as a function of the vector part $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^T$ of the next quaternion

$$R(\boldsymbol{\lambda}) = \begin{bmatrix} 1 - 2({\lambda_2}^2 + {\lambda_3}^2) & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & 1 - 2({\lambda_1}^2 + {\lambda_3}^2) & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & 1 - 2({\lambda_1}^2 + {\lambda_2}^2) \end{bmatrix} \tag{2.29}$$

Thus, the equation (2.10) can be rewritten as follow:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2, \boldsymbol{\lambda}, \boldsymbol{T_c}) \tag{2.30}$$

So the measurement model in quaternion form is obtained.

2.7 Measurement model gradient

Due to the fact that measurement model based on Rodrigues' rotation formula in the equation (2.17) is a composed function of several transformations, which involves vector and matrices, it is convenient to determine the gradient by means of matrix calculus. Let the measurement model be rewritten as follow:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{v}, \boldsymbol{T_c}) \tag{2.31}$$

Where $\boldsymbol{F} = [f_x \quad f_y]$, $\boldsymbol{C} = [u_0 \quad v_0]$, and $\boldsymbol{K_D}$=$[k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]$, and let the measurement model gradient be defined as follow:

$$Dh_{X_i} = \begin{bmatrix} \frac{\partial h_{X_i}}{\partial \boldsymbol{F}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{C}} & \frac{\partial h_{X_i}}{\partial S} & \frac{\partial h_{X_i}}{\partial \boldsymbol{K_D}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{v}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{T_c}} \end{bmatrix} \tag{2.32}$$

Where $\partial h_{X_i}/\partial \boldsymbol{F}$, $\partial h_{X_i}/\partial \boldsymbol{C}$, $\partial h_{X_i}/\partial S$, and $\partial h_{X_i}/\partial \boldsymbol{K_D}$ are defined in a matrix form:

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{F}} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial \boldsymbol{F}} \\ \frac{\partial y_{p_i}}{\partial \boldsymbol{F}} \end{bmatrix} = \begin{bmatrix} x_{d_i} + s y_{d_i} & 0 \\ 0 & y_{d_i} \end{bmatrix} \tag{2.33}$$

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{C}} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial \boldsymbol{C}} \\ \frac{\partial y_{p_i}}{\partial \boldsymbol{C}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.34}$$

$$\frac{\partial h_{X_i}}{\partial s} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial s} \\ \frac{\partial y_{p_i}}{\partial s} \end{bmatrix} = \begin{bmatrix} y_{p_i} f_x \\ 0 \end{bmatrix} \tag{2.35}$$

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{K_D}} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial \boldsymbol{K_D}} \\ \frac{\partial y_{p_i}}{\partial \boldsymbol{K_D}} \end{bmatrix} =$$

$$\begin{bmatrix} x_{cp_i}({x_{cp_i}}^2 + {y_{cp_i}}^2) & x_{cp_i}\left({x_{cp_i}}^2 + {y_{cp_i}}^2\right)^2 & 2x_{d_i}y_{d_i} & {y_{cp_i}}^2 + 3{x_{cp_i}}^2 & x_{cp_i}\left({x_{cp_i}}^2 + {y_{cp_i}}^2\right)^3 \\ y_{cp_i}({x_{cp_i}}^2 + {y_{cp_i}}^2) & y_{cp_i}\left({x_{cp_i}}^2 + {y_{cp_i}}^2\right)^2 & {x_{cp_i}}^2 + 3{y_{cp_i}}^2 & 2x_{d_i}y_{d_i} & y_{cp_i}\left({x_{cp_i}}^2 + {y_{cp_i}}^2\right)^3 \end{bmatrix} \tag{2.36}$$

With regard to the derivatives $\partial h_{X_i}/\partial \boldsymbol{v}$ and $\partial h_{X_i}/\partial \boldsymbol{T_c}$,they have remarkably complicated expressions due to the fact that the measurement model in the equation (2.17) is a composed function of several transformations.

As the rotation matrix $\boldsymbol{R}$ is a function of $\boldsymbol{v}$, see equation (2.16), the derivative $\partial h_{X_i}/\partial \boldsymbol{v}$ is

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{v}} = \frac{\partial h_{X_i}}{\partial \boldsymbol{R}}\frac{\partial \boldsymbol{R}(\boldsymbol{v})}{\partial \boldsymbol{v}} \tag{2.37}$$

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{R}} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial \boldsymbol{R}} \\ \frac{\partial y_{p_i}}{\partial \boldsymbol{R}} \end{bmatrix} \tag{2.38}$$

Where $\partial x_{p_i}/\partial \boldsymbol{R}$ and $\partial x_{p_i}/\partial \boldsymbol{R}$ are

$$\frac{\partial x_{p_i}}{\partial \boldsymbol{R}} = f_x\left(D_{r_i}\frac{\partial x_{cp_i}}{\partial \boldsymbol{R}} + x_{cp_i}\left[k1\frac{\partial(r_i{}^2)}{\partial \boldsymbol{R}} + k2\frac{\partial(r_i{}^4)}{\partial \boldsymbol{R}} + k3\frac{\partial(r_i{}^6)}{\partial \boldsymbol{R}}\right] + \left(2p_1 y_{cp_i} + 6p_2 x_{cp_i}\right)\frac{\partial x_{cp_i}}{\partial \boldsymbol{R}} + \left(2p_1 x_{cp_i} + 2p_2 y_{cp_i}\right)\frac{\partial y_{cp_i}}{\partial \boldsymbol{R}}\right) + sf_x\left(D_{r_i}\frac{\partial y_{cp_i}}{\partial \boldsymbol{R}} + y_{cp_i}\left[k1\frac{\partial(r_i{}^2)}{\partial \boldsymbol{R}} + k2\frac{\partial(r_i{}^4)}{\partial \boldsymbol{R}} + k3\frac{\partial(r_i{}^6)}{\partial \boldsymbol{R}}\right] + \left(2p_2 y_{cp_i} + 2p_1 x_{cp_i}\right)\frac{\partial x_{cp_i}}{\partial \boldsymbol{R}} + \left(2p_2 x_{cp_i} + 6p_1 y_{cp_i}\right)\frac{\partial y_{cp_i}}{\partial \boldsymbol{R}}\right) \tag{2.39}$$

$$\frac{\partial y_{p_i}}{\partial \boldsymbol{R}} = f_y\left(D_{r_i}\frac{\partial y_{cp_i}}{\partial \boldsymbol{R}} + y_{cp_i}\left[k1\frac{\partial(r_i{}^2)}{\partial \boldsymbol{R}} + k2\frac{\partial(r_i{}^4)}{\partial \boldsymbol{R}} + k3\frac{\partial(r_i{}^6)}{\partial \boldsymbol{R}}\right] + \left(2p_2 y_{cp_i} + 2p_1 x_{cp_i}\right)\frac{\partial x_{cp_i}}{\partial \boldsymbol{R}} + \left(2p_2 x_{cp_i} + 6p_1 y_{cp_i}\right)\frac{\partial y_{cp_i}}{\partial \boldsymbol{R}}\right) \tag{2.40}$$

$$D_{r_i} = 1 + k_1 r_i{}^2 + k_2 r_i{}^4 + k_3 r_i{}^6 \tag{2.41}$$

$$\frac{\partial(r_i{}^2)}{\partial \boldsymbol{R}} = 2\begin{bmatrix} x_{cp_i} & y_{cp_i} \end{bmatrix}\frac{\partial X_{cp_i}}{\partial \boldsymbol{R}} \tag{2.42}$$

$$\frac{\partial(r_i{}^4)}{\partial \boldsymbol{R}} = 2r_i{}^2\frac{\partial(r_i{}^2)}{\partial \boldsymbol{R}} = 4\left(x_{cp_i}{}^2 + y_{cp_i}{}^2\right)\begin{bmatrix} x_{cp_i} & y_{cp_i} \end{bmatrix}\frac{\partial X_{cp_i}}{\partial \boldsymbol{R}} \tag{2.43}$$

$$\frac{\partial(r_i{}^6)}{\partial \boldsymbol{R}} = 3r_i{}^4\frac{\partial(r_i{}^2)}{\partial \boldsymbol{R}} = 6\left(x_{cp_i}{}^2 + y_{cp_i}{}^2\right)^2\begin{bmatrix} x_{cp_i} & y_{cp_i} \end{bmatrix}\frac{\partial X_{cp_i}}{\partial \boldsymbol{R}} \tag{2.44}$$

In order to calculate $\partial \boldsymbol{R}/\partial \boldsymbol{v}$, some changes of variable are performed taking into account the equations (2.14) and (2.15) as follow:

$$X_1(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{u}(\boldsymbol{v}) \\ \alpha(\boldsymbol{v}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \\ \alpha \end{bmatrix} \tag{2.45}$$

Let $X_2(X_1) = [\beta, \gamma, \varphi, A, B]^T$, $\beta = \cos\alpha$, $\gamma = (1 - \cos\alpha)$, $\varphi = \sin\alpha$, $A = \boldsymbol{u}\boldsymbol{u}^T$, $B = [\boldsymbol{u}]_x$.

Then, the $\boldsymbol{R}(\boldsymbol{v})$, from the equation (2.16), can be rewritten as

$$\boldsymbol{R}(X_2) = I_3\beta + \gamma A + B\varphi \tag{2.46}$$

The derivative of $\boldsymbol{R}$ with respect to $\vec{v}$ can be defined as

$$\frac{\partial R}{\partial v} = \frac{\partial R}{\partial X_2}\frac{\partial X_2}{\partial X_1}\frac{\partial X_1}{\partial v} \tag{2.47}$$

$$\frac{\partial X_1}{\partial v} = \begin{bmatrix} \frac{1}{\alpha} - \frac{{v_1}^2}{\alpha^3} & -\frac{v_1 v_2}{\alpha^3} & -\frac{v_1 v_3}{\alpha^3} \\ -\frac{v_1 v_2}{\alpha^3} & \frac{1}{\alpha} - \frac{{v_2}^2}{\alpha^3} & -\frac{v_2 v_3}{\alpha^3} \\ -\frac{v_1 v_3}{\alpha^3} & -\frac{v_2 v_3}{\alpha^3} & \frac{1}{\alpha} - \frac{{v_3}^2}{\alpha^3} \\ \frac{v_1}{\alpha} & \frac{v_2}{\alpha} & \frac{v_3}{\alpha} \end{bmatrix} \tag{2.48}$$

$$\frac{\partial X_2}{\partial X_1} = \begin{bmatrix} 0 & 0 & 0 & -\sin\alpha \\ 0 & 0 & 0 & \cos\alpha \\ 0 & 0 & 0 & \sin\alpha \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2u_1 & 0 & 0 & 0 \\ u_2 & u_1 & 0 & 0 \\ u_3 & 0 & u_1 & 0 \\ u_2 & u_1 & 0 & 0 \\ 0 & 2u_2 & 0 & 0 \\ 0 & u_3 & u_2 & 0 \\ u_3 & 0 & u_1 & 0 \\ 0 & u_3 & u_2 & 0 \\ 0 & 0 & 2u_3 & 0 \end{bmatrix} \tag{2.49}$$

$$\frac{\partial R_k}{\partial X_2} = \left[\begin{array}{ccccccccccccccccccccc} 1 & A_{11} & B_{11} & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{21} & B_{21} & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{31} & B_{31} & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{12} & B_{12} & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi & 0 & 0 & 0 & 0 & 0 \\ 1 & A_{22} & B_{22} & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi & 0 & 0 & 0 & 0 \\ 0 & A_{32} & B_{32} & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi & 0 & 0 & 0 \\ 0 & A_{13} & B_{13} & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi & 0 & 0 \\ 0 & A_{23} & B_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi & 0 \\ 1 & A_{33} & B_{33} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi \end{array}\right] \tag{2.50}$$

If α is small, the equation (2.16) can be expressed as follow

$$\boldsymbol{R} = I_3 + [\boldsymbol{u}]_x \alpha \tag{2.51}$$

Then derivative of $\boldsymbol{R}$ with respect to $\boldsymbol{v}$ can be defined as

$$\frac{\partial R}{\partial v} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{2.52}$$

The derivative $\partial h_{X_i}/\partial \boldsymbol{T_c}$ is

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{T_c}} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial \boldsymbol{T_c}} \\ \frac{\partial y_{p_i}}{\partial \boldsymbol{T_c}} \end{bmatrix} \tag{2.53}$$

$$\frac{\partial x_{p_i}}{\partial \boldsymbol{T_c}} = f_x\left(D_{r_i}\frac{\partial x_{cp_i}}{\partial \boldsymbol{T_c}} + x_{cp_i}\left[k1\frac{\partial(r_i^2)}{\partial \boldsymbol{T_c}} + k2\frac{\partial(r_i^4)}{\partial \boldsymbol{T_c}} + k3\frac{\partial(r_i^6)}{\partial \boldsymbol{T_c}}\right] + \left(2p_1 y_{cp_i} + 6p_2 x_{cp_i}\right)\frac{\partial x_{cp_i}}{\partial \boldsymbol{T_c}} + \left(2p_1 x_{cp_i} + 2p_2 y_{cp_i}\right)\frac{\partial y_{cp_i}}{\partial \boldsymbol{T_c}}\right) + sf_x\left(D_{r_i}\frac{\partial y_{cp_i}}{\partial \boldsymbol{T_c}} + y_{cp_i}\left[k1\frac{\partial(r_i^2)}{\partial \boldsymbol{T_c}} + k2\frac{\partial(r_i^4)}{\partial \boldsymbol{T_c}} + k3\frac{\partial(r_i^6)}{\partial \boldsymbol{T_c}}\right] + \left(2p_2 y_{cp_i} + 2p_1 x_{cp_i}\right)\frac{\partial x_{cp_i}}{\partial \boldsymbol{T_c}} + \left(2p_2 x_{cp_i} + 6p_1 y_{cp_i}\right)\frac{\partial y_{cp_i}}{\partial \boldsymbol{T_c}}\right) \tag{2.54}$$

$$\frac{\partial y_{p_i}}{\partial \boldsymbol{T_c}} = f_y\left(D_{r_i}\frac{\partial y_{cp_i}}{\partial \boldsymbol{T_c}} + y_{cp_i}\left[k1\frac{\partial(r_i^2)}{\partial \boldsymbol{T_c}} + k2\frac{\partial(r_i^4)}{\partial \boldsymbol{T_c}} + k3\frac{\partial(r_i^6)}{\partial \boldsymbol{T_c}}\right] + \left(2p_2 y_{cp_i} + 2p_1 x_{cp_i}\right)\frac{\partial x_{cp_i}}{\partial \boldsymbol{T_c}} + \left(2p_2 x_{cp_i} + 6p_1 y_{cp_i}\right)\frac{\partial y_{cp_i}}{\partial \boldsymbol{T_c}}\right) \tag{2.55}$$

$$\frac{\partial(r_i^2)}{\partial \boldsymbol{T_c}} = 2\begin{bmatrix} x_{cp_i} & y_{cp_i} \end{bmatrix}\frac{\partial X_{cp_i}}{\partial \boldsymbol{T_c}} \tag{2.56}$$

$$\frac{\partial(r_i^4)}{\partial \boldsymbol{T_c}} = 2r_i^2\frac{\partial(r_i^2)}{\partial \boldsymbol{T_c}} = 4\left({x_{cp_i}}^2 + {y_{cp_i}}^2\right)\begin{bmatrix} x_{cp_i} & y_{cp_i} \end{bmatrix}\frac{\partial X_{cp_i}}{\partial \boldsymbol{T_c}} \tag{2.57}$$

$$\frac{\partial(r_i^6)}{\partial \boldsymbol{T_c}} = 3r_i^4\frac{\partial(r_i^2)}{\partial \boldsymbol{T_c}} = 6\left({x_{cp_i}}^2 + {y_{cp_i}}^2\right)^2\begin{bmatrix} x_{cp_i} & y_{cp_i} \end{bmatrix}\frac{\partial X_{cp_i}}{\partial \boldsymbol{T_c}} \tag{2.58}$$

$$\frac{\partial X_{cp_i}}{\partial \boldsymbol{T_c}} = \begin{bmatrix} \frac{\partial x_{cp_i}}{\partial \boldsymbol{T_c}} \\ \frac{\partial y_{cp_i}}{\partial \boldsymbol{T_c}} \end{bmatrix} = \begin{bmatrix} \frac{1}{z_{c_i}} & 0 & -\frac{x_{c_i}}{{z_{c_i}}^2} \\ 0 & \frac{1}{z_{c_i}} & -\frac{y_{c_i}}{{z_{c_i}}^2} \end{bmatrix} \tag{2.59}$$

As it can be seen, using the Rodrigues's rotation formula, the Jacobian matrix $\partial\boldsymbol{R}/\partial\boldsymbol{v}$ have remarkably complicated expression. On the other hand, if quaternions are used to represent the orientation matrix, the Jacobian matrix will yield a more convenient expression.

