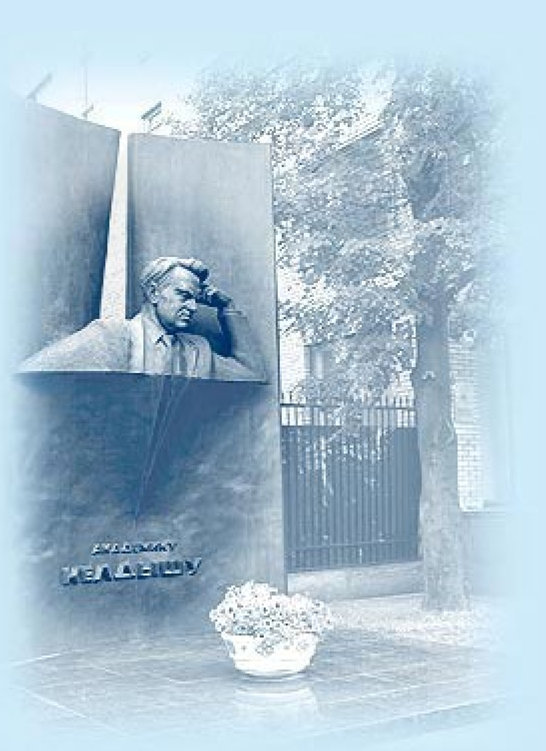
Keldysh Institute • Publication search
Keldysh Institute preprints • Preprint No. 27, 2021



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

J.C. Molina Saqui, S.S. Tkachev

Kalman filter application for the angular motion estimation by video processing

Recommended form of bibliographic references: Molina Saqui J.C., Tkachev S.S. Kalman filter application for the angular motion estimation by video processing // Keldysh Institute Preprints. 2021. No. 27. 27 p. https://doi.org/10.20948/prepr-2021-27-e
https://library.keldysh.ru/preprint.asp?id=2021-27&lg=e

О р д е н а   Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й   а к а д е м и и   н а у к

J. C. Molina Saqui, S. Tkachev

Kalman filter application for the angular motion estimation by video processing

Москва — 2020

Молина Саки Х. С., Ткачев С.С.

Использование фильтра Калмана для определения углового движения по видеоизображению

Работа посвящена задаче оценки ориентации объекта и его угловой скорости с помощью обработки видеоизображения. Для этого используется расширенный фильтр Калмана (РФК), где измерениями являются координаты точек изображения.

Результаты экспериментальных исследований показали, что при использовании модели измерения и РФК удается значительно улучшить оценку угловой скорости, в то время как, оценка кватерниона не улучшается по сравнению с локальными методами.

Ключевые слова: определение углового движения, обработка видеоизображения, расширенный фильтр Калмана

Molina Saqui J. C., Tkachev S.

Kalman filter application for the angular motion estimation by video processing

This work considers the problem of estimating the orientation and angular velocity of the object by image processing. To solve this problem, an approach based on the Extended Kalman filter (EKF), where the mesurements are the coordinates of the image points.

The results showed a significatly accuracy increase for the angular velocity estimation. As for the rotation quaternion, there was no significant improvement with respect to the local methods.

Key words: angular motion determination, image processing, Extended Kalman filter

Introduction

The angular motion is nowadays a crucial information for most applications, where autonomous motion control systems are involved. Likewise, in conventional navigation system is usually composed by inertial sensor such as accelerometers and gyroscopes, where the angular velocity information is expected to be provided. However, inertial sensors usually present accumulated drift, high-level noise sensitivity, and in-built errors.

When precise angular velocity measurements are strongly needed, the accurate gyroscopes are usually required but they are too expensive and unaffordable for cost-effective application. Therefore, gyroscope-free inertial and redundant sensors have started to be considered to get higher precision. In [1] S. Zhen by means of the data, provided by the gyroscope-free inertial system, and the Unscented Kalman Filter (UKF) the angular velocity estimation is performed based on analysis of nonlinear gyroscope inertial measurement model. In [2] M. Dehghani showed that the fusion data of gyroscope-free inertial system, stereo cameras and low-cost gyroscope improve the robustness and accuracy of navigation. Lei L. in [3] implemented an Adaptive Kalman Filter based on a nine-accelerometer configuration for angular velocity estimation.

In [4-5] Extended Kalman Filter (EKF) is implemented. Here the attitude quaternions is determined by measurements from inertial magnetic sensors processing. In [6] S. Bras presented attitude estimation of rigid body by combining rate gyros and pan-tilt camera where the image of planer scenes is required.

Inertial sensors combined with Kalman Filter are quite popular solution. The conventional cameras are used as aid sensors also. In addition, it is expected that conventional cameras can be used to replace gyroscopes or other sensor for angular velocity totally. V. Anirudha in [7] proposed the develop of an algorithm for real-time rotation estimation by means of live video feed, where the angle rotated is calculated applying thresholding techniques, blob identification, and centroid detection and object tracking. Y. Zhang in [8] considers optical flow vector of pixels combined with a polynomial system, which is obtained by using the rigid body velocity equations and the pinhole camera model. The angular velocity of the camera is determined by solving the polynomial system. M. Gardner in [9] estimates the instantaneous position, orientation, velocity, and angular velocity of an object in a free fly by means of two high speed stereo vision cameras and EKF. However, the results can be inaccurate due to the air drag and the Magnus effect.

Nowadays, it is seen that cameras started to be used in space applications as aid sensors. This is reasonable since that star trackers and accurate gyroscope are expensive and cause that projects for small satellites is not affordable. This can be considered as one of the main reasons to use cameras as low cost sensor for space application. Kim T. in [10] used the MEMS camera module, installed on a nanosatellite, and where the angular velocity vector of the camera is provided and combined with the data given by the gyroscope into the EKF.