From the equation (2.30), let the measurement model based on quaternion be rewritten as follow:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\lambda}, \boldsymbol{T_c}) \tag{2.60}$$

And its measurement model gradient is be defined as follow:

$$Dh_{X_i} = \begin{bmatrix} \frac{\partial h_{X_i}}{\partial \boldsymbol{F}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{C}} & \frac{\partial h_{X_i}}{\partial S} & \frac{\partial h_{X_i}}{\partial \boldsymbol{K_D}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{\lambda}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{T_c}} \end{bmatrix} \tag{2.61}$$

The components $\partial h_{X_i}/\partial \boldsymbol{F}$, $\partial h_{X_i}/\partial \boldsymbol{C}$, $\partial h_{X_i}/\partial S$, $\partial h_{X_i}/\partial \boldsymbol{K_D}$, and $\partial h_{X_i}/\partial \boldsymbol{T_c}$ have been already determined in previous equations. As the rotation matrix $\boldsymbol{R}$ can be expressed as a function of $\boldsymbol{\lambda}$, see the equation (2.29), the derivative $\partial h_{X_i}/\partial \boldsymbol{\lambda}$ is

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{\lambda}} = \frac{\partial h_{X_i}}{\partial \boldsymbol{R}} \frac{\partial \boldsymbol{R}(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} \tag{2.62}$$

The expression for $\partial h_{X_i}/\partial \boldsymbol{R}$ has been determined in the equation (2.38), the expression for $\partial \boldsymbol{R}/\partial \boldsymbol{\lambda}$ is

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\lambda}} = \begin{bmatrix} 0 & -4\lambda_2 & -4\lambda_3 \\ 2\lambda_2 - 2\lambda_3\lambda_1/\lambda_0 & 2\lambda_1 - 2\lambda_3\lambda_2/\lambda_0 & 2\lambda_0 - 2\lambda_3\lambda_3/\lambda_0 \\ 2\lambda_3 + 2\lambda_2\lambda_1/\lambda_0 & -2\lambda_0 + \lambda_2\lambda_2/\lambda_0 & 2\lambda_1 + 2\lambda_2\lambda_3/\lambda_0 \\ 2\lambda_2 + 2\lambda_3\lambda_1/\lambda_0 & 2\lambda_1 + 2\lambda_3\lambda_2/\lambda_0 & -2\lambda_0 + 2\lambda_3\lambda_3/\lambda_0 \\ -4\lambda_1 & 0 & -4\lambda_3 \\ 2\lambda_0 - 2\lambda_1\lambda_1/\lambda_0 & 2\lambda_3 - 2\lambda_1\lambda_2/\lambda_0 & 2\lambda_2 - 2\lambda_1\lambda_3/\lambda_0 \\ 2\lambda_3 - 2\lambda_2\lambda_1/\lambda_0 & 2\lambda_0 - 2\lambda_2\lambda_2/\lambda_0 & 2\lambda_1 - 2\lambda_2\lambda_3/\lambda_0 \\ -2\lambda_0 + 2\lambda_1\lambda_1/\lambda_0 & 2\lambda_3 + 2\lambda_1\lambda_2/\lambda_0 & 2\lambda_2 + 2\lambda_1\lambda_3/\lambda_0 \\ -4\lambda_1 & -4\lambda_2 & 0 \end{bmatrix} \tag{2.63}$$

As it can be noticed, the expression for $\partial \boldsymbol{R}/\partial \boldsymbol{\lambda}$ is simpler than the expression for $\partial R/\partial \boldsymbol{v}$, this is convenient for time computing.

3 Linearization of the measurement model

The linearization of the measurement model allows to solve the nonlinear calibration problem for cameras by linear method, in which the nonlinear radial and tangential distortion components are ignored.

The equations (2.6) and (2.7) are nonlinear functions, which perform the projection to the sensor plane, and can be linearized by means of Homogeneous coordinates provided that the vector X_{c_i} and X_{cp_i} are expressed homogeneous vectors [1], obtaining the equation (3.1) as a linear expression, where the symbol $\sim$ in the equation (3.1) means that the two homogeneous vectors are not equal, but they have the same direction.

$$\begin{bmatrix} x_{cp_i} \\ y_{cp_i} \\ z_{cp_i} \end{bmatrix} \sim \begin{bmatrix} x_{c_i} f \\ y_{c_i} f \\ z_{c_i} \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{c_i} \\ y_{c_i} \\ z_{c_i} \\ 1 \end{bmatrix} \tag{3.1}$$

With regard to the effect of the lens distortion, it is convenient to consider it to be equal to zero during the linearization process [2]. Therefore, considering this particular case it is possible to obtain a linear expression, see equation (3.2), from the nonlinear measurement model (2.9) by means of the homogeneous coordinates which is usually done in order to determine initial values of internal and external parameter.

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \sim \begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T_c} \\ 0_{1x3} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{3.2}$$

Where $\tilde{u}, \tilde{v}, \tilde{w}$ are homogeneous coordinates for the points in the ICS.

The equation (3.2) can be expressed as follows:

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \sim [H_{3x4}] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{3.3}$$

where matrix H is the transition matrix, or linear measurement model. The BF is chosen in such a way that the points X_i are located on the XY-plane, in consequence, the component z_i is zero, it means that the equation (3.3) can be reduced to equation (3.4).

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ \widetilde{w}_i \end{bmatrix} \sim \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{3.4}$$

Due to the fact that the vectors $[\tilde{u}_i, \tilde{v}_i, \widetilde{w}_i]^T$ and $H[x_i, y_i, 1]^T$ have the same direction, their cross product is zero and based on the Direct Linear Transformation (DLT) algorithm [1] the equation is

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ \widetilde{w}_i \end{bmatrix} \text{ x } H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.5}$$

$$\begin{bmatrix} 0^T & -\widetilde{w}_i X_i^T & \tilde{v}_i X_i^T \\ \widetilde{w}_i X_i^T & 0^T & -\tilde{u}_i X_i^T \end{bmatrix} L = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.6}$$

Where $L = [a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_8\ a_9]^T$ and $X_i = [x_i \quad y_i \quad 1]^T$.

As it can be seen, the equation (3.6) has the form of a homogeneous system, where L can be determined by the Single Values Decomposition (SVD). This DLT algorithm is widely used to calculate the transition matrix H where is needed a set of four points as minimum. However, because matrix H is a projective transformation, it has a non-linear nature, therefore, an iterative method can be performed in order to optimize the components of the matrix H by means of reduction of the error projection [2]. Thus, it is necessary to work in inhomogeneous coordinates.

Let the matrix H already determined by means of DLT, then

$$\begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \\ \widetilde{\omega}_i \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{3.7}$$

Where $(\tilde{x}_i, \tilde{y}_i, \widetilde{\omega}_i)$ is the homogeneous coordinate representation of a point (u_i, v_i) located in the ICS, then the projective transformation in the equation (3.7) can be written in inhomogeneous form as

$$u_i = \frac{\tilde{x}_i}{\tilde{\omega}_i} = \frac{a_1 x_i + a_2 y_i + a_3}{a_7 x_i + a_8 y_i + a_9} \tag{3.8}$$

$$v_i = \frac{\tilde{y}_i}{\tilde{\omega}_i} = \frac{a_4 x_i + a_5 y_i + a_6}{a_7 x_i + a_8 y_i + a_9} \tag{3.9}$$

Where (u_i, v_i) finally represents the mapped point in the ICS from the BF. The Jacobian matrix for projective transformation is shown below, which is widely used by the most of the iterative methods.

$$J = \frac{\partial \begin{bmatrix} u_i \\ v_i \end{bmatrix}}{\partial L} = \frac{1}{\tilde{w}_i} \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -u_i x_i & -u_i y_i & -u_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -v_i x_i & -v_i y_i & -v_i \end{bmatrix} \tag{3.10}$$

Before determining the transition matrix it is recommended to perform a normalization of the data to avoid bad results because of noisy data. In [1] is recommended a normalization data so that the centroid of the new set of points is the origin of coordinates (0,0) and the average distance from the origin equals to $\sqrt{2}$, as it is shown in the Figure 4.

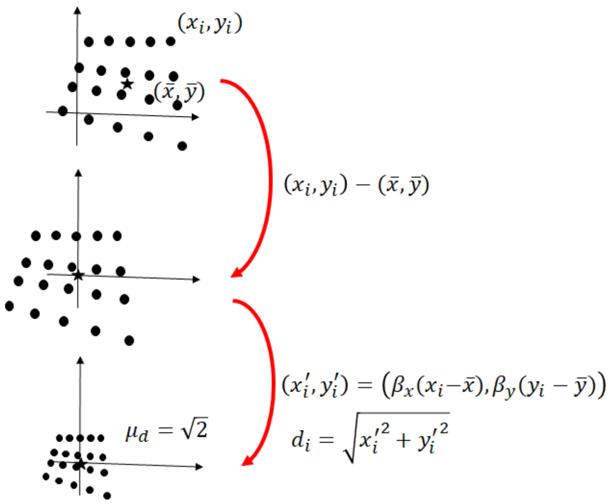


Figure 4 Preconditioning for image coordinate system points

This preconditioning can be expressed as a matrix H_{prec} as below:

$$H_{prec} = \begin{bmatrix} \beta_x & 0 & -\beta_x \bar{x} \\ 0 & \beta_y & -\beta_y \bar{y} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

Where $\bar{x}, \bar{y}$ are means of the location of the points in the image, and β_x and β_y are given in [2] as:

$$\beta_x = \frac{1}{\frac{1}{N}\sum_{i=1}^{N}|x_i - \bar{x}|} \tag{3.12}$$

$$\beta_y = \frac{1}{\frac{1}{N}\sum_{i=1}^{N}|y_i - \bar{y}|} \tag{3.13}$$

The explained above is summarized in the Algorithm 1

I. **Initialize data:**

- Let $i = 1,2,\dots,n$, where $n \geq 4$ is the number of mapped points.
- Let $X_i = [x_i \quad y_i \quad 1]^T$ be a homogeneous coordinate representation of a i-th point from the BF, where the component z_i is zero.
- Let $(\tilde{u}_i, \tilde{v}_i, \tilde{w}_i)$ be a homogeneous coordinate representation of a i-th point located in the ICS.
- Let $\tilde{w}_i$ to be one, in order to make $(\tilde{u}_i, \tilde{v}_i)$ points measured in the ICS.
  - ✓ Apply the preconditioning matrix to each point as follow:

$$\begin{bmatrix} \tilde{u}_i' \\ \tilde{v}_i' \\ \tilde{w}_i' \end{bmatrix} = H_{prec} \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ \tilde{w}_i \end{bmatrix}$$

- Write the homogeneous system according to the equation (3.6) for n points:

$$\begin{bmatrix} 0^T & -\tilde{w}_1' X_1^T & \tilde{v}_1' X_1^T \\ \tilde{w}_1' X_1^T & 0^T & -\tilde{u}_1' X_1^T \\ & \vdots & \\ 0^T & -\tilde{w}_n' X_n^T & \tilde{v}_n' X_n^T \\ \tilde{w}_n' X_n^T & 0^T & -\tilde{u}_n' X_n^T \end{bmatrix} L = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

- Solve the homogenous system in order to obtain L, and obtain the transition matrix H from L.
- Update the transition matrix as follow:

$$H \leftarrow H/a_9$$

Algorithm 1. Computing Transition matrix H

In the next section is shown how the Computing transition matrix algorithm is not only used to linearized the measurement model, it also can be used to estimate the rotation matrix and translation vector of the BF with respect to CCS by a linear method.

4 Calibration Algorithm

In this section the calibration algorithm of the widely known tool for camera calibration developed by Bouguet, J. Y. "Camera Calibration Toolbox for Matlab" [2] is described in details.

In the Figure 5, it is shown that a chessboard is photographed with different orientations and translation vectors in order to obtain considerable amount of points for calibration process. Additionally, intrinsic parameters are shown, which are internal fixed parameters of the camera itself. They have to be determined in the calibration process and then will remain fixed. On the other hand, extrinsic parameters, rotation matrix and translation vector, are determined for each image, and they are not fixed parameters because the location and orientation of the object can change.

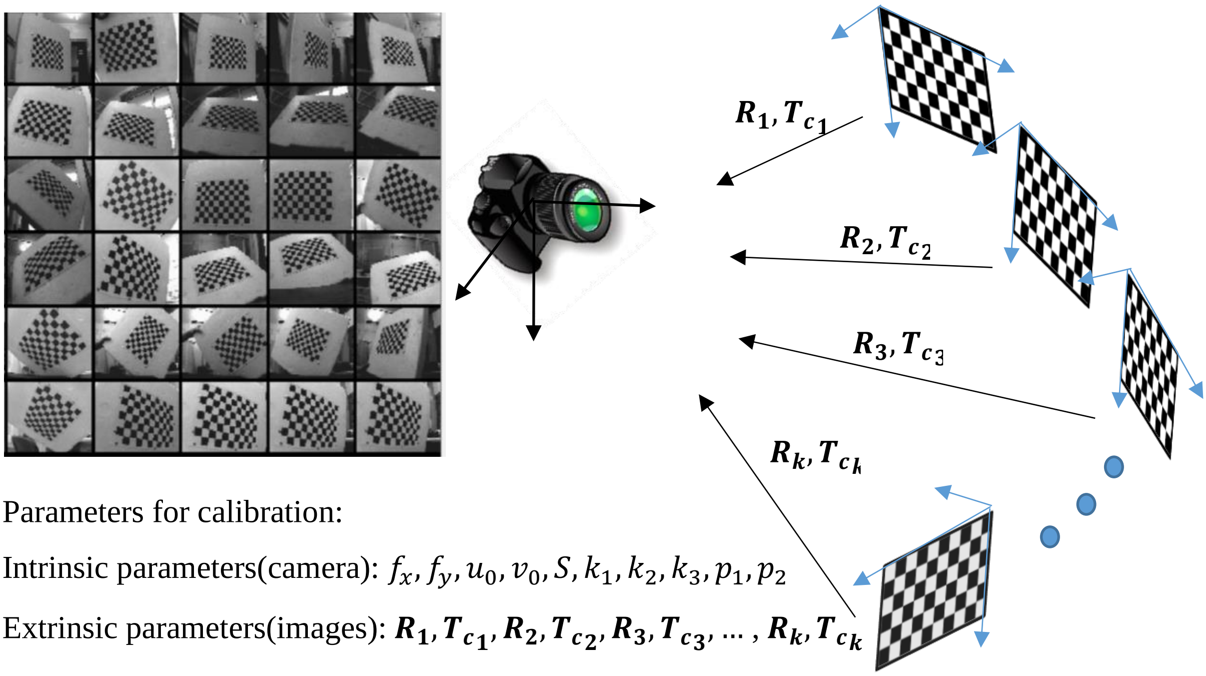


Figure 5 Intrinsic and Extrinsic parameters

Calibration Process is based on two main steps: Initialization of the parameters and optimization of the parameters by the gradient method.

4.1 Initialization of the parameters

The initial value of the principal point can be initialized as the center point of the image, for example, if the resolution of the camera is 640x480 pixels, then the principal point $\boldsymbol{P} = (u_0, v_0) = (320{,}240)$. The Skew parameter can be initialized as zero as well as the distortions coefficients k_1, k_2, k_3, p_1, p_2.

With regards to the focal distance (f_x, f_y), it can be initialized using vanishing points as in [1] and different methods as in [2] and [4], which make use of transition matrices from BF to the ICS by using the Algorithm 1.

Considering initial values for skew factor 'S' and distortion coefficients $\boldsymbol{K_D}$ equal to zero, the points in the ICS (x_i', y_i') can be transformed into CCS as follows:

$$x_{cp_i} = (x_i' - u_0)/f_x \tag{4.1}$$

$$y_{cp_i} = (y_i' - v_0)/f_y \tag{4.2}$$

The equations above show that the point (x_i', y_i') is located in the sensor plane in the CCS, and it is related to the BF by the next equation, where the points in the ICS and BF are expressed by homogeneous coordinate.

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \sim \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T_c} \\ \boldsymbol{0_{1x3}} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{4.3}$$

Because of component z_i is zero for flat objects, equation (4.3) can be rewritten as

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \sim [\boldsymbol{r_1} \quad \boldsymbol{r_2} \quad \boldsymbol{T_c}] \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = H_r \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{4.4}$$

where $\boldsymbol{T_c}$ is translation vector and $\boldsymbol{r_i}$ are the columns of rotation matrix **R**, and $\|\boldsymbol{r_i}\| = 1$. The matrix H_r can be computed by means of the Algorithm 1 , and additionally it is necessary to perform a normalization so that the vectors $\boldsymbol{r_i}$ have modulus equal to one, then to use the QR decomposition to obtain a better result in the orthogonality of the vector $\boldsymbol{r_i}$.

4.2 Optimization process

Due to the non-linearity of the measurement model, an optimization process is required to be performed in order to tune up the parameters, which have been initialized previously. The essential step is the definition of the equations system.

Let us considerer a scheme where it is available just one image, as it is shown in the Figure 6. Let $i = 1,2,\dots,n$, where n is the number of mapped points to the ICS. Let $X'_i = [x'_i, y'_i]^T$ and $X_i = [x_i, y_i, 0]^T$ be the vector representations of a point in the ICS and BF, X'_i and X_i are known values.

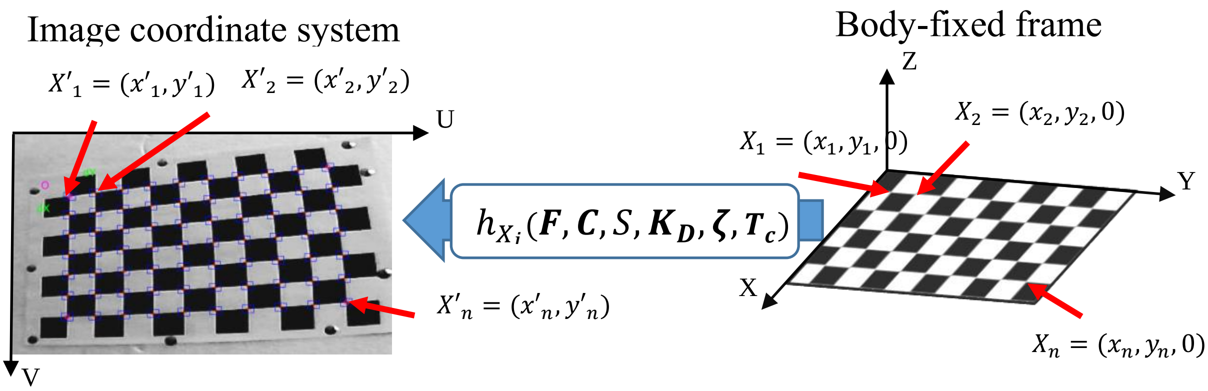


Figure 6. Projection from Body-fixed frame to image coordinate system

Let $Xp_i = [xp_i, yp_i]^T$ be i-th point already mapped to the ICS from the BF by using the measurement model h_{X_i} from the equation (2.31) or from the equation (2.60), where the rotation matrix can be expressed by using the vector $\boldsymbol{\zeta}$, below the system of equations for one image with n points.

$$\begin{aligned} Xp_1 &= h_{X_1}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta}, \boldsymbol{T_c}) \\ Xp_2 &= h_{X_2}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta}, \boldsymbol{T_c}) \\ &\vdots \\ Xp_i &= h_{X_i}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta}, \boldsymbol{T_c}) \\ &\vdots \\ Xp_n &= h_{X_n}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta}, \boldsymbol{T_c}) \end{aligned} \tag{4.5}$$

Let ΔX be the error vector, which is defined as the difference between the points Xp_i and X'_i as follow.

$$\Delta X = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} Xp_1 - X'_1 \\ Xp_2 - X'_2 \\ \vdots \\ Xp_n - X'_n \end{bmatrix} = \begin{bmatrix} xp_1 - x'_1 \\ yp_1 - y'_1 \\ xp_2 - x'_2 \\ yp_2 - y'_2 \\ \vdots \\ xp_n - x'_n \\ yp_n - y'_n \end{bmatrix} \tag{4.6}$$

Now let us consider that is available m images with n points in each image. Let $i = 1,2, \ldots, n$, and $k = 1,2, \ldots, m$; where n is the number of mapped points to the ICS, and m is the number images, it is important to mention that the location of points in each image depends on the translation vector and orientation of the BF with respect to the CCS.

Let $X'^k_i = [x'^k_i, y'^k_i]^T$ be the vector representation of the i-th point in the k-th image (ICS).

Let $X_i = [x_i, y_i, 0]^T$ be the vector representation of the i-th point in the BF.

Let $Xp^k_i = [xp^k_i, yp^k_i]^T$ be the point X_i already mapped to the k-th image (ICS) from the BF by using the nonlinear model $h^k_{X_i}$, which represent the projection of the point X_i to the k-th image. Below the system of equations for m images with n points in each image.

$$\begin{gathered} Xp^1_1 = h^1_{X_1}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_1}, \boldsymbol{T_{c_1}}) \\ Xp^1_2 = h^1_{X_2}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_1}, \boldsymbol{T_{c_1}}) \\ \vdots \\ Xp^1_n = h^1_{X_n}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_1}, \boldsymbol{T_{c_1}}) \\ Xp^2_1 = h^2_{X_1}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_2}, \boldsymbol{T_{c_2}}) \\ Xp^2_2 = h^2_{X_2}(\boldsymbol{F}, C, S, \boldsymbol{K_D}, \boldsymbol{\zeta_2}, \boldsymbol{T_{c_2}}) \\ \vdots \\ Xp^2_n = h^2_{X_n}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_2}, \boldsymbol{T_{c_2}}) \\ \vdots \\ Xp^k_i = h^k_{X_i}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_k}, \boldsymbol{T_{c_k}}) \\ \vdots \\ Xp^m_1 = h^m_{X_1}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_m}, \boldsymbol{T_{c_m}}) \\ Xp^m_2 = h^m_{X_2}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_m}, \boldsymbol{T_{c_m}}) \\ \vdots \\ Xp^m_n = h^m_{X_n}(\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_m}, \boldsymbol{T_{c_m}}) \end{gathered} \tag{4.7}$$

Let h^k be defined as

$$h^k(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_k},\boldsymbol{T_{c_k}}) = \begin{bmatrix} h^k_{X_1}(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_k},\boldsymbol{T_{c_k}}) \\ h^k_{X_2}(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_k},\boldsymbol{T_{c_k}}) \\ \vdots \\ h^k_{X_n}(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_k},\boldsymbol{T_{c_k}}) \end{bmatrix} \tag{4.8}$$

From the measurement model gradients in the equations (2.32) and (2.61), the partial derivatives of h^k are

$$\begin{bmatrix} \frac{\partial h^k}{\partial \boldsymbol{F}} & \frac{\partial h^k}{\partial \boldsymbol{C}} & \frac{\partial h^k}{\partial S} & \frac{\partial h^k}{\partial \boldsymbol{K_D}} & \frac{\partial h^k}{\partial \boldsymbol{\zeta_k}} & \frac{\partial h^k}{\partial \boldsymbol{T_{c_k}}} \end{bmatrix} = \begin{bmatrix} \frac{\partial h^k_{X_1}}{\partial \boldsymbol{F}} & \frac{\partial h^k_{X_1}}{\partial \boldsymbol{C}} & \frac{\partial h^k_{X_1}}{\partial S} & \frac{\partial h^k_{X_1}}{\partial \boldsymbol{K_D}} & \frac{\partial h^k_{X_1}}{\partial \boldsymbol{\zeta_k}} & \frac{\partial h^k_{X_1}}{\partial \boldsymbol{T_{c_k}}} \\ \frac{\partial h^k_{X_2}}{\partial \boldsymbol{F}} & \frac{\partial h^k_{X_2}}{\partial \boldsymbol{C}} & \frac{\partial h^k_{X_2}}{\partial S} & \frac{\partial h^k_{X_2}}{\partial \boldsymbol{K_D}} & \frac{\partial h^k_{X_2}}{\partial \boldsymbol{\zeta_k}} & \frac{\partial h^k_{X_2}}{\partial \boldsymbol{T_{c_k}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h^k_{n}}{\partial \boldsymbol{F}} & \frac{\partial h^k_{n}}{\partial \boldsymbol{C}} & \frac{\partial h^k_{n}}{\partial S} & \frac{\partial h^k_{n}}{\partial \boldsymbol{K_D}} & \frac{\partial h^k_{n}}{\partial \boldsymbol{\zeta_k}} & \frac{\partial h^k_{n}}{\partial \boldsymbol{T_{c_k}}} \end{bmatrix} \tag{4.9}$$

Then the system of equations (4.7) can be expressed in a shorter form as follow

$$\begin{gathered} Xp^1 = h^1(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_1},\boldsymbol{T_{c_1}}) \rightarrow \Delta X^1 = Xp^1 - X'^1 \\ Xp^2 = h^2(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_2},\boldsymbol{T_{c_2}}) \rightarrow \Delta X^2 = Xp^2 - X'^2 \\ \vdots \\ Xp^k = h^k(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_k},\boldsymbol{T_{c_k}}) \rightarrow \Delta X^k = Xp^k - X'^k \\ \vdots \\ Xp^m = h^m(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_m},\boldsymbol{T_{c_m}}) \rightarrow \Delta X^m = Xp^m - X'^m \end{gathered} \tag{4.10}$$

Where $Xp^k = [xp^k_1, yp^k_1, xp^k_2, yp^k_2, \dots, xp^k_n, yp^k_n]^T$ is column vector representation of the points mapped to the k-th image (ICS) from the BF, and $X'^k = [x'^k_1, y'^k_1, x'^k_2, y'^k_2, \dots, x'^k_n, y'^k_n]^T$ is vector representation of all the points in the k-th image.

In the equation (4.6), the error vector ΔX can be express as ΔX^k ,where k indicates the error vector for the corresponding k-th image and nonlinear model h^k.

Finally, the system of equation can be express as a column of functions.

$$\hbar(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{R_1},\boldsymbol{T_{c_1}},\dots,\boldsymbol{R_m},\boldsymbol{T_{c_m}}) = \begin{bmatrix} Xp^1 \\ Xp^2 \\ \vdots \\ Xp^m \end{bmatrix} = \begin{bmatrix} h^1(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_1},\boldsymbol{T_{c_1}}) \\ h^2(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_2},\boldsymbol{T_{c_2}}) \\ \vdots \\ h^m(\boldsymbol{F},\boldsymbol{C},S,\boldsymbol{K_D},\boldsymbol{\zeta_m},\boldsymbol{T_{c_m}}) \end{bmatrix} \tag{4.11}$$

Let X' be the column vector $\left[X'^{1^T}, X'^{2^T}, \dots, X'^{m^T}\right]^T$, and the global error vector can be defined as follow:

$$\boldsymbol{\epsilon} = \hbar(M) - X' = \begin{bmatrix} \Delta X^1 \\ \Delta X^2 \\ \vdots \\ \Delta X^m \end{bmatrix} \tag{4.12}$$

The Gauss–Newton Method is used to solve the optimization problem, which is based on the minimization of the global error vector $\boldsymbol{\epsilon}$; Let $\boldsymbol{M} = [\boldsymbol{F}, \boldsymbol{C}, S, \boldsymbol{K_D}, \boldsymbol{\zeta_1}, \boldsymbol{T_{c_1}}, \dots, \boldsymbol{\zeta_m}, \boldsymbol{T_{c_m}}]$ be the vector of parameters, let $\boldsymbol{M_0}$ be the initial values for the vector of parameters $\boldsymbol{M}$, and let $\boldsymbol{\epsilon_0}$ be the initial error vector.

$$\boldsymbol{\epsilon_0} = \hbar(\boldsymbol{M_0}) - X' \tag{4.13}$$

Let $\boldsymbol{\epsilon_l}$ the error vector and $\boldsymbol{M_l}$ be vector of parameters, which are updated for each iteration, as follow:

$$\boldsymbol{\epsilon_l} = \hbar(\boldsymbol{M_l}) \ - X' \tag{4.14}$$

$$\Delta \boldsymbol{M} = (\boldsymbol{J}^T \boldsymbol{J})^{-1} \boldsymbol{J}^T \boldsymbol{\epsilon_l} \tag{4.15}$$

$$\boldsymbol{M}_{l+1} = \boldsymbol{M}_l + \Delta \boldsymbol{M} \tag{4.16}$$

Where $\boldsymbol{J}$, the Jacobian matrix, is defined as follow:

$$\boldsymbol{J} = \frac{\partial \hbar}{\partial M} = \begin{bmatrix} \frac{\partial h^1}{\partial \boldsymbol{M}} \\ \frac{\partial h^2}{\partial \boldsymbol{M}} \\ \frac{\partial h^3}{\partial \boldsymbol{M}} \\ \vdots \\ \frac{\partial h^k}{\partial \boldsymbol{M}} \\ \vdots \\ \frac{\partial h^m}{\partial \boldsymbol{M}} \end{bmatrix} = \begin{bmatrix} \frac{\partial h^1}{\partial \boldsymbol{F}} & \frac{\partial h^1}{\partial \boldsymbol{C}} & \frac{\partial h^1}{\partial S} & \frac{\partial h^1}{\partial \boldsymbol{K_D}} & \frac{\partial h^1}{\partial \zeta_1} & \frac{\partial h^1}{\partial \boldsymbol{T_{c_1}}} & 0 & 0 & 0 & 0 & \dots & 0 \\ \frac{\partial h^2}{\partial \boldsymbol{F}} & \frac{\partial h^2}{\partial \boldsymbol{C}} & \frac{\partial h^2}{\partial S} & \frac{\partial h^2}{\partial \boldsymbol{K_D}} & 0 & 0 & \frac{\partial h^2}{\partial \zeta_2} & \frac{\partial h^2}{\partial \boldsymbol{T_{c_2}}} & 0 & 0 & \dots & 0 \\ \frac{\partial h^3}{\partial \boldsymbol{F}} & \frac{\partial h^3}{\partial \boldsymbol{C}} & \frac{\partial h^3}{\partial S} & \frac{\partial h^3}{\partial \boldsymbol{K_D}} & 0 & 0 & 0 & 0 & \frac{\partial h^3}{\partial \zeta_3} & \frac{\partial h^3}{\partial \boldsymbol{T_{c_3}}} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ \frac{\partial h^k}{\partial \boldsymbol{F}} & \frac{\partial h^k}{\partial \boldsymbol{C}} & \frac{\partial h^k}{\partial S} & \frac{\partial h^k}{\partial \boldsymbol{K_D}} & 0 & 0 & 0 & \dots & 0 & \frac{\partial h^k}{\partial \zeta_k} & \frac{\partial h^k}{\partial \boldsymbol{T_{c_k}}} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ \frac{\partial h^m}{\partial \boldsymbol{F}} & \frac{\partial h^m}{\partial \boldsymbol{C}} & \frac{\partial h^m}{\partial S} & \frac{\partial h^m}{\partial \boldsymbol{K_D}} & 0 & 0 & 0 & 0 & \dots & 0 & \frac{\partial h^m}{\partial \zeta_m} & \frac{\partial h^m}{\partial \boldsymbol{T_{c_m}}} \end{bmatrix} \tag{4.17}$$

Where $F = [f_x \quad f_y]$, $C = [u_0 \quad v_0]$, and K_D=$[k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]$.

4.3 Algorithm for camera calibration

Given m images with n points in each image, let $i = 1,2,\dots,n$, and $k = 1,2,\dots,m$; where n is the number of mapped points to the ICS, and m is the number images. The calibration algorithm is shown below.

I. **Initialize parameters:** Use the algorithm 1 to initialize the vector parameters $M = \left[F, C, S, K_D, \vec{v}_1, T_{c_1}, \dots, \vec{v}_m, T_{c_m}\right]$, if quaternions are used the vector parameters is $M = \left[F, C, S, K_D, \vec{\lambda}_1, T_{c_1}, \dots, \vec{\lambda}_m, T_{c_m}\right]$.

II. **Initialize global error vector $\boldsymbol{\epsilon_0}$:** $\epsilon_0 = \hbar(M_0) - X'$

III. **Iterative process:**

   a. $\Delta M_0 = ({J_0}^T J_0)^{-1} {J_0}^T \epsilon_0$, where J_0 is J jacobian matrix evaluated at M_0
   b. $M_1 = M_0 + \Delta M_0$
   c. Change ← $|\Delta M_0| / |M_1|$
   d. Iteration ← 0
   e. While ((Change > 1e-10) & (Iterarion < MaxIteration))
      i. $\epsilon_1 = \hbar(M_1) - X'$
      ii. $\Delta M_1 = ({J_1}^T J_1)^{-1} {J_1}^T \epsilon_1$
      iii. $M_2 = M_1 + \Delta M_1$
      iv. If Quaternions is used, the quaternion part of M_2 must be normalized for each iteration, and then M_2 must be updated.
      v. Change ← $|\Delta M_1| / |M_2|$
      vi. Iterarion ← Iterarion + 1

   The vector $\boldsymbol{M_2}$ is the optimal vector $\boldsymbol{M}$.

Algorithm 2. Camera calibration

After the calibration, the Algorithm 2 can be used to determine the matrix rotation and T_c without considering the other parameters in vector M.

4.4 Application algorithm for camera calibration

Using 50 images and 70 points per image. The images were taken using the camera Model FI8918W with resolution 480x640 pixels.

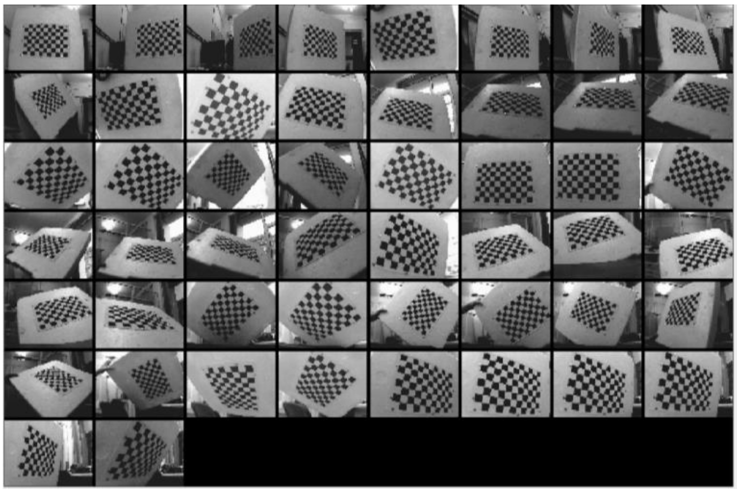


Figure 7 Image used for calibration

As the result of the calibration process using the Algorithm 2 the values of the intrinsic parameters are obtained:

- Principal point $P = (318.85122, 255.46648)$ (pixel)
- Focal length axis-x $f_x = 633.54607$(pixel)
- Focal length axis-y $f_y = 634.02213$(pixel)
- Skew $s = 0.0$
- Distortion coefficients $k_1 =$ -0.46378, $k_2 =$ 0.28011, k_3 =0.0, p_1 =0.00083, p_2 =0.00269
- The total error is expressed in pixels $\sigma_x = 0.20879, \sigma_y = 0.24828$

It is necessary to keep in mind that only intrinsic parameters remain fixed because they are fixed values which depend on the camera assembly. On the other hand, the external parameters, rotation matrix and translation vector change as the BF or the camera move.

In the Figure 8 the extrinsic parameters by mean of the locations and orientations of the chessboard with respect to the CCS are shown, which has been obtained during the calibration process.

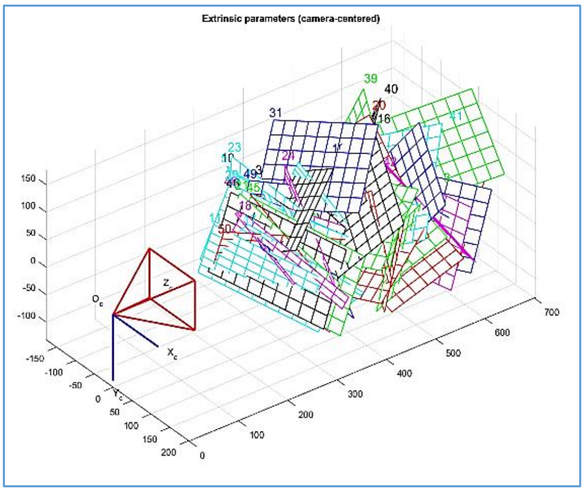


Figure 8 Visualization of the extrinsic parameters with a fixed camera

As a result of the calibration process, the equation (2.17) for the measurement model based on Rodrigues' rotation formula can be rewritten as follow:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_1}(\vec{v}, T_c) \tag{4.18}$$

And the equation (2.30) for the measurement model based on quaternions can be rewritten as follows:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(\vec{\lambda}, T_c) \tag{4.19}$$

Thus, if the points X_{p_i} and points X_i are known, it is possible to determine the orientation and the translation vector by means of Algorithm 2 as in the next section is shown.

4.5 Testing measurement model

In this part, the results of the camera calibration by means of a rotating table is presented. The facilities for testing are shown in the Figure 9. It is used to determine how accurate the measurement model is. The kinematic equation of the schema is analyzed for initial time t_0 and for the time t_1 when a rotation angle α around the axis-$Y_T$is performed.