Volpe R. in [11] considered a scenery of a docking maneuver, where the chaser satellite equipped with a distance sensor and a camera. The measurements are provided at a certain time to be applied into UKF, whereby the relative position, attitude, linear and angular velocity are predicted. In [12] C. Pirat studied a solution for rendezvous and docking system for CubeSats, where a single camera is installed on the chaser satellite and Light-Emitting Diodes (LEDs) are installed on the target object. The proposed solution analyzes LEDs signals variations from Sun reflections, and solve the non-linear measurement equations based on the vision measurement model, rotation and translation dynamic.

The purpose of this work is the estimation of the object attitude and the angular velocity at the same time by using quaternions and by means of a conventional low-cost camera without any additional sensor. This work can also be applied for short-distance maneuvers, such as rendezvous and docking operations.

The section 1 elaborates the problem statement of this research, and it is explained the importance of measurement model, which is the mathematical model for the camera. In the Section 2 different measurement models are defined, and rotation quaternion is obtained.

Because the estimation of the angular velocity from consecutives rotation has low precision, in the Section 3 is shown the modeling system and the application of the Extended Kalman Filter to improve object attitude accuracy and for angular velocity estimation. Finally, in the Section 4 the results for object attitude determination angular velocity estimation are shown.

1. Problem statement

The problem of the angular velocity estimation by the image processing is considered. The source of measurements is the camera that captures the object's movement by taking photographs at a certain frequency (see Figure 1).

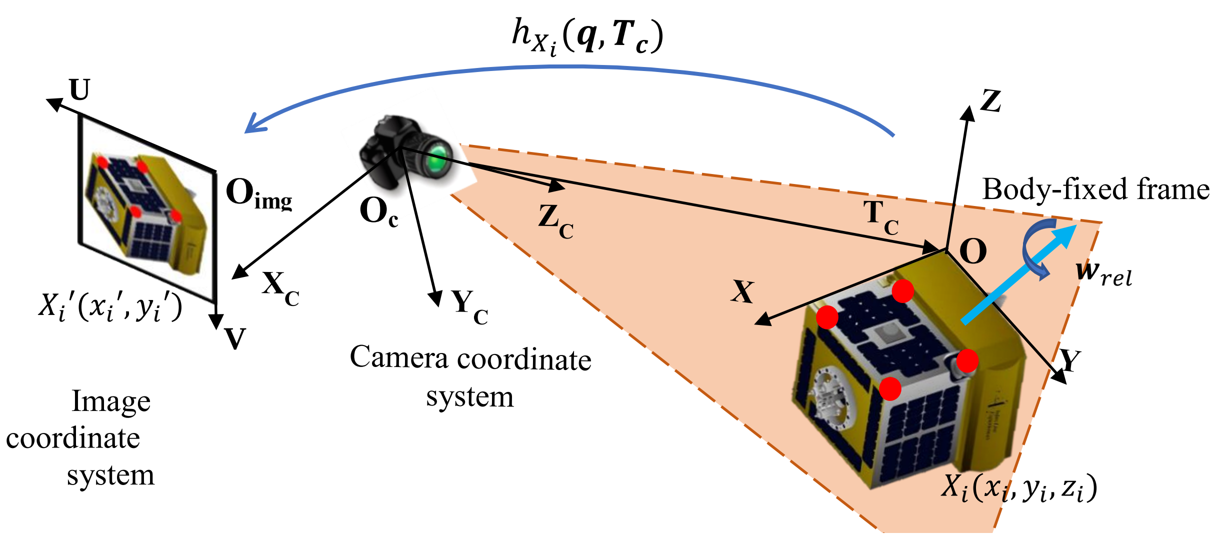


Figure 1. Diagram of the problem statement

The Cartesian coordinates systems used in this work are:

- $O_cX_cY_cZ_c$– Camera Coordinate System (CCS) 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.
- $OXYZ -$ Body-fixed Frame (BF) is placed in any location on the object in such a way that the points X_i are known with respect to BF. Its origin O is represented by the vector $\boldsymbol{T_c}$ with respect to CCS
- $O_{img}UV$ – Image Coordinate System (ICS), also known as the Image plane, is a space of 2D pixel coordinates, where each 2D pixel coordinate is the result of the conversion of points which are located on the image sensor plane in CCS, to 2D coordinate pixel. Its origin O_{img} is located on the top-left corner of the image, $O_{img}U$ extends from left to right and $O_{img}V$ extends downward, for more details about the relation between ICS, BF and the measurement model $h_{X_i}(\boldsymbol{\lambda}, \boldsymbol{T_c})$ [15].

The pictures taken by the camera are processed to estimate the angular velocity 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 points X_i, which are visualized and located in the image coordinate system (image).
- h_{X_i}: called measurement model, that performs the projection of point X_i into the Image coordinate system

The measurement model $h_{X_i}(\boldsymbol{\lambda}, \boldsymbol{T_c})$ is determined as function of the vector part $\boldsymbol{\lambda}$ of a rotation quaternion, and the translation vector $\boldsymbol{T_c}$ with respect to the camera [15].

The angular velocity $\boldsymbol{w}_{rel}$ is considered to be estimated with respect to the Body-fixed Frame, and its estimation is based on the rotation quaternion changes information during consecutives rotations as will be explained further.

2. Measurement models

The measurement model $h_{X_i}(\boldsymbol{\lambda}, \boldsymbol{T_c})$ can be named as the camera's mathematical model, this model performs the projection of a point X_i into the image, and is expressed as follows:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(\boldsymbol{\lambda}, \boldsymbol{T_c}), \tag{2.1}$$

where X_{p_i} represents the mapped point in the ICS from the BF, and is define in [15] as follows:

$$h_{X_i}(\boldsymbol{\lambda}, \boldsymbol{T_c}) = \begin{bmatrix} (x_{d_i} + s y_{d_i}) f_x + u_0 \\ y_{d_i} f_y + v_0 \end{bmatrix},$$

where $X_{d_i} = \left[x_{d_i}, y_{d_i}\right]^T$ is the distorted point coordinates because of the non-ideal lens. X_{d_i} is defined as follows:

$$X_{d_i} = \begin{bmatrix} x_{d_i} \\ y_{d_i} \end{bmatrix} = \begin{bmatrix} x_{cp_i}(1 + k_1 r_i^2 + k_2 r_i^4 + k_3 r_i^6) + 2p_1 x_{cp_i} y_{cp_i} + p_2\left(r_i^2 + 2{x_{cp_i}}^2\right) \\ y_{cp_i}(1 + k_1 r_i^2 + k_2 r_i^4 + k_3 r_i^6) + 2p_2 x_{cp_i} y_{cp_i} + p_1\left(r_i^2 + 2{y_{cp_i}}^2\right) \end{bmatrix},$$

$$r_i^2 = {x_{cp_i}}^2 + {y_{cp_i}}^2,$$

$$X_{cp_i} = \begin{bmatrix} x_{cp_i} \\ y_{cp_i} \end{bmatrix} = \begin{bmatrix} x_{c_i}/z_{c_i} \\ y_{c_i}/z_{c_i} \end{bmatrix},$$