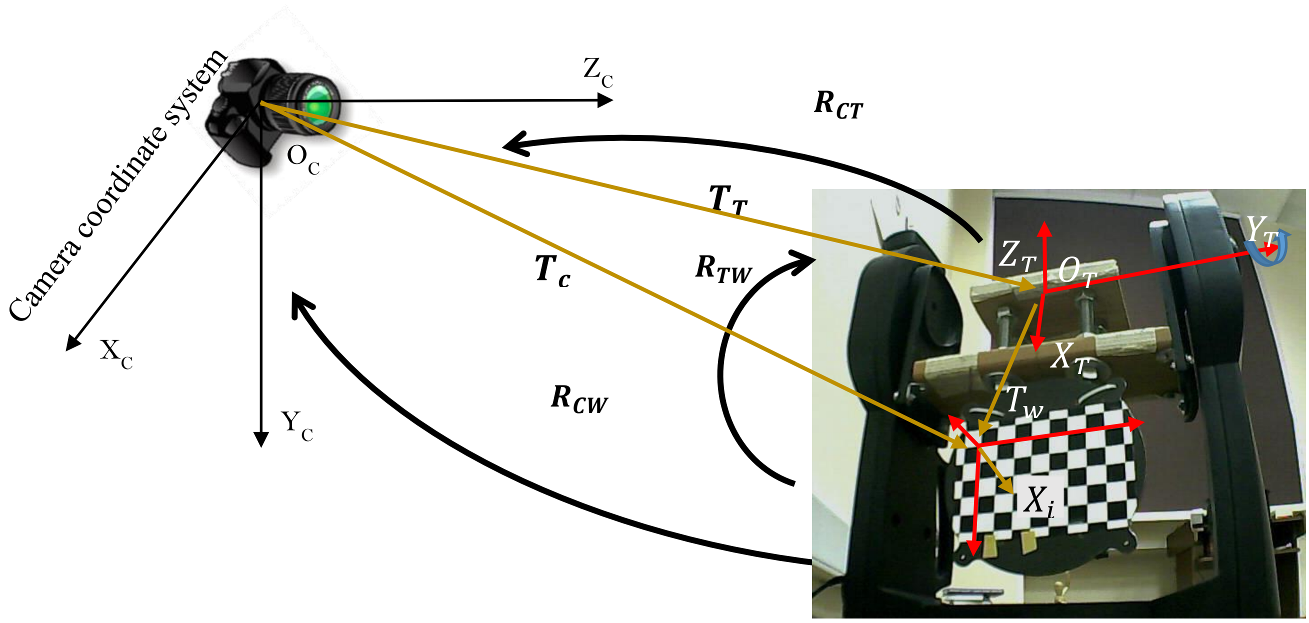


Figure 9 Testing schema using rotation table

Kinematic equation in t_0:

$$R_{CW_{t0}} X_i + T_{C_{t0}} = R_{CT}(R_{TW} X_i + T_W) + T_T \tag{4.20}$$

$$R_{CW_{t0}} X_i + T_{C_{t0}} = R_{CT} R_{TW} X_i + R_{CT} T_W + T_T \tag{4.21}$$

Kinematic equation in t_1:

$$R_{CW_{t1}} X_i + T_{C_{t1}} = R_{CT} R_{TW} R_\alpha X_i + R_{CT} T_W + T_T \tag{4.22}$$

where:

X_i: Points with respect to the coordinate system $O_w X_w Y_w Z_w$.

R_{TW}: Transformation matrix from the coordinate system $O_w X_w Y_w Z_w$ to the rotating table coordinate system $O_T X_T Y_T Z_T$.

R_{CT} : Transformation matrix from the rotating table coordinate system $O_T X_T Y_T Z_T$ to the CCS.

${R_{CW}}_{t0}, {R_{CW}}_{t1}$: Transformation matrix at time t_0 and t_1 from the coordinate system $O_w X_w Y_w Z_w$ to the CCS obtained by the Algorithm 2.

${T_C}_{t0}, {T_C}_{t1}$: Transformation vectors at time t_0 and t_1 with respect to the camera coordinate system obtained by the Algorithm 2.

T_W: Translation vectors with respect to the rotating table coordinate system $O_T X_T Y_T Z_T$.

T_T: Translation vectors with respect to the CCS.

From the equation (4.21) and (4.22) it is seen that:

$$ {R_{CW}}_{t0} = R_{CT} R_{TW} \tag{4.23} $$

$$ {R_{CW}}_{t1} = R_{CT} R_{TW} R_\alpha \tag{4.24} $$

From the previous equations it is possible to obtain a direct formula to estimate the rotation matrix R_α (intrinsic rotation) with respect to $O_w X_w Y_w Z_w$, see the next equation.

$$ R_\alpha = {R_{CW}}_{t0}^{-1} {R_{CW}}_{t1} \tag{4.25} $$

The equation (4.25) can be rewritten using quaternions:

$$ \Lambda_\alpha = {\Lambda_{CW}}_{t0}^{-1} \circ {\Lambda_{CW}}_{t1} \tag{4.26} $$

As it can be noticed in the previous equations (4.25), R_α depends on two consecutives rotations of coordinate system $O_w X_w Y_w Z_w$. The rotation matrix R_{CW}, which can be expressed as a function of $\boldsymbol{v}$, and Λ_{CW} can be obtained by mean of Algorithm 2 considering only extrinsic parameters, orientation and the translation vector, with regard to intrinsic parameters, they are not included in the parameters, because they are fixed values, and they already have been determined during the calibration.

A testing with the rotating table which consists of three rotations of 1°, 2° and 3° around axis-Y_T is performed where the measurement model accuracy is shown. In the Figure 10 it is noticed that the mean (μ) of consecutively rotations is very close to the true angle α with small standard deviation (σ).

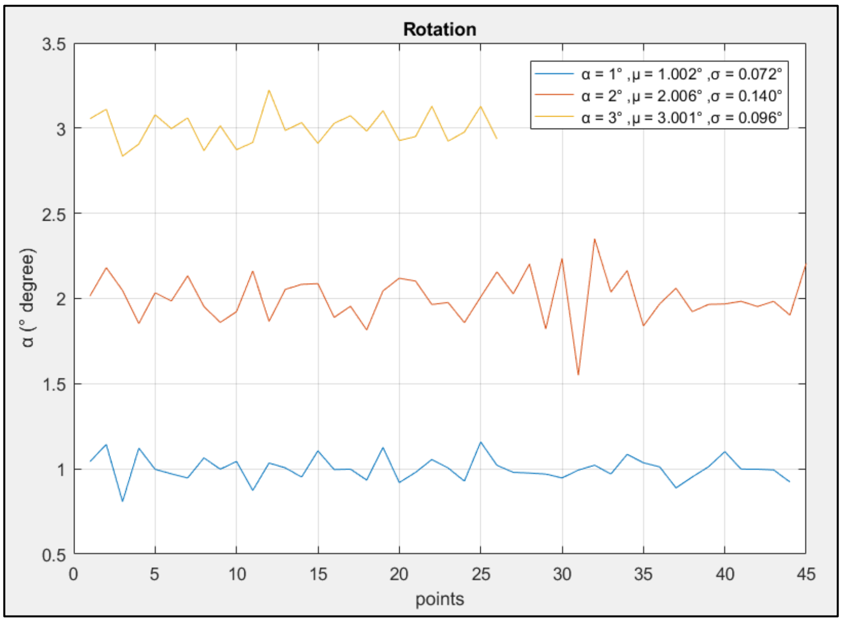


Figure 10 Detections for three rotations of 1°,2° and 3° using the intrinsic parameters

Another testing is performed in order to know if it is possible to detect very small rotation angles such as 1 arcmin (0.0167°), 5 arcmin (0.0833°) and 15 arcmin (0.25°) using low resolution camera.

As it can be seen in the Figure 11, the accuracy, defined as how close the mean value (μ) to the true angle value (α) is, it is less as the rotation angle is smaller. On the other hand, the precision (σ) is still maintained.

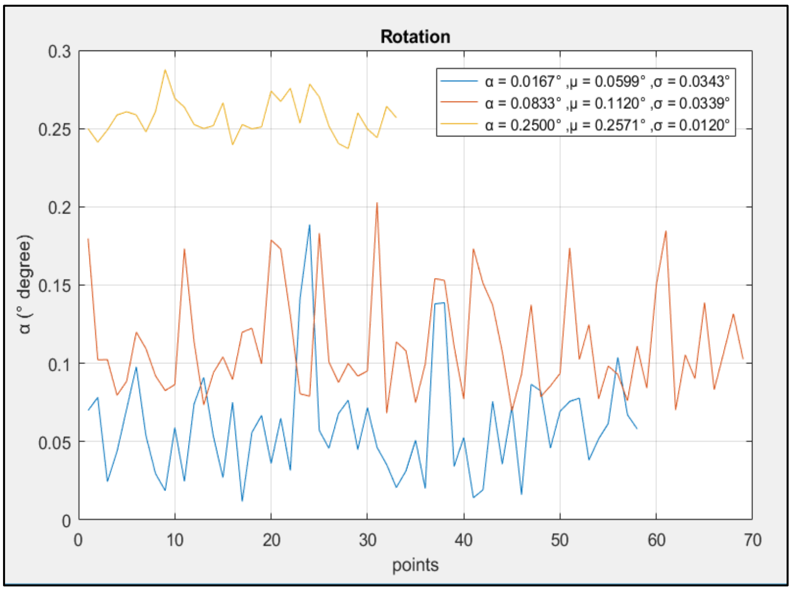


Figure 11 Detection of three rotations 1arcmin (0.0167°), 5arcmin (0.0833°) and 15arcmin (0.25°) using the intrinsic parameters

Until this point the testing has been performed using chessboard where a remarkable amount of points is provided. However, it is not possible to establish the correspondence between the point from the BF and the ICS automatically, this required the user support. It is very important that the program for image processing detects and localizes automatically and accurately the points of correspondence between the BF and the ICS, since the accuracy of the rotation matrix and translation vector depends on it.

It is shown in the Figure 12, that once the four points are detected and their position in the image is evaluated, it is impossible to determine which point is P1, P2, P3 or P4. Therefore, the correspondences are not possible to be determined.

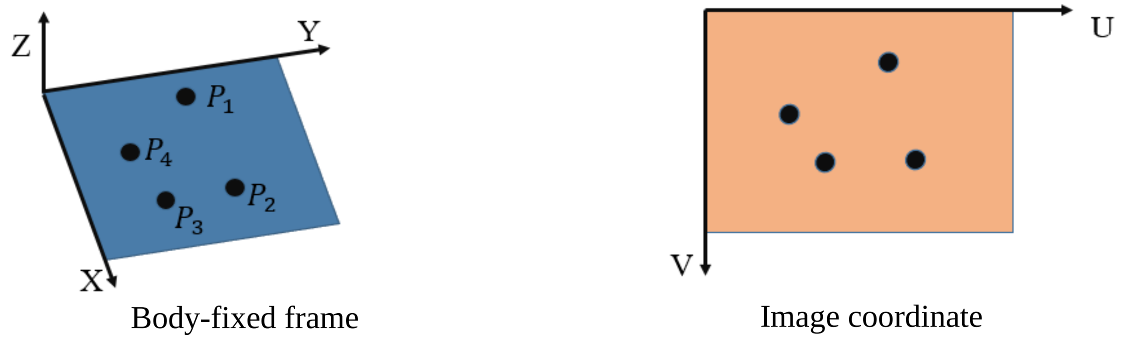


Figure 12 Example where correspondences are not possible determined

To solve this problem, a pattern between each point can be used in order to determine the correspondences. In order to do that the utilization of the Aruco pattern is considered [4], [5]. It helps to establish the correspondence between the point from the BF and the ICS as it can be seen in the next figure.

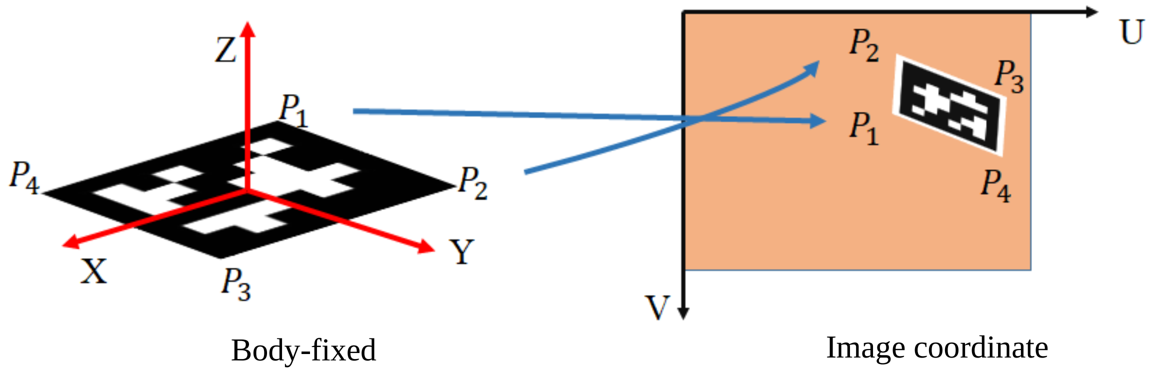


Figure 13 Correspondences determined by using Aruco patterns

In this experiment the correspondences are established automatically using the aruco library. As it is understood, the measurement model's error is inherent, and in addition to that error, another source of errors appears, such as: the error produced by change of brightness in the enviroment, by the digitization of the image, and by the algorithm for corner detection.

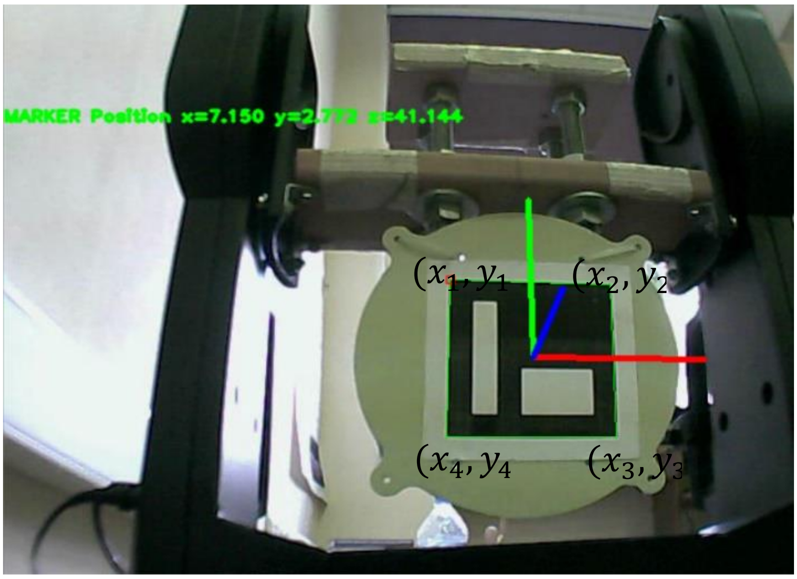


Figure 14 Aruco pattern and rotating table

In the Figure 14 the used aruco pattern is shown, and in the Figure 15 it is shown how the locations of the detected corners change for each image with the rotating table being static.

From the Figure 15 to the Figure 18, it can be seen that the located corners present in coordinates x(pixel) and y(pixel) maximum standard deviation 0.110 and 0.124 respectively.

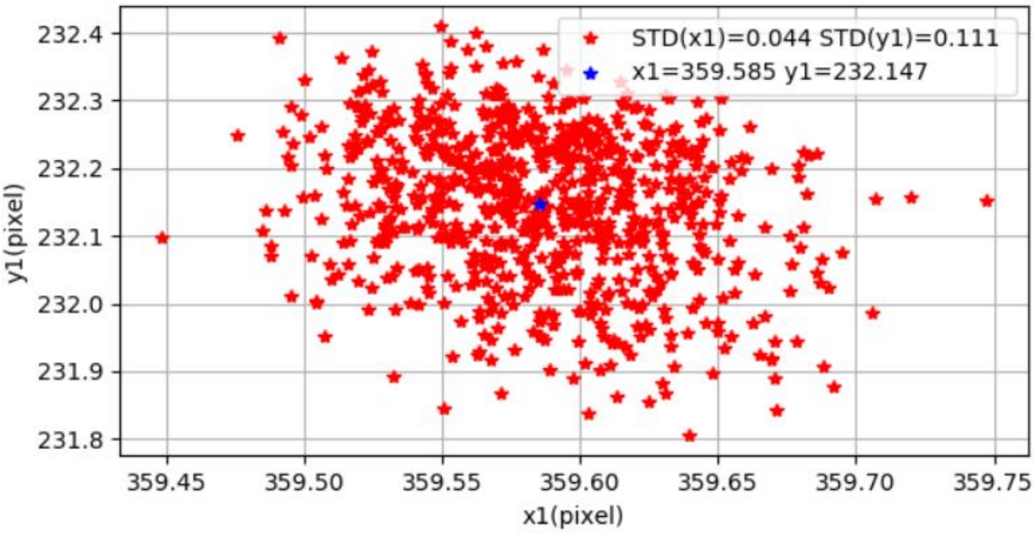


Figure 15 Mean and standard deviation (STD) of the point X1

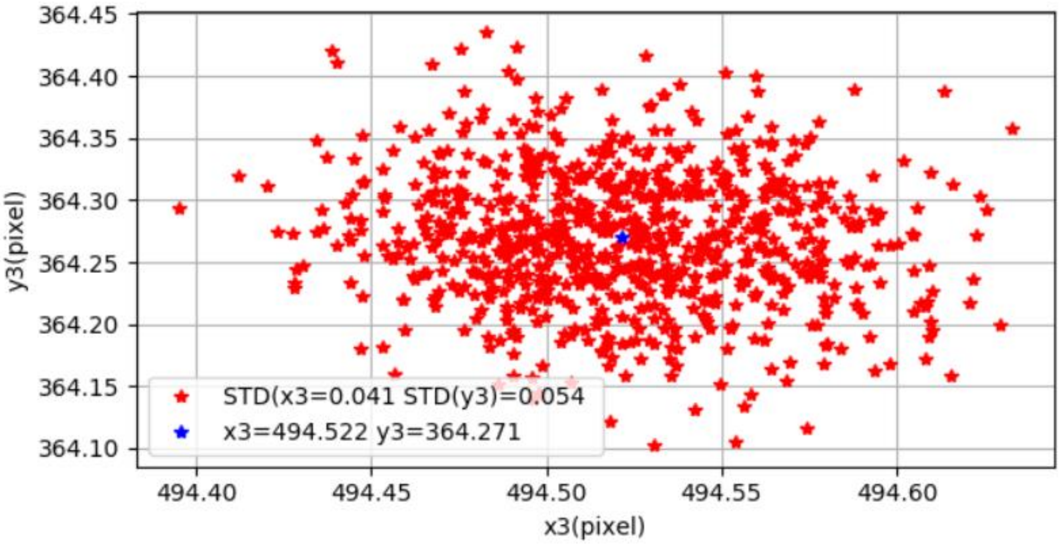


Figure 16 Mean and standard deviation (STD) of the point X3

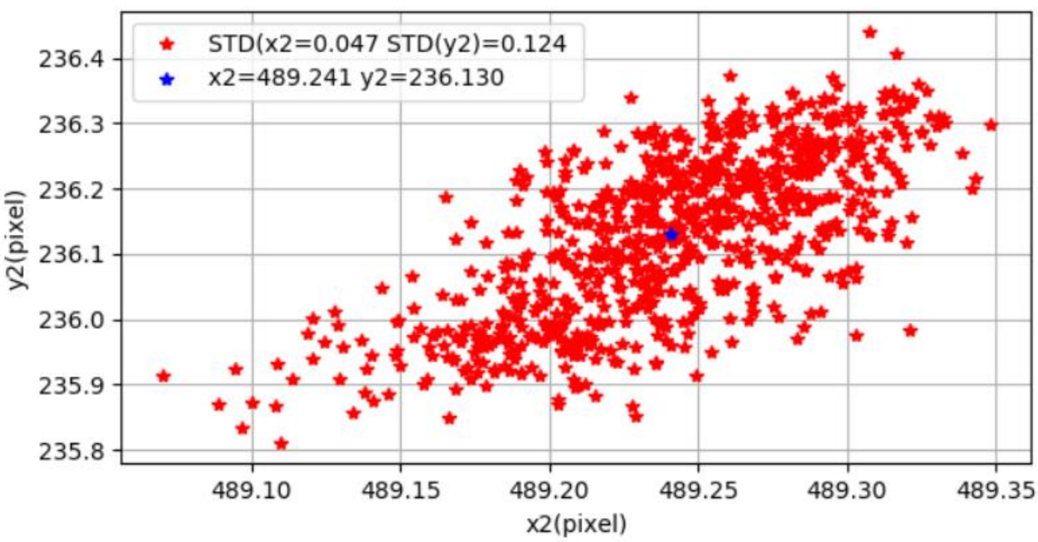


Figure 17 Mean and standard deviation (STD) of the point X2

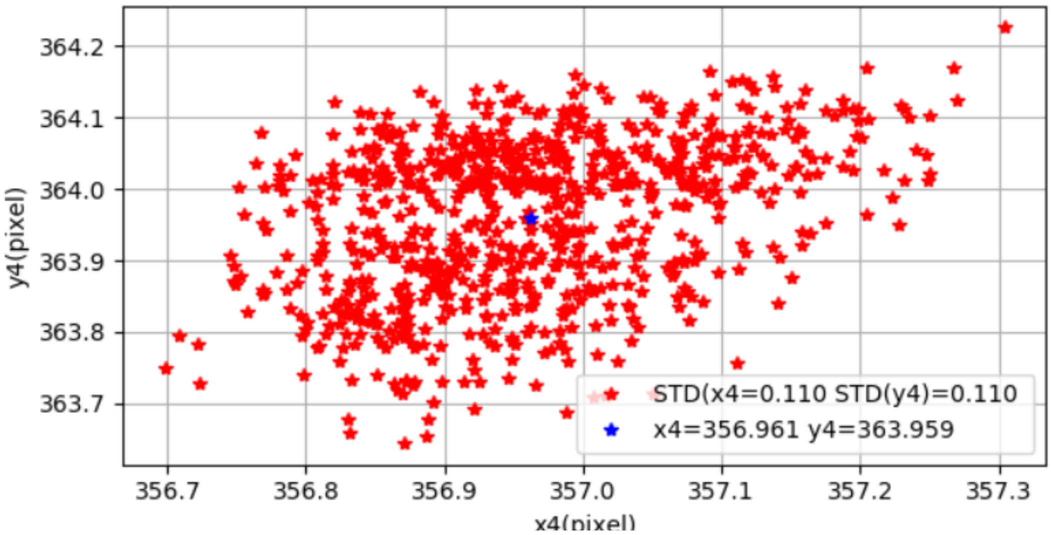


Figure 18 Mean and standard deviation (STD) of the point X4

As it is oberved in the previous figures the location of the each detected corner has small deviation in each image even if the rotating table is static, and the effects of this deviation are reflected in the precision (σ) of the rotation angle. In the Figure 19(a) it is shown that the rotation angle has a mean value of 158.777°, and the standard deviaton (σ) equals to 0.156°. The distance showed in the Figure 19(b) represent the modulus of the translation vector $\boldsymbol{T_C}$.

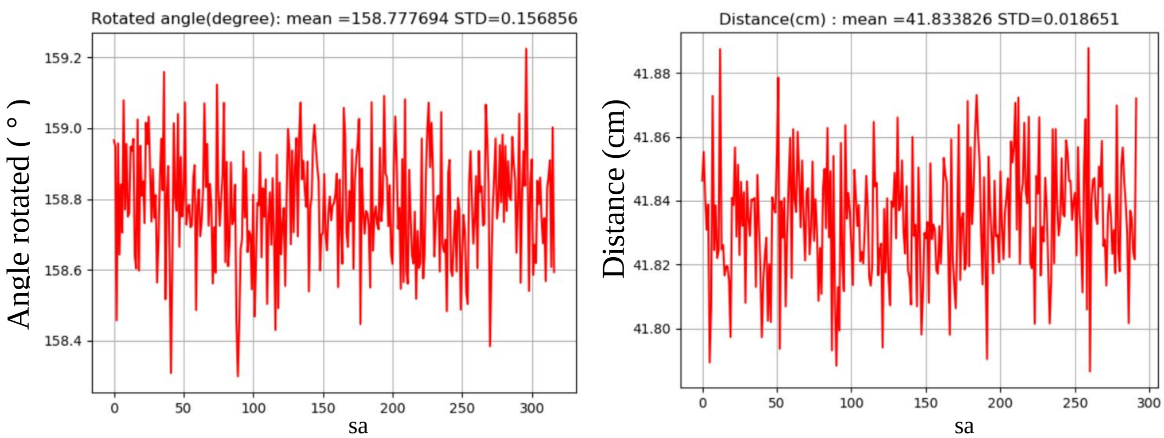


Figure 19 (a) Estimation of initial angle position with the rotating table being static. (b) Estimated distance with the rotating table being static.

Another experiment has been peformed where the rotating table rotates 90° around the axis-Z. In the Figure 20 , it is seen that, as it is expected, the estimated rotated angle is close to 90°. Additionally, the featuring of some peaks are seen, which appers due to the corner detecter's errors

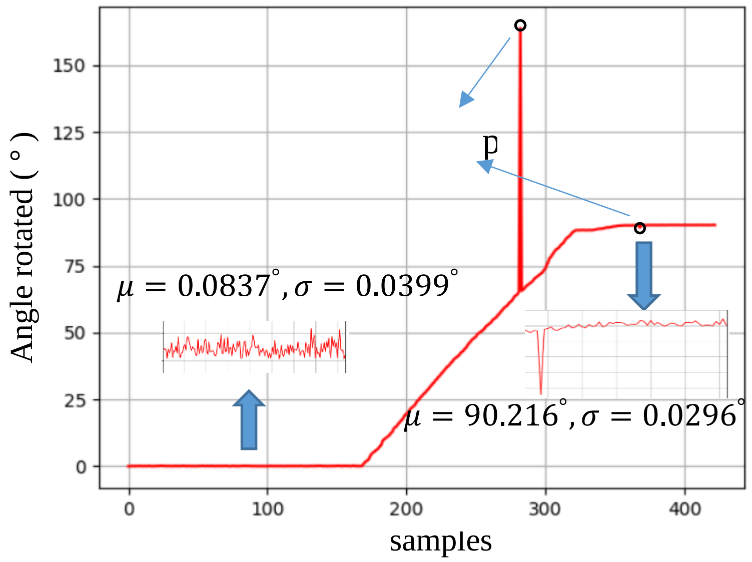


Figure 20 Estimated angle position with a rotation of 90°

The angular velocity can be calculated from consecutive rotation matrices, see Figure 21.

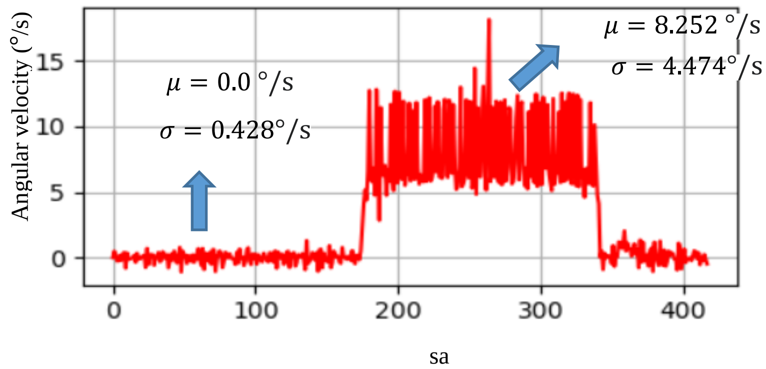


Figure 21 Angular velocity (°/s)

However, it can be observed that the angular velocity measurement is strongly imprecise, its standard deviation σ can reach 4.474°/s. In order to improve the angular velocity precision kalman filter is required to be implemented.

5 Extended Kalman Filter and system modeling

In this section is given a briefly introduction to Extended Kalman filter (EKF) [17-19], whereby it is pretended to improve the 3-D object attitude and angular velocity precision taking in to account the state-space models of our system.

5.1 Extended Kalman Filter

A system can be expressed as a continuous-time as follow:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t) + \boldsymbol{w}(t) \tag{5.1}$$

$$\boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{x}(t)) + \boldsymbol{v}(t) \tag{5.2}$$

The equation (5.1) represent the motion equation of the system, where $\boldsymbol{f}$ represents the state transition model, which depends on the state vector $\boldsymbol{x}$. With regard to the equation **Error! Reference source not found.**, $\boldsymbol{z}$ is called the measurement vector and $\boldsymbol{h}$ is called observation model.

Due to the fact that every system is affected by external and inherent noise, $\boldsymbol{w}$ and $\boldsymbol{v}$ are supposed to be noises with Gaussian distribution with zero expected value, $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}(t))$ and $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}(t))$.

Similarly, a nonlinear system can be expressed as a discrete-time system as follow:

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_{k-1} \tag{5.3}$$

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{5.4}$$

Where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are supposed to be noises with Gaussian distribution with zero expected value, $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$ and $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$.

Considering the continuous-time nonlinear system in the equations (5.1) and (5.2) the EKF is described below.

Let $\hat{\boldsymbol{x}}_k^+$ be the posteriori estimation of the state vector estimation at t_k , let $\hat{\boldsymbol{x}}_{k+1}^-$ be the priori estimation of the state vector at the moment of time t_{k+1}, $\hat{\boldsymbol{x}}_{k+1}^-$ is calculated by integration of nonlinear equation (5.1) without considering the noise component $\boldsymbol{w}$ using the state vector $\hat{\boldsymbol{x}}_k^+$.

The Discrete Riccatti equation is used for prediction of the error covariance matrix vector estimation P_{k+1}^- at time t_{k+1}.

$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{F}_k \boldsymbol{P}_k^{+} \boldsymbol{F}_k^{T} + \boldsymbol{Q}_k \tag{5.5}$$

Where $\boldsymbol{Q}_k = \boldsymbol{Q}(t_k)$**,** and $\boldsymbol{F}_k$ is the linearization of the state transition model $\boldsymbol{f}$ in the neighborhood of $\widehat{\boldsymbol{x}}_k^{+}$,called transition matrix from the state $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$, let $\boldsymbol{P}_k^{+}$ be the error covariance matrix at t_k.

Due to the fact that the measurements are frequently taken in a discrete form, the measurement model $\boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{x}(t)) + \boldsymbol{v}(t)$ (5.2) are given by

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k) \tag{5.6}$$

Where $\boldsymbol{x}_{\boldsymbol{k}} = \boldsymbol{x}(t_k)$**.** The gain matrix $\boldsymbol{K}_k$ can be written as

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^{-} \boldsymbol{H}_{k+1}^{T} [\boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1}^{-} \boldsymbol{H}_{k+1}^{T} + \boldsymbol{R}]^{-1} \tag{5.7}$$

Where $\boldsymbol{H}_k$ is the linearization of the observation model in the neighborhood of $\widehat{\boldsymbol{x}}_{k+1}^{-}$.

The corrected posteriori estimation is $\widehat{\boldsymbol{x}}_{k+1}^{+}$ of the Kaman filter is given by

$$\widehat{\boldsymbol{x}}_{k+1}^{+} = \widehat{\boldsymbol{x}}_{k+1}^{-} + \boldsymbol{K}_{k+1}[\boldsymbol{z}_{k+1} - \boldsymbol{h}(\widehat{\boldsymbol{x}}_{k+1}^{-})] \tag{5.8}$$

A posteriori estimation for the error covariance matrix is given by the formula

$$\boldsymbol{P}_{k+1}^{+} = [I - \boldsymbol{K}_{k+1} \boldsymbol{H}_{k+1}] \boldsymbol{P}_{k+1}^{-} \tag{5.9}$$

where I is an identity matrix.

The EKF algorithm for discrete-time system is remarkably similar for continuous-time system, but with $\widehat{\boldsymbol{x}}_{k+1}^{-}$ being calculated by means of the nonlinear equation (5.2) without considering the noise component $\boldsymbol{w}_{k-1}$ using the state vector $\widehat{\boldsymbol{x}}_k^{+}$.

The error covariance matrix vector estimation $\boldsymbol{P}_{k+1}^{-}$ at time t_{k+1} is calculated from the equation (5.5).

5.2 State-space modeling

In the Figure 22 the studied system in this thesis work is shown, The camera captures the object’s movement by taking photographs at a certain frequency (see Figure 22). Then by means of image processing it is detected and determined the position of the points $X_i{}'$ in the image. The points represent the projection of the points X_i into the image.

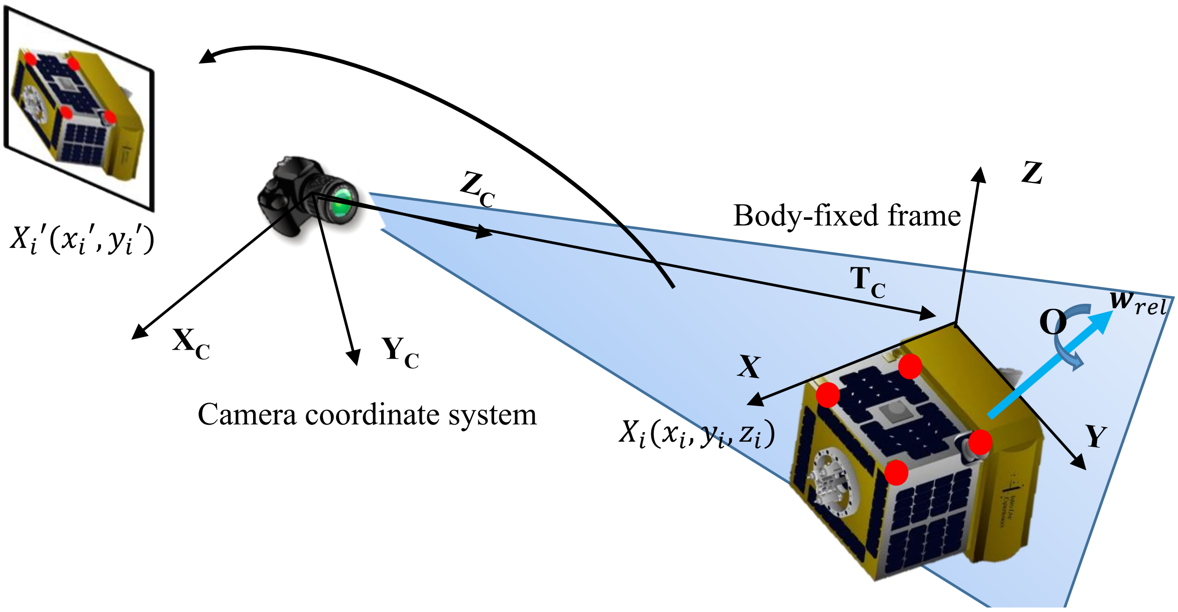


Figure 22 System Diagram

The angular motion of the object can be defined by means of its rotation matrix R with respect to the CCS, and its angular velocity $\boldsymbol{w}_{rel}$, which is respect to the BF.

The continuous-time angular motion equation can be obtained using Poisson equation for relative motion also can be expressed using quaternions

$$\dot{\Lambda} = \frac{1}{2}\Lambda \circ \boldsymbol{w}_{rel}, \ |\Lambda| = 1 \tag{5.10}$$

and can be written in a matrix form

$$\dot{\Lambda} = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{w}_{rel}{}^{T} \\ \boldsymbol{w}_{rel} & -[\boldsymbol{w}_{rel}]_{x} \end{bmatrix}\Lambda \tag{5.11}$$

$$\dot{\Lambda} = \frac{1}{2}\Psi(\boldsymbol{w}_{rel})\Lambda \tag{5.12}$$

Were $\Psi(\boldsymbol{w}_{rel})$ is defined as follow:

$$\Psi(\boldsymbol{w}_{rel}) = \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \tag{5.13}$$

The solution of the equation (5.10) for interval of time Δt, where $\boldsymbol{w}_{rel}$ can be assumed to be constant, can be written in a linear discrete-time form:

$$\Lambda_k = [I_{4x4} + \frac{1}{2}\Psi_{k-1}\Delta t]\Lambda_{k-1},\ \Delta t = t_k - t_{k-1} \tag{5.14}$$

Observation model is a function that provides information whereby directly or indirectly allow for estimation of the state of the system. Therefore, an observation model is closely related to the sensors function.

In the Figure 23 is shown that camera function can be written as a mathematical model by means of the measurement model h_{X_i} expressed in the equation (4.18) and in the equation (4.19), which are based on Rodrigues' rotation formula and quaternions respectively.

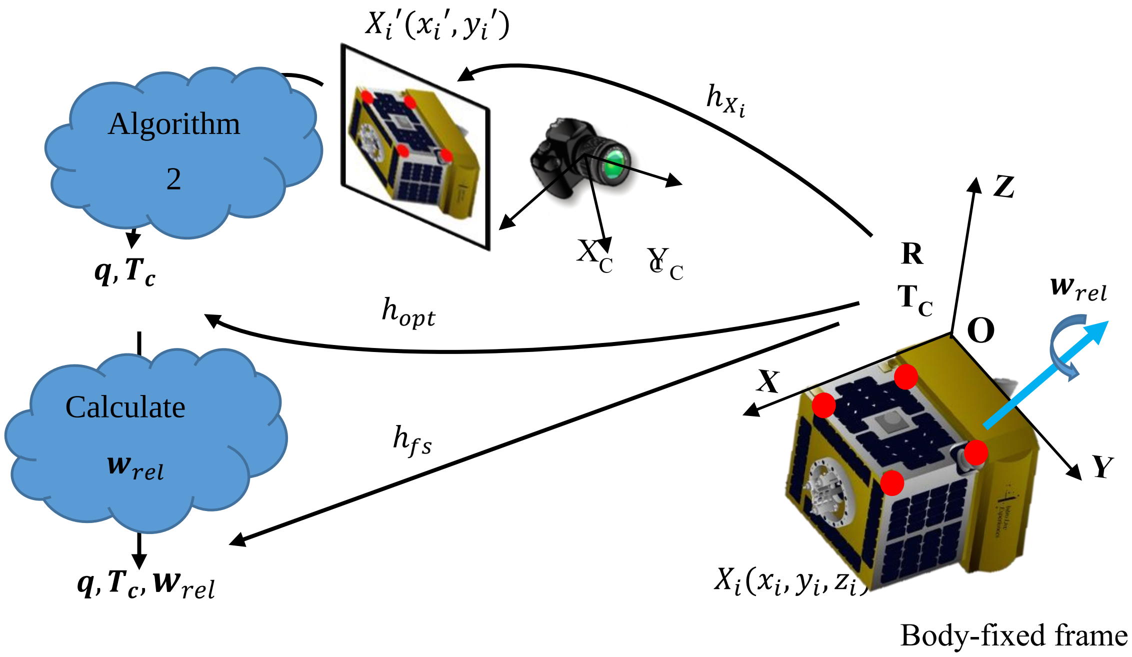


Figure 23 Measurement models

Additionally, in the section 4.5 it has been shown that by means of the Algorithm 2 it is possible to determine 3-D object attitude by obtaining the vector $\boldsymbol{q}$, this process can be represented by the observation model h_{opt} represented in the Figure 23.