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

where $f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2$ are constant internal parameters of the camera, and R represents the rotation matrix, which can be expressed by means of vector part $\boldsymbol{\lambda}$ of a rotation quaternion

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

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^{\boldsymbol{T}}$, $\lambda_0 = \sqrt{\mathbf{1} - |\boldsymbol{\lambda}|}$, and the rotation quaternion is $\Lambda = [\lambda_0, \lambda_1, \lambda_2, \lambda_3]^{\boldsymbol{T}}$

The camera captures the object's movement by taking photographs every period of time Δt, and at every object's movement sample the orientation quaternion is obtained by means of measurement model $h_{X_i}(\boldsymbol{\lambda}, \boldsymbol{T_c})$ and the least squares method [15] this process can be represented by the next expression

$$\begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{T_c} \end{bmatrix} = h_{opt}(\boldsymbol{\lambda}, \boldsymbol{T_c}). \tag{2.2}$$

In the Figure 2, it is shown measurement models which can be defined in the system camera - object.

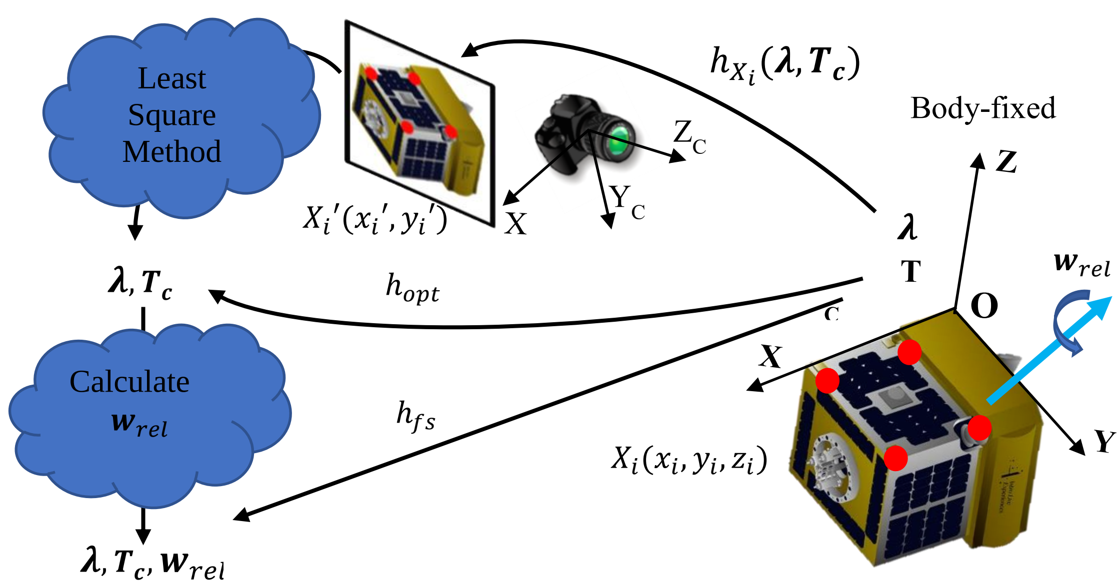


Figure 2. Measurement models

Additionally, it is understood that the angular velocity $\boldsymbol{w_{rel}}$ depends on the local rotation with respect to BF for period of time $\Delta t = t_1 - t_0$, this local rotation can be expressed by using quaternion as follows:

$$\Lambda_\alpha = {\Lambda_{t0}}^{-1} \circ \Lambda_{t1},$$

where Λ_{t0} and Λ_{t1} represent rotation quaternions the object for initial time t_0 and for the time t_1 respectively. Additionally, Λ_α is composed by the rotation angle α and its rotation axis $\boldsymbol{n}$ as shown below:

$$\Lambda_\alpha = \begin{bmatrix} \lambda_{0\alpha} \\ \boldsymbol{\lambda}_\alpha \end{bmatrix} = \begin{bmatrix} cos\frac{\alpha}{2} \\ \boldsymbol{n} sin\frac{\alpha}{2} \end{bmatrix},$$

where α and $\boldsymbol{n}$ are obtained by means of the next expressions

$$\alpha = 2arcos(q_{0\alpha}),$$

$$\boldsymbol{n} = \boldsymbol{\lambda}_\alpha sin(\alpha/2),$$

which allow the instantaneous angular velocity to be calculated from two consecutive rotation for period of time Δt as follows:

$$\boldsymbol{w_{rel}} = \frac{\alpha \boldsymbol{n}}{\Delta t}. \tag{2.3}$$

From the above mentioned, additional measurement model can be defined as follows:

$$\begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{T_c} \\ \boldsymbol{w_{rel}} \end{bmatrix} = h_{fs}(\boldsymbol{\lambda}, \boldsymbol{T_c}). \tag{2.4}$$

By using the equation (2.3), the angular velocity $\boldsymbol{w_{rel}}$ is calculated and showed in the Figure 3.