The observation model h_{opt} is expressed as follow:

$$\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{T_c} \end{bmatrix} = h_{opt}(\boldsymbol{q}, \boldsymbol{T_c}) \tag{5.15}$$

Where $\boldsymbol{I_6}$ is an identity matrix. As it is shown in the equation (5.15), the measurement model h_{opt} has the advantage of being represented by a linear function. However, for measuring it requires more computational time. The measurement model h_{X_i}, in contrast, does not requires much computational time, but is strongly nonlinear.

In some cases, it is required to add angular velocity measurements, which are obtained from consecutive 3-D object attitudes measured every period of time Δt. Then the measurement model h_{fs} can be defined as follow:

$$\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{T_c} \\ \boldsymbol{w}_{rel} \end{bmatrix} = h_{fs}(\boldsymbol{q}, \boldsymbol{T_c}, \boldsymbol{w}_{rel}) \tag{5.16}$$

In order to apply the Extender Kalman Filter, it is required that our system be represented by means of state-space model. Different state-space models are described below, and then in the section 6 the best state-space model for Kalman filter is going to be choose.

In order to apply the Extender Kalman Filter, it is required that our system be represented by means of state-space model. Different state-space models are described below, and then in the section 6 the best state-space model for Kalman filter is going to be choose. In this chapter is considered to use a state-space model based on quaternion. However, a state-space model based on Rodrigues rotation formula is given in detailed in the Appendix A.

5.2.1 State-space model based on Quaternions

Let the state vector be represented by:

$$\boldsymbol{x} = [\boldsymbol{q}^T, \boldsymbol{T_c}^T, \boldsymbol{w}_{rel}{}^T]^T \tag{5.17}$$

Where $\boldsymbol{q}$ represents the vector part of the unit quaternion, $\boldsymbol{T_c}$ represent the distance vector between the camera and the BF, and $\boldsymbol{w}_{rel}$ represents angular velocity with respect to the BF.

5.2.1.1 Observation model

By means of the measurement model h_{X_i} using quaternions, defined in the equation (4.19), Let the observation model $\boldsymbol{h_q}$ be defined as follow:

$$\boldsymbol{h_q}(\boldsymbol{x}) = \begin{bmatrix} xp_1 \\ yp_1 \\ xp_2 \\ yp_2 \\ xp_3 \\ yp_3 \\ xp_4 \\ yp_4 \end{bmatrix} = \begin{bmatrix} h_{X_1}(\boldsymbol{q}, T_c) \\ h_{X_2}(\boldsymbol{q}, T_c) \\ h_{X_3}(\boldsymbol{q}, T_c) \\ h_{X_4}(\boldsymbol{q}, T_c) \end{bmatrix} \tag{5.18}$$

From the observation model defined in the equation (5.18), let $\boldsymbol{H_q}$ be the linearized matrix of the observation model $\boldsymbol{h_q}$

$$\boldsymbol{H_q} = \begin{bmatrix} \partial h_{X_1}/\partial \boldsymbol{x} \\ \partial h_{X_2}/\partial \boldsymbol{x} \\ \partial h_{X_3}/\partial \boldsymbol{x} \\ \partial h_{X_4}/\partial \boldsymbol{x} \end{bmatrix} \tag{5.19}$$

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial h_{X_i}}{\partial \boldsymbol{q}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{T_c}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{w}_{rel}} \end{bmatrix},\ i = 1, \dots, 4 \tag{5.20}$$

Where $\partial h_{X_1}/\partial \boldsymbol{q}$ and $\partial h_{X_1}/\partial \boldsymbol{T_c}$ were defined in the equation (2.62) and (2.53) respectively.

From the measurement model h_{opt} , defined in the equation (5.15), the observation model $\boldsymbol{h_{opt}}$ can be defined as follow:

$$\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{T_c} \end{bmatrix} = \boldsymbol{h_{opt}}(\boldsymbol{q}, \boldsymbol{T_c}) \tag{5.21}$$

Let $\boldsymbol{H_{opt}}$ be the linearized matrix of the observation model $\boldsymbol{h_{opt}}$ defined as:

$$\boldsymbol{H_{opt}} = \frac{\partial \boldsymbol{h_{opt}}}{\partial x} = \begin{bmatrix} \frac{\partial q}{\partial x} \\ \frac{\partial T_c}{\partial x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I_3} & \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} \\ \boldsymbol{0_{3x3}} & \boldsymbol{I_3} & \boldsymbol{0_{3x3}} \end{bmatrix} \tag{5.22}$$

Where $\boldsymbol{I_3}$ is identity matrix.

Additionally, from the measurement model h_{fs} , defined in the equation (5.16), the observation model $\boldsymbol{h_{fs}}$ can be defined as follow:

$$\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{T_c} \\ \boldsymbol{w}_{rel} \end{bmatrix} = \boldsymbol{h_{fs}}(\boldsymbol{q}, \boldsymbol{T_c}, \boldsymbol{w}_{rel}) \tag{5.23}$$

Let $\boldsymbol{H_{fs}}$ be the linearized matrix of the observation model $\boldsymbol{h_{fs}}$ defined as:

$$\boldsymbol{H}_{fx} = \frac{\partial \boldsymbol{h}_{fx}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}} \\ \frac{\partial \boldsymbol{T}_c}{\partial \boldsymbol{x}} \\ \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{I}_3 \end{bmatrix} \tag{5.24}$$

It is important to mention that the observation models $\boldsymbol{h}_q$, $\boldsymbol{h}_{opt}$ and $\boldsymbol{h}_{fs}$ are discrete-time measurements, and they are suitable for discrete-time state transition model $\boldsymbol{f}$.

Due to the fact that the measurements, in most physical continuous-time system, are frequently taken in a discrete form the observation models $\boldsymbol{h}_q$, $\boldsymbol{h}_{opt}$ and $\boldsymbol{h}_{fs}$ are completely suitable for the continuous-time state transition model.

5.2.1.2 State transition model

5.2.1.2.1 Discrete-time model

From the equation (5.14), the state transition model $\boldsymbol{f}$ can be expressed in a discrete-time form:

$$\Lambda_k = [I_{4x4} + \tfrac{1}{2}\Psi_{k-1}\Delta t]\Lambda_{k-1} \tag{5.25}$$

$$\boldsymbol{T}_{c_k} = \boldsymbol{T}_{c_{k-1}} \tag{5.26}$$

$$\boldsymbol{w}_{rel_k} = \boldsymbol{w}_{rel_{k-1}} \tag{5.27}$$

As it has been mentioned previously, the state transition models for $\boldsymbol{T}_{c_k}$ and $\boldsymbol{w}_{rel_k}$ are unknown, thus it is convenient to consider them to be constant for small period of time Δt.

From the equations (5.25), (5.26) and (5.27), let $\boldsymbol{F}$ be the linearized matrix of the state transition model defined as:

$$\boldsymbol{F} = \begin{bmatrix} \partial \boldsymbol{q}_k / \partial \boldsymbol{x} \\ \partial \boldsymbol{T}_{c_k} / \partial \boldsymbol{x} \\ \partial \boldsymbol{w}_{rel_k} / \partial \boldsymbol{x} \end{bmatrix} \tag{5.28}$$

In order to obtain the expression for $\partial \boldsymbol{q}_k / \partial \boldsymbol{x}$, it is performed $\partial \Lambda_k / \partial \boldsymbol{x}$

$$\frac{\partial \Lambda_k}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \Lambda_k}{\partial \boldsymbol{q}} & \frac{\partial \Lambda_k}{\partial \boldsymbol{T}_c} & \frac{\partial \Lambda_k}{\partial \boldsymbol{w}_{rel}} \end{bmatrix} \tag{5.29}$$

The equation (5.25) can be expressed as follow:

$$\Lambda_k = \Lambda_{k-1} + \tfrac{1}{2}\Psi_{k-1}\Lambda_{k-1}\Delta t \tag{5.30}$$

Then $\partial\Lambda_k/\partial\boldsymbol{q}$ is

$$\frac{\partial\Lambda_k}{\partial\boldsymbol{q}} = \frac{\partial\Lambda_{k-1}}{\partial\boldsymbol{q}} + \frac{\Delta t}{2}\frac{\partial(\Psi_{k-1}\Lambda_{k-1})}{\partial\boldsymbol{q}} \tag{5.31}$$

Where $\partial\Lambda_{k-1}/\partial\boldsymbol{q}$ is:

$$\frac{\partial\Lambda_{k-1}}{\partial\boldsymbol{q}} = \begin{bmatrix} \frac{-q_1}{q_0} & \frac{-q_2}{q_0} & \frac{-q_3}{q_0} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.32}$$

And $\partial(\Psi_{k-1}\Lambda_{k-1})/\partial\boldsymbol{q}$ is:

$$\frac{\partial(\Psi_{k-1}\Lambda_{k-1})}{\partial\boldsymbol{q}} = \begin{bmatrix} -w_x & -w_y & -w_z \\ \frac{-q_1}{q_0}w_x & \frac{-q_2}{q_0}w_x + w_z & \frac{-q_3}{q_0}w_x - w_y \\ \frac{-q_1}{q_0}w_y - w_z & \frac{-q_2}{q_0}w_y & \frac{-q_3}{q_0}w_y + w_x \\ \frac{-q_1}{q_0}w_z - w_y & \frac{-q_2}{q_0}w_z - w_x & \frac{-q_3}{q_0}w_z \end{bmatrix} \tag{5.33}$$

Thus, $\partial\Lambda_k/\partial\boldsymbol{q}$ from the equation (5.31) can be rewritten as follow:

$$\frac{\partial\Lambda_k}{\partial\boldsymbol{q}} = \begin{bmatrix} \frac{\partial q_0}{\partial\boldsymbol{q}} \\ \frac{\partial q_1}{\partial\boldsymbol{q}} \\ \frac{\partial q_2}{\partial\boldsymbol{q}} \\ \frac{\partial q_3}{\partial\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} -\frac{q_1}{q_0} - \frac{w_x\Delta t}{2} & -\frac{q_2}{q_0} - \frac{w_y\Delta t}{2} & -\frac{q_3}{q_0} - \frac{w_z\Delta t}{2} \\ 1 - \frac{q_1}{q_0}\frac{w_x\Delta t}{2} & -\frac{q_2}{q_0}\frac{w_x\Delta t}{2} + \frac{w_z\Delta t}{2} & -\frac{q_3}{q_0}\frac{w_x\Delta t}{2} - \frac{w_y\Delta t}{2} \\ -\frac{q_1}{q_0}\frac{w_y\Delta t}{2} - \frac{w_z\Delta t}{2} & 1 - \frac{q_2}{q_0}\frac{w_y\Delta t}{2} & -\frac{q_3}{q_0}\frac{w_y\Delta t}{2} + \frac{w_x\Delta t}{2} \\ -\frac{q_1}{q_0}\frac{w_z\Delta t}{2} - \frac{w_y\Delta t}{2} & -\frac{q_2}{q_0}\frac{w_z\Delta t}{2} - \frac{w_x\Delta t}{2} & 1 - \frac{q_3}{q_0}\frac{w_z\Delta t}{2} \end{bmatrix} \tag{5.34}$$

Then the $\partial\boldsymbol{q}_k/\partial\boldsymbol{q}$ is:

$$\frac{\partial\boldsymbol{q}_k}{\partial\boldsymbol{q}} = \begin{bmatrix} \frac{\partial q_1}{\partial\boldsymbol{q}} \\ \frac{\partial q_2}{\partial\boldsymbol{q}} \\ \frac{\partial q_3}{\partial\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} 1 - \frac{q_1}{q_0}\frac{w_x\Delta t}{2} & -\frac{q_2}{q_0}\frac{w_x\Delta t}{2} + \frac{w_z\Delta t}{2} & -\frac{q_3}{q_0}\frac{w_x\Delta t}{2} - \frac{w_y\Delta t}{2} \\ -\frac{q_1}{q_0}\frac{w_y\Delta t}{2} - \frac{w_z\Delta t}{2} & 1 - \frac{q_2}{q_0}\frac{w_y\Delta t}{2} & -\frac{q_3}{q_0}\frac{w_y\Delta t}{2} + \frac{w_x\Delta t}{2} \\ -\frac{q_1}{q_0}\frac{w_z\Delta t}{2} - \frac{w_y\Delta t}{2} & -\frac{q_2}{q_0}\frac{w_z\Delta t}{2} - \frac{w_x\Delta t}{2} & 1 - \frac{q_3}{q_0}\frac{w_z\Delta t}{2} \end{bmatrix} \tag{5.35}$$

Because Λ_k does not depends on $\boldsymbol{T_c}$, $\partial\Lambda_k/\partial\boldsymbol{T_c}$ is a null matrix, thus $\partial\boldsymbol{q}_k/\partial\boldsymbol{T_c}$ is:

$$\frac{\partial\boldsymbol{q}_k}{\partial\boldsymbol{T}_c} = 0_{3x3} \tag{5.36}$$

Additionally, $\partial\Lambda_k/\partial\boldsymbol{w}_{rel}$ is:

$$\frac{\partial \Lambda_k}{\partial \boldsymbol{w}_{rel}} = \begin{bmatrix} \frac{\partial q_0}{\partial \boldsymbol{w}_{rel}} \\ \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{w}_{rel}} \end{bmatrix} = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \quad (5.37)$$

$$\frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{w}_{rel}} = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \quad (5.38)$$

From the equations (5.26) and (5.27), the $\partial \boldsymbol{T}_{c_k}/\partial \boldsymbol{x}$ and $\partial \boldsymbol{w}_{rel_k}/\partial \boldsymbol{x}$ are expressed as follows:

$$\frac{\partial \boldsymbol{T}_{c_k}}{\partial \boldsymbol{x}} = [0_{3x3} \quad I_3 \quad 0_{3x3}] \quad (5.39)$$

$$\frac{\partial \boldsymbol{w}_{rel_k}}{\partial \boldsymbol{x}} = [0_{3x3} \quad 0_{3x3} \quad I_3] \quad (5.40)$$

5.2.1.2.2 Continuous-time model

The State-space model based on Quaternions can be also expressed in a continuous-time form:

$$\dot{\Lambda} = \frac{1}{2}\Lambda \circ \boldsymbol{w}_{rel}, \ |\Lambda| = 1 \quad (5.41)$$

$$\dot{\boldsymbol{T}}_{\boldsymbol{c}} = \boldsymbol{0}_{3x1} \quad (5.42)$$

$$\dot{\boldsymbol{w}}_{rel} = \boldsymbol{0}_{3x1} \quad (5.43)$$

The linearized matrix of state transition model $\boldsymbol{F}$ is defined as:

$$\begin{bmatrix} \delta\dot{\boldsymbol{q}} \\ \delta\dot{\boldsymbol{T}}_{\boldsymbol{c}} \\ \delta\dot{\boldsymbol{w}}_{rel} \end{bmatrix} = \boldsymbol{F} \begin{bmatrix} \delta\boldsymbol{q} \\ \delta\boldsymbol{T}_{\boldsymbol{c}} \\ \delta\boldsymbol{w}_{rel} \end{bmatrix} \quad (5.44)$$

Where

$$\boldsymbol{F} = \begin{bmatrix} -[\boldsymbol{w}_{rel}]_x & \boldsymbol{0}_{3x3} & 0.5\boldsymbol{I}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} \end{bmatrix} \quad (5.45)$$

The linearized matrix $\boldsymbol{H}$ of the observable model is the same as in the equation (5.19), because the $\boldsymbol{h}(\mathbf{x})$ is supposed that estimations of the state vector are not continuous but discrete

From the continuous-time model, its linearized matrix $\boldsymbol{F}_{\boldsymbol{Cont}}$ shows that $\delta\dot{\boldsymbol{w}}_{rel}/\delta\boldsymbol{x}$ results in a null matrix $\boldsymbol{0}_{\boldsymbol{3x9}}$, and from the gain matrix $\boldsymbol{K}_k$ in the equation $\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^{-}\boldsymbol{H}_{k+1}^{T}\ [\boldsymbol{H}_{k+1}\ \boldsymbol{P}⍰k + 1⍰-\boldsymbol{H}⍰k + 1⍰T⍰ + \boldsymbol{R}]⍰ - 1⍰$ (5.7) **Error!**

Reference source not found., is deducted that it is necessary angular velocity measurements to estimate angular velocity by EKF.

As the mentioned above, the measurement model $\boldsymbol{h}_{\boldsymbol{fs}}$ is suitable for the continuous-time model. With regard to discrete-time model, its linearized matrix $\boldsymbol{F}_{\boldsymbol{Disc}}$ shows that its component $\partial \boldsymbol{w}_{rel_k}/\partial \boldsymbol{x}$ is different than a null matrix, it gives the possibility to use the measurement models $\boldsymbol{h}_{\boldsymbol{opt}}$ and $\boldsymbol{h}_{\boldsymbol{fs}}$.

On the other hand, because of the high non-linearity of the measurement model $\boldsymbol{h}_{\boldsymbol{q}}$ it would require more analysis for future work.

6 Experiment and results

In this section, the results of the rotation matrix and angular velocity estimation by means of a rotating table are presented.

The facilities are shown in the Figure 24, where Aruco marker is used to allows to stablish the correspondences between the points in the coordinate system $OXYZ$ and the points located in the images. This marker is installed on the rotating table in a way that the maker will be rotated around its axis-Z.