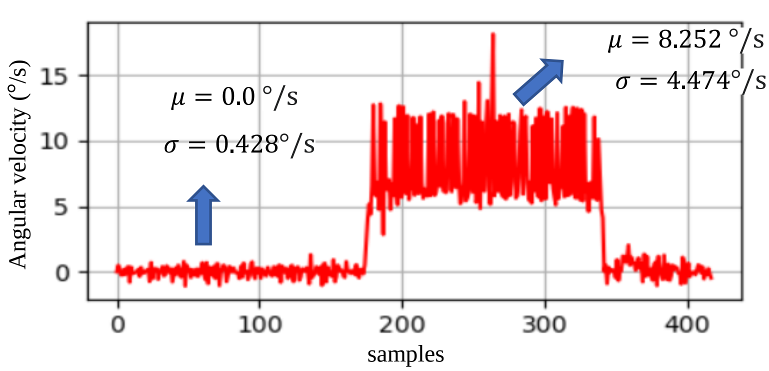


Figure 3. Angular velocity (°/s)

As it can be observed the angular velocity calculation is strongly imprecise, its standard deviation σ can reach 4.47°/s, therefore to apply Kalman filter technique for improving the angular velocity precision is to be recommended.

3. Extended Kalman Filter and system modeling

In this section is given a brief introduction to the Extended Kalman filter (EKF) [16-18], whereby it is pretended to improve the rotation quaternion and angular velocity precision taking in to account the state-space models of our system.

3.1. Extended Kalman Filter

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

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t) + \boldsymbol{w}(t), \tag{3.1}$$

$$\boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{x}(t)) + \boldsymbol{v}(t). \tag{3.2}$$

The equation (3.1) represents 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 (3.2), $\boldsymbol{z}$ is called the measurement vector and $\boldsymbol{h}$ is called the 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, $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are assumed constants.

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

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_{k-1}, \tag{3.3}$$

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

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

Considering the continuous-time nonlinear system in the equations (3.1) and (3.2) 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 (3.1) without considering the noise component $\boldsymbol{w}$ using the state vector $\hat{\boldsymbol{x}}_k^+$.

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

$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{F}_k \boldsymbol{P}_k^{+} \boldsymbol{F}_k^{T} + \boldsymbol{Q}, \tag{3.4}$$

where $\boldsymbol{F}_k$ 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 (3.2) is given by

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

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

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^{-} \boldsymbol{H}_{k+1}^{T} \boldsymbol{S}_k{}^{-1}, \tag{3.5}$$

$$\boldsymbol{S}_k = \boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1}^{-} \boldsymbol{H}_{k+1}^{T} + \boldsymbol{R},$$

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

$$\widehat{\boldsymbol{x}}_{k+1}^{+} = \widehat{\boldsymbol{x}}_{k+1}^{-} + \boldsymbol{K}_{k+1}[\boldsymbol{z}_{k+1} - \boldsymbol{h}(\widehat{\boldsymbol{x}}_{k+1}^{-})]. \tag{3.6}$$

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

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

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 (3.3) 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 (3.4).

3.2. State-space modeling

In order to apply the Extended Kalman Filter, it is required that our system be represented by means of state-space model. It means that the mathematical model, which is formed by state transition model and observation model, of our system must be defined.

It is considered to use a state-space model based on quaternion, where the state vector $\boldsymbol{x}$ is represented by:

$$\boldsymbol{x} = [\boldsymbol{\lambda}^T, \boldsymbol{T_c}^T, \boldsymbol{w}_{rel}{}^T]^T.$$

The vector $\boldsymbol{\lambda}$ represents the vector part of the rotation quaternion Λ with respect to the CCS, $\boldsymbol{T_c}$ represents the distance vector between the camera and the BF, and $\boldsymbol{w}_{rel}$ represents angular velocity with respect to the BF.

3.2.1. State transition model

From the studied system shown in the Figure 1, the angular motion of the object can be defined by means of its rotation quaternion Λ, and by its angular velocity $\boldsymbol{w}_{rel}$.

In kinematic, the continuous-time angular motion equation can be obtained using Poisson equation for relative motion using quaternions

$$\dot{\Lambda} = \frac{1}{2}\Lambda \circ \boldsymbol{w}_{rel}, \ |\Lambda| = 1, \tag{3.7}$$

where $\boldsymbol{w}_{rel} = [w_x, w_y, w_z]^T$. The equation (3.7) can be written in a matrix form

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

where $\Psi(\boldsymbol{w}_{rel})$ is defined as follows:

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

The solution of the equation (3.7) for interval of time Δt, where $\boldsymbol{w}_{rel}$ is assumed to be constant, can be written in a linear discrete-time form:

$$\Lambda_k = [I_{4x4} + \frac{1}{2}\Psi_{k-1}\Delta t]\Lambda_{k-1}, \ \Delta t = t_k - t_{k-1}. \tag{3.8}$$

It is important to mention that the dynamic differential equation for angular motion is not considered, because the inertial matrix of the object is unknown.

a) Discrete-time model

From the equation (3.8), the state transition model $\boldsymbol{f}$ can be expressed in a discrete-time form:

$$\Lambda_k = [I_{4x4} + \frac{1}{2}\Psi_{k-1}\Delta t]\Lambda_{k-1}, \tag{3.9}$$

$$\boldsymbol{T}_{\boldsymbol{c}_k} = \boldsymbol{T}_{\boldsymbol{c}_{k-1}}, \tag{3.10}$$

$$\boldsymbol{w}_{rel_k} = \boldsymbol{w}_{rel_{k-1}}. \tag{3.11}$$

Because the state transition models for $\boldsymbol{T}_{\boldsymbol{c}_k}$ and $\boldsymbol{w}_{rel_k}$ are unknown, it is convenient to consider them to be constant for small period of time Δt. From the equations (3.9), (3.10) and (3.11), let $\boldsymbol{F}$ be the linearized matrix of the state transition model defined as:

$$\boldsymbol{F} = \begin{bmatrix} \partial\boldsymbol{\lambda}_k/\partial\boldsymbol{x} \\ \partial\boldsymbol{T}_{\boldsymbol{c}_k}/\partial\boldsymbol{x} \\ \partial\boldsymbol{w}_{rel_k}/\partial\boldsymbol{x} \end{bmatrix}.$$

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

$$\frac{\partial\Lambda_k}{\partial\boldsymbol{x}} = \begin{bmatrix} \frac{\partial\Lambda_k}{\partial\boldsymbol{\lambda}} & \frac{\partial\Lambda_k}{\partial\boldsymbol{T_c}} & \frac{\partial\Lambda_k}{\partial\boldsymbol{w}_{rel}} \end{bmatrix}.$$

The equation (3.9) can be expressed as follows:

$$\Lambda_k = \Lambda_{k-1} + \frac{1}{2}\Psi_{k-1}\Lambda_{k-1}\Delta t,$$

then $\partial\Lambda_k/\partial\boldsymbol{\lambda}$ is

$$\frac{\partial\Lambda_k}{\partial\boldsymbol{\lambda}} = \frac{\partial\Lambda_{k-1}}{\partial\boldsymbol{\lambda}} + \frac{\Delta t}{2}\frac{\partial(\Psi_{k-1}\Lambda_{k-1})}{\partial\boldsymbol{\lambda}}, \tag{3.12}$$

where $\partial\Lambda_{k-1}/\partial\boldsymbol{\lambda}$ is:

$$\frac{\partial\Lambda_{k-1}}{\partial\boldsymbol{\lambda}} = \begin{bmatrix} \frac{-\lambda_1}{\lambda_0} & \frac{-\lambda_2}{\lambda_0} & \frac{-\lambda_3}{\lambda_0} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$\frac{\partial(\Psi_{k-1}\Lambda_{k-1})}{\partial\boldsymbol{\lambda}} = \begin{bmatrix} -w_x & -w_y & -w_z \\ \frac{-\lambda_1}{\lambda_0}w_x & \frac{-\lambda_2}{\lambda_0}w_x + w_z & \frac{-\lambda_3}{\lambda_0}w_x - w_y \\ \frac{-\lambda_1}{\lambda_0}w_y - w_z & \frac{-\lambda_2}{\lambda_0}w_y & \frac{-\lambda_3}{\lambda_0}w_y + w_x \\ \frac{-\lambda_1}{\lambda_0}w_z - w_y & \frac{-\lambda_2}{\lambda_0}w_z - w_x & \frac{-\lambda_3}{\lambda_0}w_z \end{bmatrix}.$$

Thus, $\partial\Lambda_k/\partial\boldsymbol{\lambda}$ from the equation (3.12) can be rewritten as follows:

$$\frac{\partial \Lambda_k}{\partial \boldsymbol{\lambda}} = \begin{bmatrix} \frac{\partial q_0}{\partial \boldsymbol{\lambda}} \\ \frac{\partial q_1}{\partial \boldsymbol{\lambda}} \\ \frac{\partial q_2}{\partial \boldsymbol{\lambda}} \\ \frac{\partial q_3}{\partial \boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda_1}{\lambda_0} - \frac{w_x \Delta t}{2} & -\frac{\lambda_2}{\lambda_0} - \frac{w_y \Delta t}{2} & -\frac{\lambda_3}{\lambda_0} - \frac{w_z \Delta t}{2} \\ 1 - \frac{\lambda_1}{\lambda_0}\frac{w_x \Delta t}{2} & -\frac{\lambda_2}{\lambda_0}\frac{w_x \Delta t}{2} + \frac{w_z \Delta t}{2} & -\frac{\lambda_3}{\lambda_0}\frac{w_x \Delta t}{2} - \frac{w_y \Delta t}{2} \\ -\frac{\lambda_1}{\lambda_0}\frac{w_y \Delta t}{2} - \frac{w_z \Delta t}{2} & 1 - \frac{\lambda_2}{\lambda_0}\frac{w_y \Delta t}{2} & -\frac{\lambda_3}{\lambda_0}\frac{w_y \Delta t}{2} + \frac{w_x \Delta t}{2} \\ -\frac{\lambda_1}{\lambda_0}\frac{w_z \Delta t}{2} - \frac{w_y \Delta t}{2} & -\frac{\lambda_2}{\lambda_0}\frac{w_z \Delta t}{2} - \frac{w_x \Delta t}{2} & 1 - \frac{\lambda_3}{\lambda_0}\frac{w_z \Delta t}{2} \end{bmatrix},$$

then the $\partial \boldsymbol{\lambda}_k / \partial \boldsymbol{\lambda}$ is:

$$\frac{\partial \boldsymbol{\lambda}_k}{\partial \boldsymbol{\lambda}} = \begin{bmatrix} \frac{\partial \lambda_1}{\partial \boldsymbol{\lambda}} \\ \frac{\partial \lambda_2}{\partial \boldsymbol{\lambda}} \\ \frac{\partial \lambda_3}{\partial \boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} 1 - \frac{\lambda_1}{\lambda_0}\frac{w_x \Delta t}{2} & -\frac{\lambda_2}{\lambda_0}\frac{w_x \Delta t}{2} + \frac{w_z \Delta t}{2} & -\frac{\lambda_3}{\lambda_0}\frac{w_x \Delta t}{2} - \frac{w_y \Delta t}{2} \\ -\frac{\lambda_1}{\lambda_0}\frac{w_y \Delta t}{2} - \frac{w_z \Delta t}{2} & 1 - \frac{\lambda_2}{\lambda_0}\frac{w_y \Delta t}{2} & -\frac{\lambda_3}{\lambda_0}\frac{w_y \Delta t}{2} + \frac{w_x \Delta t}{2} \\ -\frac{\lambda_1}{\lambda_0}\frac{w_z \Delta t}{2} - \frac{w_y \Delta t}{2} & -\frac{\lambda_2}{\lambda_0}\frac{w_z \Delta t}{2} - \frac{w_x \Delta t}{2} & 1 - \frac{\lambda_3}{\lambda_0}\frac{w_z \Delta t}{2} \end{bmatrix}.$$