The camera FI8918W, previously calibrated in section 4.4, is used to capture the Aruco marker's movement every period of time Δt, where $\Delta t = 1/15$ seconds.

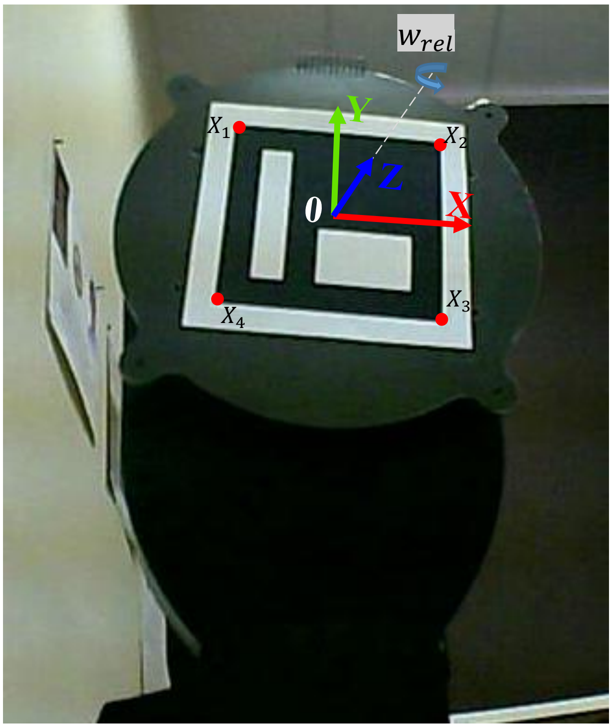


Figure 24 Rotary table rotates on the axis-Z

In order to estimate the rotation matrix and the angular velocity of the Aruco marker the EKF is implemented in according to the section 5.

Let $\boldsymbol{x} = [\boldsymbol{q}^T, \boldsymbol{w}_{rel}{}^T]^T$ be the state vector of the continuous-time state space model where $\boldsymbol{q} = [q_1, q_2, q_3]^T$ is the vector part of a unit quaternion $\boldsymbol{\Lambda}$. This quaternion represents the Aruco marker's rotation matrix with respect to the camera.

The vector $\boldsymbol{w}_{rel} = [w_x, w_y, w_z]^T$ represents the local angular velocity, with respect to the coordinate system $OXYZ$.

The process model is represented by the equations (5.41) and (5.43). The observation model is based on the equation (5.23) and is defined as follows:

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{w}_{rel} \end{bmatrix} = \boldsymbol{h_{fs}}(\boldsymbol{q}, \boldsymbol{w}_{rel})$$

The translation vector $\boldsymbol{T_c}$ is not taken into account. The process model integration is performed every period of time Δt by means of the Runge-Kutta 4th order method.

The covariance matrix of the process noise $\boldsymbol{Q}$ is

$$\boldsymbol{Q} = diag([\sigma_{q1}^2, \sigma_{q2}^2, \sigma_{q3}^2, \sigma_{q4}^2, \sigma_{q5}^2, \sigma_{q6}^2])$$

where σ_{q1}^2 =1e-8, σ_{q2}^2 =1e-8, σ_{q3}^2 =1e-8, σ_{q4}^2 =9.243e-5, σ_{q5}^2 =1.329e-4, σ_{q6}^2=2.172e-5. The covariance of the observation noise $\boldsymbol{R}$ is

$$\boldsymbol{R} = diag([\sigma_{r1}^2, \sigma_{r2}^2, \sigma_{r3}^2, \sigma_{r4}^2, \sigma_{r5}^2, \sigma_{r6}^2])$$

where σ_{r1}^2=6.123e-4, σ_{r2}^2=1.318e-8, σ_{r3}^2=3.377e-7, σ_{r4}^2=2.73e-4, σ_{r5}^2=6.069e-4, σ_{r6}^2=3.627e-5.

Similarly, the state vector **x** is also used for the discrete-time state model, its process model is represented by the equation (5.25) and (5.27). The observation model is based on the equation (5.21) and is defined as follows:

$$\boldsymbol{q} = \boldsymbol{h_{opt}}(\boldsymbol{q})$$

The covariance matrix of the process noise $\boldsymbol{Q_k}$ is

$$\boldsymbol{Q_k} = diag([\sigma_{q_k1}^2, \sigma_{q_k2}^2, \sigma_{q_k3}^2, \sigma_{q_k4}^2, \sigma_{q_k5}^2, \sigma_{q_k6}^2])$$

where $\sigma_{q_k1}^2$ =1e-8, $\sigma_{q_k2}^2$ =1e-8, $\sigma_{q_k3}^2$ =1e-8, $\sigma_{q_k4}^2$ =9.243e-5, $\sigma_{q_k5}^2$ =1.329e-4, $\sigma_{q_k6}^2$=2.172e-5. The covariance of the observation noise $\boldsymbol{R_k}$ is

$$\boldsymbol{R_k} = diag([\sigma_{r_k1}^2, \sigma_{r_k2}^2, \sigma_{r_k3}^2, \sigma_{r_k4}^2, \sigma_{r_k5}^2, \sigma_{r_k6}^2])$$

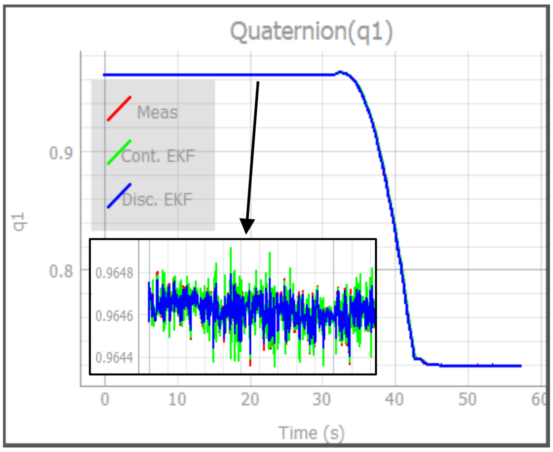
where $\sigma_{r_k1}^2$ =6.123e-4, $\sigma_{r_k2}^2$ =1.318e-8, $\sigma_{r_k3}^2$ =3.377e-7, $\sigma_{r_k4}^2$ =2.73e-4, $\sigma_{r_k5}^2$=6.069e-4, $\sigma_{r_k6}^2$=3.627e-5.

The covariance matrices of the process noise and observation noise were determined experimentally by means of a graphical user interface (GUI) developed in Python 3.7 during this thesis work. This GUI is implemented in order to fine-tune the covariance matrix.

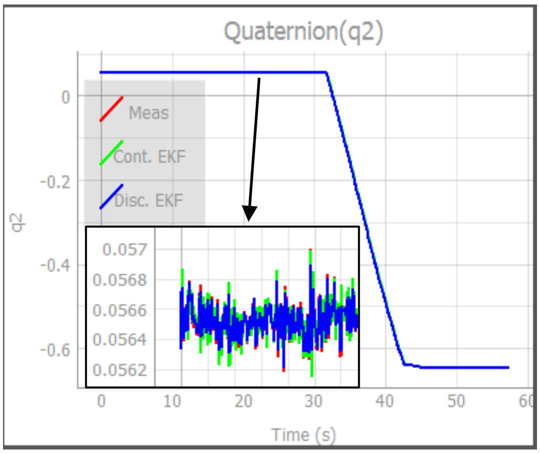
An experiment has been performed where the rotating table rotates 90° around the axis-Z. The next pictures shown three graphics defined as follows:

- Red line: Measurement without filter.
- Green line: Results for EKF using the continuous-time process model integrated every period of time Δt by means of the Runge-Kutta 4th order method.
- Blue lines: Results for EKF using the discrete-time process model.

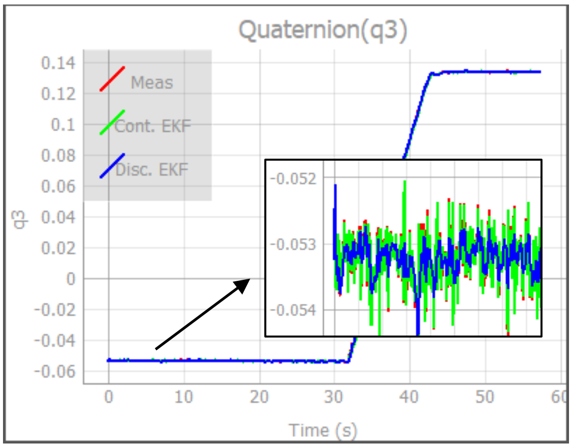
The components of the vector $\boldsymbol{q}$ are shown in the Figure 25.



(a) Quaternion component



(b) Quaternion component



(c) Quaternion component q3

Figure 25 Components of the vector part of the unit quaternion

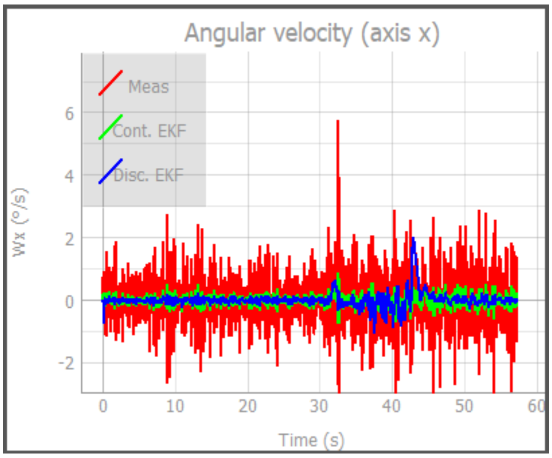
In the next table , the mean (μ) and standard deviation (σ) for the vector $\boldsymbol{q}$ are calculated for the first 30 seconds of the experiment, when the rotating table is static.

It can be seen that as for the orientation, which is determined by the quaternions, there was no significant improvement, this is because the measurement models $\boldsymbol{h}_{fs}$ and $\boldsymbol{h}_{opt}$ is already accurate for orientation determination.

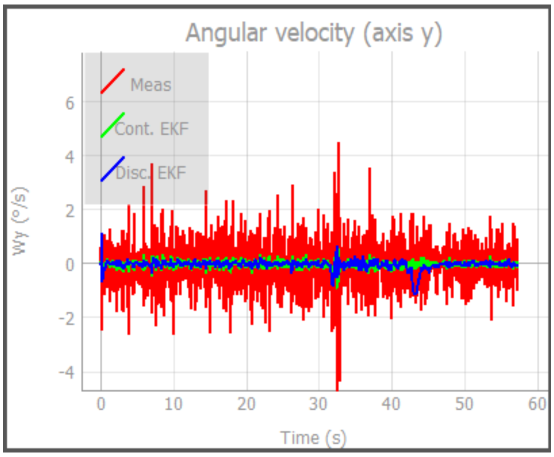
Table 1 Quaternion measurements for the first 30 seconds for a static rotating table

	Mean (μ)			**Standard deviation (σ)**		
	μ_{meas}	$\mu_{Cont.EKF}$	$\mu_{Disc.EKF}$	σ_{meas}	$\sigma_{Cont.EKF}$	$\sigma_{Disc.EKF}$
q_1	0.96463	0.96463	0.96463	0.00009	0.00010	0.00007
q_2	0.05652	0.05652	0.05652	0.00012	0.00012	0.00010
q_3	-0.05326	-0.05326	-0.05326	0.00043	0.00043	0.00026

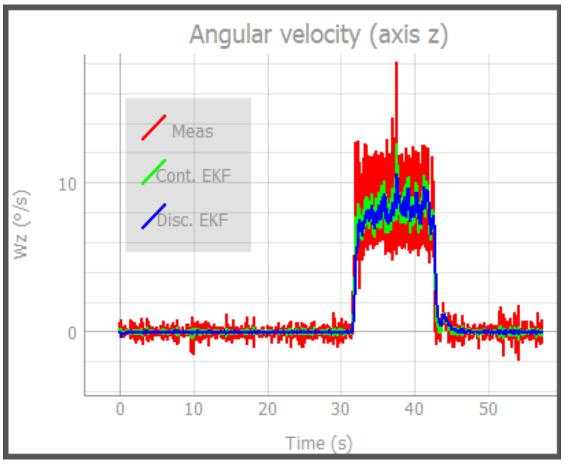
In the Figure 26, the results for the angular velocity estimation are shown. It can be noticed that there is a remarkable increase in precision.



(a) Angular velocity along the axis-X



(b) Angular velocity along the axis-Y



(c) Angular velocity along the axis-Z

Figure 26 Angular velocity (°/**s**) with respect to the coordinate system OXYZ

In the next table, it is shown the mean (μ) and standard deviation (σ) measurements of the angular velocity for the first 30 seconds of the experiment, when the rotating table is static. It can be seen that the precision increase can reach up to 89.7% by means of the Discrete EKF, it is a better option than the Continuous-time EKF which can reach up to 86.2%. However, It is important to mention that the previous results depend on the efficiency for covariance matrices determination. On the other hand, it is remarkable that an important precision increase is obtained with regard to angular velocity estimation.

Table 2 Angular velocity measurements for the first 30 seconds for a static rotating table

	Mean (°/s)			**Standard deviation (°/s)**			**Improve (%)**	
	μ_{meas}	$\mu_{Cont.EKF}$	$\mu_{Disc.EKF}$	σ_{meas}	$\sigma_{Cont.EKF}$	$\sigma_{Disc.EKF}$	**Cont.EKF**	**Disc.EKF**
w_x	0.00237	0.0019	-0.0013	0.8411	0.1503	0.0863	82.1	89.7
w_y	0.00008	-0.0012	-0.0036	1.005	0.1352	0.1117	86.5	88.8
w_z	-0.00216	-0.0016	-0.0054	0.405	0.1205	0.0511	70.3	87.4

In the Figure 27, The angle rotated around the axis-Z is shown, and as it is expected, the estimated rotated angle is close to 90° with an error of not more than 0.1 °.

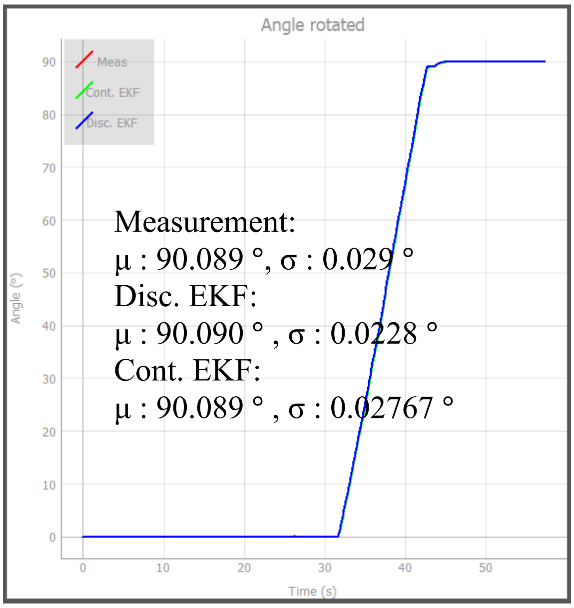


Figure 27 Angle rotated (°) around the axis-Z

The angle rotated is not calculated by angular velocity integration, but it is measured taking into account the first rotation matrix to the current rotation matrix, due to the fact that orientation estimation y more precise.

Conclusion

This work is dedicated to the problem of estimating the orientation of an object and its angular velocity by image processing. Two different approches were considered: The rotation matrix determination by means of the measurement model adapted for the use of quaternions, in addition the implementation of EKF for the angular velocity estimation.

As result of using quaternions, simulations showed that there is no difference with respect to the precision with its analog adapted measurement model for Rodrgiues rotation formula. However, using measurement model based on quaternions is a slight advantage in computing time.

Experiments for rotation matrix determination by means of quaternions showed high precision. However, the estimation of the angular velocity from consecutives rotation matrix has low precision, thus ,in order to improve the precision of the angular velocity measurement the EKF is implemented.

The EKF has been implemented taking into account the matrix rotation measurment ,which are express by quaternions, the result showed a significatly accuracy increase for angular velocity estimation. As for the rotation matrix, which is determined by the quaternions, there was no significant improvement, this is because the measurement is already accurate.

References

[1] Richard Hartley and Andrew Zisserman. *Multi View Geometry in computer vision* ,2nd Edition , 2004, pp 88-92, 155, 109

[2] Bouguet, J. Y. *Camera Calibration Toolbox for Matlab*, 2015, Available: http://www.vision.caltech.edu/bouguetj/calib_doc/ (дата обращения: 20.06.220).

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV*", 1st Edition, 2008, pp 375-378

[4] R. Orghidan, J. Salvi, M. Gordan, B. Orza. *Camera calibration using two or three vanishing points*, Proceedings of the Federated Conference on Computer Science and Information Systems pp. 123–130

[5] Z. Zhang. *Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*, Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, Greece, September 20-27,1999.

[6] R. Y. Tsai, "*A Versatile Camera Calibration Techniaue for High-Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses*, IEEE Journal on Robotics and Automation, 1987, Vol. 3, No. 4, pp. 323 – 344.

[7] Janne Heikkilä and Olli Silvén. *A Four-step Camera Calibration Procedure with Implicit Image Correction*, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, USA, June, 17-19 ,1997.

[8] M. Dhome, M. Richetin, J. Lapreste, G. Rives. *Determination of the Attitude of 3-D Objects from a Single Perspective View*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, Vol. 11, No. 12, pp. 1265 – 1278.

[9] H. Kim, B. You, G. D. Hager, S. Oh, Ch. W. Lee. *Three-Dimensional Pose Determination for a Humanoid Robot using Binocular Head* System, Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289), Kyongju, South Korea, South Korea, October, 17-21, 1999.

[10] Z. Zhong, J. Yi, D. Zhao. *Pose Estimation and Structure Recovery from Point Pairs*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, Spain, April, 18-22, 2005

[11] X. Zhang, Z. Zhang, Y. Li, X. Zhu. *Robust camera pose estimation from unknown or known line correspondences,* OSA Journal on Applied Optics,2012, Vol. 51, No. 7, pp. 936-948.

[12] G. Gallego, D.Scaramuzza. *Accurate Angular Velocity Estimation with an Event Camera,* IEEE Robotics and Automation Letters, 2017, Vol. 2, No. 2, pp. 632 - 639.

[13] S. Wang, Q. Li, B. Guan. *A computer vision method for measuring angular velocity*, Optics and Lasers in Engineering, Vol. 45, No. 11, pp. 1037-1048.

[14] Zhang Y, Wen C, Zhang Y. *Estimation of motion parameters from blurred images*. Pattern Recognition Letters, 2000, Vol. 21, No. 5, pp. 425-433.