Because Λ_k does not depend on $\boldsymbol{T_c}$, the expression for $\partial \Lambda_k / \partial \boldsymbol{T_c}$ is a null matrix, thus $\partial \boldsymbol{\lambda}_k / \partial \boldsymbol{T_c}$ is:

$$\frac{\partial \boldsymbol{\lambda}_k}{\partial \boldsymbol{T_c}} = 0_{3x3}.$$

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

$$\frac{\partial \Lambda_k}{\partial \boldsymbol{w}_{rel}} = \begin{bmatrix} \frac{\partial \lambda_0}{\partial \boldsymbol{w}_{rel}} \\ \frac{\partial \boldsymbol{\lambda}_k}{\partial \boldsymbol{w}_{rel}} \end{bmatrix} = \begin{bmatrix} -\lambda_1 & -\lambda_2 & -\lambda_3 \\ \lambda_0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & \lambda_0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & \lambda_0 \end{bmatrix},$$

where $\partial \boldsymbol{\lambda}_k / \partial \boldsymbol{w}_{rel}$ is

$$\frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{w}_{rel}} = \begin{bmatrix} \lambda_0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & \lambda_0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & \lambda_0 \end{bmatrix}.$$

From the equations (3.10) and (3.11), the $\partial \boldsymbol{T_c}_k / \partial \boldsymbol{x}$ and $\partial \boldsymbol{w}_{rel_k} / \partial \boldsymbol{x}$ are expressed as follows:

$$\frac{\partial \boldsymbol{T_c}_k}{\partial \boldsymbol{x}} = [0_{3x3} \quad I_3 \quad 0_{3x3}],$$

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

b) Continuous-time model

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

$$\dot{\Lambda} = \frac{1}{2}\Lambda \circ \boldsymbol{w}_{rel}, \ |\Lambda| = 1, \tag{3.13}$$

$$\dot{\boldsymbol{T}}_{\boldsymbol{c}} = \boldsymbol{0}_{3x1}, \tag{3.14}$$

$$\dot{\boldsymbol{w}}_{rel} = \boldsymbol{0}_{3x1}. \tag{3.15}$$

Let $\boldsymbol{F_{Cont}}$ be the linearized matrix of state transition model for continuous time model defined as:

$$\begin{bmatrix} \delta\dot{\boldsymbol{\lambda}} \\ \delta\dot{\boldsymbol{T}}_{\boldsymbol{c}} \\ \delta\dot{\boldsymbol{w}}_{rel} \end{bmatrix} = \boldsymbol{F_{Cont}} \begin{bmatrix} \delta\boldsymbol{\lambda} \\ \delta\boldsymbol{T}_{\boldsymbol{c}} \\ \delta\boldsymbol{w}_{rel} \end{bmatrix},$$

where

$$\boldsymbol{F_{Cont}} = \begin{bmatrix} -[\boldsymbol{w}_{rel}]_x & \boldsymbol{0_{3x3}} & 0.5\boldsymbol{I_{3x3}} \\ \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} \\ \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} \end{bmatrix}, \tag{3.16}$$

where $[\boldsymbol{w}_{rel}]_x$ is a skew-symmetric matrix defined as follows:

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

From the continuous-time model, its linearized matrix $\boldsymbol{F_{Cont}}$ shows that $\delta\dot{\boldsymbol{w}}_{rel}/\delta\boldsymbol{x}$ results in a null matrix $\boldsymbol{0_{3x9}}$, and further it is analyzed its effect in the Kalman gain matrix $\boldsymbol{K}_k$ and in the angular velocity estimation.

3.2.2. Observation model

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 which is represented by the camera.

By means of the measurement model $h_{X_i}(\boldsymbol{\lambda}, \boldsymbol{T_c})$ defined in (2.1), let the observation model $\boldsymbol{h_\lambda}$ be defined as follows:

$$\boldsymbol{h}_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \begin{bmatrix} xp_1 \\ yp_1 \\ xp_2 \\ yp_2 \\ xp_3 \\ yp_3 \\ xp_4 \\ yp_4 \end{bmatrix} = \begin{bmatrix} h_{X_1}(\boldsymbol{\lambda}, T_c) \\ h_{X_2}(\boldsymbol{\lambda}, T_c) \\ h_{X_3}(\boldsymbol{\lambda}, T_c) \\ h_{X_4}(\boldsymbol{\lambda}, T_c) \end{bmatrix}, \tag{3.17}$$

where is performed the projection of the points X_1, X_2, X_3 and X_4 into the image, the minimum amount of points is 4 [15]. Then, let $\boldsymbol{H}_{\boldsymbol{\lambda}}$ be the linearized matrix of the observation model $\boldsymbol{h}_{\boldsymbol{\lambda}}$ as follows:

$$\boldsymbol{H}_{\boldsymbol{\lambda}} = \begin{bmatrix} \partial h_{X_1}/\partial \boldsymbol{x} \\ \partial h_{X_2}/\partial \boldsymbol{x} \\ \partial h_{X_3}/\partial \boldsymbol{x} \\ \partial h_{X_4}/\partial \boldsymbol{x} \end{bmatrix}, \tag{3.18}$$

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

Because h_{X_i} does not depend on $\boldsymbol{w}_{rel}$, $\partial h_{X_i}/\partial \boldsymbol{w}_{rel}$ is a null matrix 0_{2x3}. In [15] the derivative $\partial h_{X_1}/\partial \boldsymbol{\lambda}$ is as follows:

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

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

where $\partial y_{p_i}/\partial R$ and $\partial x_{p_i}/\partial R$ are

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

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

$$D_{r_i} = 1 + k_1 r_i^2 + k_2 r_i^4 + k_3 r_i^6,$$

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

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

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

$$\frac{\partial X_{cp_i}}{\partial R} = \begin{bmatrix}\frac{\partial x_{cp_i}}{\partial R}\\ \frac{\partial y_{cp_i}}{\partial R}\end{bmatrix} = \begin{bmatrix}\frac{x_i}{z_{c_i}} & 0 & -\frac{x_{c_i}}{{z_{c_i}}^2}x_i & \frac{y_i}{z_{c_i}} & 0 & -\frac{x_{c_i}}{{z_{c_i}}^2}y_i & \frac{z_i}{z_{c_i}} & 0 & -\frac{x_{c_i}}{{z_{c_i}}^2}z_i\\ 0 & \frac{x_i}{z_{c_i}} & -\frac{y_{c_i}}{{z_{c_i}}^2}x_i & 0 & \frac{y_i}{z_{c_i}} & -\frac{y_{c_i}}{{z_{c_i}}^2}y_i & 0 & \frac{z_i}{z_{c_i}} & -\frac{y_{c_i}}{{z_{c_i}}^2}z_i\end{bmatrix},$$

and the expression for $\partial R/\partial\boldsymbol{\lambda}$ is

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

In [15] the derivative $\partial h_{X_1}/\partial\boldsymbol{T_c}$ is defined as follows:

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

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

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

$$\frac{\partial (r_i{}^2)}{\partial \boldsymbol{T_c}} = 2[x_{cp_i} \quad y_{cp_i}] \frac{\partial X_{cp_i}}{\partial \boldsymbol{T_c}},$$

$$\frac{\partial (r_i{}^4)}{\partial \boldsymbol{T_c}} = 2r_i{}^2 \frac{\partial (r_i{}^2)}{\partial \boldsymbol{T_c}} = 4 \left(x_{cp_i}{}^2 + y_{cp_i}{}^2 \right) [x_{cp_i} \quad y_{cp_i}] \frac{\partial X_{cp_i}}{\partial \boldsymbol{T_c}},$$

$$\frac{\partial (r_i{}^6)}{\partial \boldsymbol{T_c}} = 3r_i{}^4 \frac{\partial (r_i{}^2)}{\partial \boldsymbol{T_c}} = 6 \left(x_{cp_i}{}^2 + y_{cp_i}{}^2 \right)^2 [x_{cp_i} \quad y_{cp_i}] \frac{\partial X_{cp_i}}{\partial \boldsymbol{T_c}},$$

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

Because of the high non-linearity of the observation model $\boldsymbol{h_\lambda}$ the EKF is not convenient, and it would require more analysis for future work.

From the measurement model h_{opt} defined in (2.2) the observation model $\boldsymbol{h_{opt}}$ can be defined as follows:

$$\begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{T_c} \end{bmatrix} = \boldsymbol{h_{opt}}(\boldsymbol{\lambda}, \boldsymbol{T_c}). \tag{3.19}$$

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

$$\boldsymbol{H_{opt}} = \frac{\partial \boldsymbol{h_{opt}}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{x}} \\ \frac{\partial \boldsymbol{T_c}}{\partial \boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I_3} & \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} \\ \boldsymbol{0_{3x3}} & \boldsymbol{I_3} & \boldsymbol{0_{3x3}} \end{bmatrix}, \tag{3.20}$$

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

Below, it is analyzed the effect of missing angular velocity measurement in the Kalman gain matrix $\boldsymbol{K}_k$, Let the continuous-time system for angular motion be defined as:

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

$$\dot{\boldsymbol{w}}_{rel} = \boldsymbol{0}_{3x1}.$$

From the expression (3.16), let us define the state-transition model

$$\boldsymbol{F}_k = \begin{bmatrix} -[\boldsymbol{w}_{rel}]_x & 0.5\boldsymbol{I_{3x3}} \\ \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} \end{bmatrix},$$

and the observation model

$$\boldsymbol{H}_k = [\boldsymbol{I}_{\boldsymbol{3x3}} \quad \boldsymbol{0}_{\boldsymbol{3x3}}],$$

where only rotation quaternion measurement is available.

Let us define the covariance of the process noise

$$\boldsymbol{Q} = \begin{bmatrix} Q1 & Q2 \\ Q3 & Q4 \end{bmatrix},$$

where $Q1$, $Q2$, $Q3$, and $Q4$ are 3x3 matrices. The covariance of the observation noise $\boldsymbol{R}$ is a 3x3 matrix. The matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are considered to be constants.

The error covariance matrix is

$$\boldsymbol{P}_k^+ = \begin{bmatrix} P1 & P2 \\ P3 & P4 \end{bmatrix},$$

where $P1$, $P2$, $P3$, and $P4$ are 3x3 matrices.

From the expression (3.4), is obtained the Predicted error covariance $\boldsymbol{P}_{k+1}^-$

$$\boldsymbol{P}_{k+1}^- = \begin{bmatrix} \Upsilon & \boldsymbol{0}_{\boldsymbol{3x3}} \\ \boldsymbol{0}_{\boldsymbol{3x3}} & \boldsymbol{0}_{\boldsymbol{3x3}} \end{bmatrix} + \boldsymbol{Q},$$

where

$$\Upsilon = -[\boldsymbol{w}_{rel}]_x(P1\,[\boldsymbol{w}_{rel}]_x + 0.5P2) + 0.5(P3[\boldsymbol{w}_{rel}]_x + 0.5P4)$$

From the expression (3.5) Kalman gain matrix $\boldsymbol{K}_k$ is calculated

$$\boldsymbol{K}_k = \begin{bmatrix} \Upsilon\boldsymbol{S}_k{}^{-1} \\ \boldsymbol{0}_{\boldsymbol{3x3}} \end{bmatrix} + \begin{bmatrix} Q1\boldsymbol{S}_k{}^{-1} \\ Q3\boldsymbol{S}_k{}^{-1} \end{bmatrix},$$

$$\boldsymbol{S}_k = \Upsilon + \ Q1 + \boldsymbol{R}.$$

From the previous results are deducted that the angular velocity estimation depends on $Q3$, $Q1$ and $\boldsymbol{R}$.

The matrix $\boldsymbol{R}$ can be determined by performing measurements. The covariance of the process noise $\boldsymbol{Q}$, in contrast, it requires a tuning process for each element which is highly complex. To avoid this complexity the number of elements is reduced by considering $\boldsymbol{Q}$ to be diagonal. However, a diagonal matrix $\boldsymbol{Q}$ turns the submatrix $Q3$ to be a null matrix, in consequence, the matrix it $\boldsymbol{K}_k$ keep the angular velocity estimation constant.