[15] F. J. Romero-Ramirez, R. Muñoz-Salinas, R. Medina-Carnicer. *Speeded up detection of squared fiducial markers*, Image and Vision Computing, 2018, vol 76, pages 38-47.

[16] S. Garrido-Jurado, R. Muñoz Salinas, F.J. Madrid-Cuevas, R. Medina-Carnicer. *Generation of fiducial marker dictionaries using mixed integer linear programming*, Pattern Recognition, 2016, Vol. 51, pp. 481-491.

[17] D. Simon, “Optimaal State Estimation : Kalman, H∞, and Nonlinear Approaches”, 1st Edition.

[18] Y. Mashtakov, M. Ovchinnikov, F. Woske, B. Rievers. *Attitude determination and control system design for gravity recovery mission like GRACE*, Acta Astronautica, 2020, Vol. 173, pp. 172-182.

[19] D. Ivanov, M. Ovchinnikov. *Analytical study of microsatellite attitude determination algorithms*, 2015, Vol. 116, pp. 339-348.

[20] A.A.Boguslavsky, V.V.Sazonov, S.M.Sokolov, A.I.Smirnov, Kh.S.Saigiraev. *Automatic vision-based monitoring of the spacecraft docking approach with the International Space Station*, Proceedings of the First International Conference on Informatics in Control, Automation and Robotics, 2004, pp. 79-86

[21] Ivanov D.S., Karpenko S.O., Ovchinnikov M.Yu., Sakovich M.A. *Определение относительного движения спутников при их разделении по результатам обработки видеоизображения*, Препринт ИПМ им. М.В. Келдыша, 2012, № 57, 24 с.

[22] Коптев М.Д., Прошунин Н.Н., Иванов Д. С., *Определение движения макетов системы управления микроспутников на аэродинамическом столе с использованием видеокамеры*, Препринт ИПМ им. М.В. Келдыша, 2015, № 109, 32 с.

Appendix A. State-space model based on Rodrigues' rotation formula

The angular motion of the object can be defined by means of its rotation matrix R with respect to the CCS, and its angular velocity $\boldsymbol{w}_{rel}$, which is respect to the BF.

The continuous-time angular motion equation can be obtained using Poisson equation for relative motion

$$\dot{R} = R\Omega \tag{A. 1}$$

Where Ω is skew-symmetric matrix of the vector $\boldsymbol{w}_{rel}$:

$$\Omega = [\boldsymbol{w}_{rel}]_x = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{A.2}$$

Due to the fact that the angular velocity $\boldsymbol{w}_{rel}$ depends on time, it is convenient to perform an analysis for small enough interval of time $[t_0, t_f]$, where $\boldsymbol{w}_{rel}$ can be considered constant; Taking into account the mentioned above, the general solution can be written as follow:

$$R(t) = R_0 e^{\Omega t} \tag{A.3}$$

Where $R_0(t_0) = R_0$ and $t \in [t_0, t_f]$, additionally, the solution can be expressed in a discrete-time form:

$$R_k = R_{k-1} e^{\Omega_{k-1}\Delta t},\ \Delta t = t_k - t_{k-1},\ k = 1,2,3, \dots \tag{A.4}$$

Where Δt is small enough to assumed Ω_{k-1} to be constant in the interval of time $[t_k, t_{k-1}]$. Also Δt represents the period of time whereby the system is updated.

By mean of the definition of matrix exponential the equation can be expressed as a Taylor series

$$R_k = R_{k-1}(I + \Omega_{k-1}\Delta t + \tfrac{1}{2}(\Omega_{k-1}\Delta t)^2 + \cdots) \tag{A.5}$$

Considering only the first two terms, a linear form can be given as follow:

$$R_k = R_{k-1}(I + \Omega_{k-1}\Delta t) \tag{A.6}$$

Below is described the state-space model based on Rodrigues' rotation formula.

The measurement model $h_{X_1}(\boldsymbol{v}, \boldsymbol{T_c})$ originally does not depends on the angular velocity $\boldsymbol{w}_{rel}$. However, due to the fact that the angular velocity is needed in every angular motion, the angular velocity $\boldsymbol{w}_{rel}$ must be considered in the state vector $\boldsymbol{x}$. Let the state vector be $\boldsymbol{x} = [\boldsymbol{v}^T, \boldsymbol{T_c}^T, \boldsymbol{w}_{rel}{}^T]^T$ and the observation model be defined as follow:

$$\boldsymbol{h_{rod}}(\boldsymbol{x}) = \begin{bmatrix} xp_1 \\ yp_1 \\ xp_2 \\ yp_2 \\ xp_3 \\ yp_3 \\ xp_4 \\ yp_4 \end{bmatrix} = \begin{bmatrix} h_{X_1}(\boldsymbol{v}, \boldsymbol{T_c}) \\ h_{X_2}(\boldsymbol{v}, \boldsymbol{T_c}) \\ h_{X_3}(\boldsymbol{v}, \boldsymbol{T_c}) \\ h_{X_4}(\boldsymbol{v}, \boldsymbol{T_c}) \end{bmatrix} \tag{A.7}$$

It is important to notice that the previous observation model consists on the projection of for points from the BF to the ICS.

In order to apply the Extended Kalman filter, it is required a linearization of the state transition and observation model.

Let $\boldsymbol{H_{rod}}$ be the linearized matrix of the observation model $\boldsymbol{h_{rod}}(\boldsymbol{x})$ defined as:

$$\boldsymbol{H_{rod}} = \begin{bmatrix} \partial h_{X_1}/\partial \boldsymbol{x} \\ \partial h_{X_2}/\partial \boldsymbol{x} \\ \partial h_{X_3}/\partial \boldsymbol{x} \\ \partial h_{X_4}/\partial \boldsymbol{x} \end{bmatrix} \tag{A.8}$$

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial h_{X_i}}{\partial \boldsymbol{v}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{T_c}} & \frac{\partial h_{X_i}}{\partial \boldsymbol{w}_{rel}} \end{bmatrix}, \ i = 1, \ldots, 4 \tag{A.9}$$

Where $\partial h_{X_i}/\partial \boldsymbol{v}$ and $\partial h_{X_i}/\partial \boldsymbol{T_c}$ were defined in the equation (2.37) and (2.53) respectively, and $\partial h_{X_i}/\partial \boldsymbol{w}_{rel}$ results in a null matrix $\boldsymbol{0_{2x2}}$, because h_{X_i} does not depends on $\boldsymbol{w}_{rel}$.

According to the state vector, the state transition model in continuous-time is required to be in the next form:

$$\dot{\boldsymbol{v}} = f_1(\boldsymbol{v}, \boldsymbol{T_c}, \boldsymbol{w}_{rel}, t) \tag{A.10}$$

$$\dot{\boldsymbol{T}}_{\boldsymbol{c}} = f_2(\boldsymbol{v}, \boldsymbol{T_c}, \boldsymbol{w}_{rel}, t) \tag{A.11}$$

$$\dot{\boldsymbol{w}}_{rel} = f_3(\boldsymbol{v}, \boldsymbol{T_c}, \boldsymbol{w}_{rel}, t) \tag{A.12}$$

However, there is no a direct expression for previous differential system of equations, therefore with Rodrigues' rotation formula it is preferable to work with discrete-time system with the next form:

$$\boldsymbol{v}_k = f_1(\boldsymbol{v}_{k-1}, \boldsymbol{T}_{\boldsymbol{c}_{k-1}}, \boldsymbol{w}_{rel_{k-1}}, k-1) \tag{A.13}$$

$$\boldsymbol{T}_{\boldsymbol{c_k}} = f_2(\boldsymbol{v}_{k-1}, \boldsymbol{T}_{\boldsymbol{c}_{k-1}}, \boldsymbol{w}_{rel_{k-1}}, k-1) \tag{A.14}$$

$$\boldsymbol{w}_{rel_k} = f_3(\boldsymbol{v}_{k-1}, \boldsymbol{T}_{\boldsymbol{c}_{k-1}}, \boldsymbol{w}_{rel_{k-1}}, k-1) \tag{A.15}$$

The state transition model for $\boldsymbol{v}_k$ can be derived from the equation (A.6), let the rotation matrix R_k be defined as follow:

$$R_k = R_{k-1}[I + \Omega_{k-1}\Delta t] \tag{A.16}$$

$$R_k = \begin{bmatrix} r_{k,11} & r_{k,12} & r_{k,13} \\ r_{k,21} & r_{k,22} & r_{k,23} \\ r_{k,31} & r_{k,32} & r_{k,31} \end{bmatrix} \tag{A.17}$$

Where R_{k-1} is the matrix rotation at moment of time t_{k-1}, R_{k-1} is defined as function of $\boldsymbol{v}_{k-1}$ in the equation (2.16), and Ω_{k-1} is the skew-symmetric matrix of the vector $\boldsymbol{w}_{rel_{k-1}}$ as in the equation (A.2). Thus, R_k can be expressed as $R_k(\boldsymbol{v}_{k-1}, \boldsymbol{w}_{rel_{k-1}})$.

As it is widely known, from rotation matrix angle rotation α_k can be determined as follow:

$$\alpha_k = \arccos\left(\frac{Tr(R_k)-1}{2}\right) \tag{A.18}$$

Where $Tr(R_k)$ is the trace of the matrix R_k. The axis of rotation $\boldsymbol{u}_k$ is defined as:

$$\boldsymbol{u}_k = \frac{\boldsymbol{b}}{2sin\alpha_k}, \ |\boldsymbol{u}_k| = 1 \tag{A.19}$$

Where the vector $\boldsymbol{b}$ is:

$$\boldsymbol{b} = [r_{k,32} - r_{k,23}, r_{k,13} - r_{k,31}, r_{k,21} - r_{k,12}]^T \tag{A.20}$$

Then $\boldsymbol{v}_k$ is obtained as a function of $\boldsymbol{v}_{k-1}$ and $\boldsymbol{w}_{rel_{k-1}}$ as follow:

$$\boldsymbol{v}_k = f_1(\boldsymbol{v}_{k-1}, \boldsymbol{w}_{rel_{k-1}}) = \alpha_k \vec{u}_k \tag{A.21}$$

Due to the fact that the state transition models f_2 and f_3 are unknown for $\boldsymbol{T}_{\boldsymbol{c}_{\boldsymbol{k}}}$ and $\boldsymbol{w}_{rel_k}$ respectively, it is possible to consider them to be constant for small period of time Δt.

$$\boldsymbol{T}_{\boldsymbol{c}_k} = \boldsymbol{T}_{\boldsymbol{c}_{k-1}} \tag{A.22}$$

$$\boldsymbol{w}_{rel_k} = \boldsymbol{w}_{rel_{k-1}} \tag{A.23}$$

From the equations (A.13), (A.14) and (A.15) , Let $\boldsymbol{F}$ be the linearized matrix of state transition model $\boldsymbol{f}$ defined in the equation defined as:

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F_1} \\ \boldsymbol{F_2} \\ \boldsymbol{F_3} \end{bmatrix} = \begin{bmatrix} \partial f_1/\partial \boldsymbol{x} \\ \partial f_2/\partial \boldsymbol{x} \\ \partial f_3/\partial \boldsymbol{x} \end{bmatrix} \tag{A.24}$$

From the equation (A.21), The linearized matrix $\boldsymbol{F_1}$ is defined as follow:

$$\boldsymbol{F_1} = \frac{\partial f_1}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{x}} \tag{A.25}$$

Where $\boldsymbol{v}_k$ can be written as a function of the state vector $\boldsymbol{x}$:

$$\boldsymbol{v}_k = \alpha_k(\boldsymbol{x})\, \boldsymbol{u}_k(\boldsymbol{x}) \tag{A.26}$$

Then

$$\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{v}_k}{\partial \alpha_k}\frac{\partial \alpha_k}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{u}_k}\frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}} \tag{A.27}$$

Where $\partial \boldsymbol{v}_k/\partial \alpha_k = \boldsymbol{u}_k$, and $\partial \boldsymbol{v}_k/\partial \boldsymbol{u}_k = \alpha_k I_3$

$$\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{x}} = \boldsymbol{u}_k \frac{\partial \alpha_k}{\partial \boldsymbol{x}} + \alpha_k \frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}} \tag{A.28}$$

From the equation (A.18)**,** let $\phi = (Tr(R_k) - 1)/2$, then $\partial \alpha_k/\partial \boldsymbol{x}$ can be expressed by mean of partial derivatives:

$$\frac{\partial \alpha_k}{\partial \boldsymbol{x}} = \frac{\partial \alpha_k}{\partial \phi}\frac{\partial \phi}{\partial Tr}\frac{\partial Tr}{\partial R_k}\frac{\partial R_k}{\partial \boldsymbol{x}} \tag{A.29}$$

Where $\partial \alpha_k/\partial \phi = -1/\sqrt{1-\phi^2}$, $\partial \phi/\partial Tr = 1/2$, and $\partial \phi/\partial Tr$ is:

$$\frac{\partial \phi}{\partial Tr} = [1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1] \tag{A.30}$$

From the equation (A.19), let $\zeta = 1/(2sin\alpha_k)$, then $\boldsymbol{u}_k$ can be written as:

$$\boldsymbol{u}_k = \zeta(\alpha_k)\boldsymbol{b}(R_k) \tag{A.31}$$

Thus $\partial \boldsymbol{u}_k/\partial \boldsymbol{x}$ can be defined as:

$$\frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{u}_k}{\partial \zeta}\frac{\partial \zeta}{\partial \alpha_k}\frac{\partial \alpha_k}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{b}}\frac{\partial \boldsymbol{b}}{\partial R_k}\frac{\partial R_k}{\partial \boldsymbol{x}} \quad (A.32)$$

Where $\partial \boldsymbol{u}_k/\partial \zeta = \boldsymbol{b}$, $\partial \zeta/\partial \alpha_k = -cos(\alpha_k)/(2sin^2(\alpha_k))$, $\partial \boldsymbol{u}_k/\partial \boldsymbol{b} = 1/(2sin\alpha_k)$, and $\partial \boldsymbol{b}/\partial R_k$ is:

$$\frac{\partial v}{\partial R_k} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (A.33)$$

The equation (A.16) can be rewritten as follow:

$$R_k = R_{k-1}M \quad (A.34)$$

Where $M = I + \Omega_{k-1}\Delta t$, then by means of matrix calculus $\partial R_k/\partial \boldsymbol{x}$ is expressed as:

$$\frac{\partial R_k}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial R_{k-1}M}{\partial R_{k-1}} & \frac{\partial R_{k-1}M}{\partial M} \end{bmatrix} \begin{bmatrix} \frac{\partial R_{k-1}}{\partial \boldsymbol{x}} \\ \frac{\partial M}{\partial \boldsymbol{x}} \end{bmatrix} \quad (A.35)$$

$$\frac{\partial R_k}{\partial \boldsymbol{x}} = [M^T \otimes I_3 \quad I_3 \otimes R_{k-1}] \begin{bmatrix} \frac{\partial R_{k-1}}{\partial \boldsymbol{v}_{k-1}} & \frac{\partial R_{k-1}}{\partial \boldsymbol{T}_{c_{k-1}}} & \frac{\partial R_{k-1}}{\partial \boldsymbol{w}_{rel_{k-1}}} \\ \frac{\partial M}{\partial \boldsymbol{v}_{k-1}} & \frac{\partial M}{\partial \boldsymbol{T}_{c_{k-1}}} & \frac{\partial M}{\partial \boldsymbol{w}_{rel_{k-1}}} \end{bmatrix} \quad (A.36)$$

$$\frac{\partial R_k}{\partial \boldsymbol{x}} = [M^T \otimes I_3 \quad I_3 \otimes R_{k-1}] \begin{bmatrix} \frac{\partial R_{k-1}}{\partial \boldsymbol{v}_{k-1}} & \boldsymbol{0}_{9x3} & \boldsymbol{0}_{9x3} \\ \boldsymbol{0}_{9x3} & \boldsymbol{0}_{9x3} & \frac{\partial M}{\partial \boldsymbol{w}_{rel_{k-1}}} \end{bmatrix} \quad (A.37)$$

$$\frac{\partial R_k}{\partial \boldsymbol{x}} = \begin{bmatrix} M^T \otimes I_3 \frac{\partial R_{k-1}}{\partial \boldsymbol{v}_{k-1}} & \boldsymbol{0}_{9x3} & I_3 \otimes R_{k-1} \frac{\partial M}{\partial \boldsymbol{w}_{rel_{k-1}}} \end{bmatrix} \quad (A.38)$$

Where $\otimes$ is Kronecker product operator, the partial derivative $\partial R_{k-1}/\partial \boldsymbol{v}_{k-1}$ was calculated in the equation (2.47), and $\partial M/\partial \boldsymbol{w}_{rel_{k-1}}$ is:

$$\frac{\partial M}{\partial \boldsymbol{w}_{rel_{k-1}}} = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \end{bmatrix}^{\boldsymbol{T}} \Delta t \quad (A.39)$$

From the equation **Error! Reference source not found.** , the linearized matrix $\boldsymbol{F_2}$ can be defined as follow:

$$\boldsymbol{F_2} = \frac{\partial f_2}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{T}_{c_k}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{T}_{c_{k-1}}}{\partial \boldsymbol{v}_{k-1}} & \frac{\partial \boldsymbol{T}_{c_{k-1}}}{\partial \boldsymbol{T}_{c_{k-1}}} & \frac{\partial \boldsymbol{T}_{c_{k-1}}}{\partial \boldsymbol{w}_{rel_{k-1}}} \end{bmatrix} \quad (A.40)$$

$$\boldsymbol{F_1} = [0_{3x3} \quad I_3 \quad 0_{3x3}] \quad (A.41)$$

Similarly, from the equation (A.23) for the linearized matrix $\boldsymbol{F_3}$**:**

$$\boldsymbol{F_3} = \frac{\partial f_3}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{w}_{rel_k}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{w}_{rel_{k-1}}}{\partial \boldsymbol{v}_{k-1}} & \frac{\partial \boldsymbol{w}_{rel_{k-1}}}{\partial \boldsymbol{T}_{c_{k-1}}} & \frac{\partial \boldsymbol{w}_{rel_{k-1}}}{\partial \boldsymbol{w}_{rel_{k-1}}} \end{bmatrix} \tag{A.42}$$

$$\boldsymbol{F_3} = \begin{bmatrix} 0_{3x3} & 0_{3x3} & I_3 \end{bmatrix} \tag{A.43}$$

In this section was described how is the state-space modeling with Rodrigues rotation formula.