As an alternative solution, in order to use a diagonal matrix $\boldsymbol{Q}$, consist of sending angular velocity information through the measurements into the Kalman filter. The angular velocity can be calculated by the formula (2.3).

As it has been mentioned, to add angular velocity measurements to the observation model is required. Thus, from the measurement model h_{fs}, defined in the equation (2.4), the observation model $\boldsymbol{h}_{fs}$ can be defined as follows:

$$\begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{T}_c \\ \boldsymbol{w}_{rel} \end{bmatrix} = \boldsymbol{h}_{fs}(\boldsymbol{\lambda}, \boldsymbol{T}_c, \boldsymbol{w}_{rel}). \tag{3.21}$$

Let $\boldsymbol{H}_{fs}$ be the linearized matrix of the observation model $\boldsymbol{h}_{fs}$ defined as:

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

It can be noticed that the observation model $\boldsymbol{h}_{opt}$ has the advantage of being represented by a linear function $\boldsymbol{H}_{opt}$ in the equation (3.20). However, for measuring $\boldsymbol{h}_{opt}$ in addition to point detection, it requires more computational time because of the Least Square Method, see Figure 4. The observation model $\boldsymbol{h}_{\lambda}$, in contrast, does only require point detection, but it is strongly nonlinear.

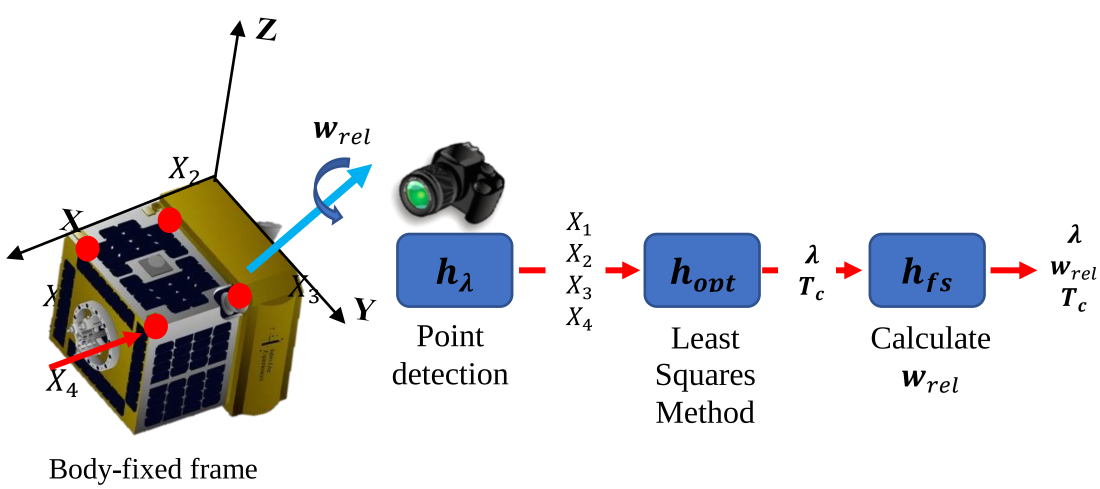


Figure 4. Observation models

It is important to mention that in discrete-time system and in continuous-time system, the observation models are in discrete-time form, because the measurements, in most physical continuous-time system, are frequently taken in a discrete form. Thus, the observation models $\boldsymbol{h}_{\lambda}$, $\boldsymbol{h}_{opt}$ and $\boldsymbol{h}_{fs}$ are completely suitable for the continuous-time state transition model. However, as it has been mentioned, angular velocity measurements for the continuous-time is required (3.1). Thus, the observation model $\boldsymbol{h_{fs}}$ is suitable for the continuous-time model (3.13).

With regard to discrete-time model, its linearized matrix $\boldsymbol{F_{Disc}}$ 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 observation models $\boldsymbol{h_{opt}}$ and $\boldsymbol{h_{fs}}$.

4. Experiment and results

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

The facilities are shown in the Figure 5 , where Aruco marker [13-14] is used to allow to establish the correspondence 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 [15], is used to capture the Aruco marker's movement every period of time Δt, where $\Delta t = 1/15$ seconds.

In order to estimate the orientation quaternion and the angular velocity of the Aruco marker, the EKF is implemented in according to the section 3.

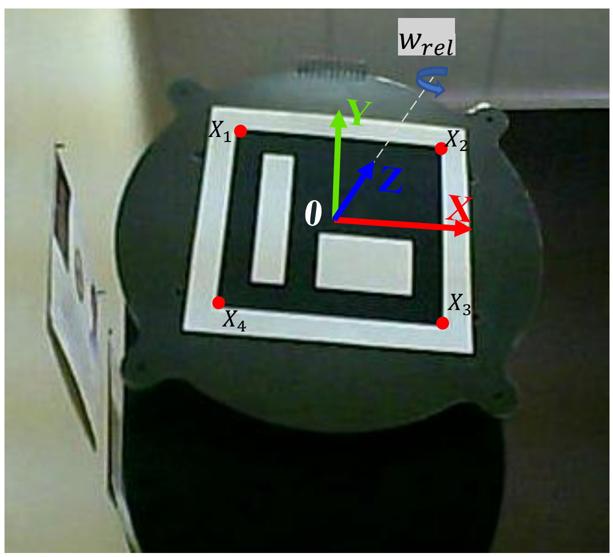


Figure 5. Rotary table rotates on the axis-Z

Let $\boldsymbol{x} = [\boldsymbol{\lambda}^T, \boldsymbol{w}_{rel}{}^T]^T$ be the state vector of the continuous-time state space model where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^{\boldsymbol{T}}$ is the vector part of orientation quaternion $\boldsymbol{\Lambda}$**.** This quaternion represents the Aruco marker's attitude with respect to the camera.

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

The process model is represented by the equations (3.13) and (3.15). The observation model is based on the equation (3.21) and is defined as follows:

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{w}_{rel} \end{bmatrix} = \boldsymbol{h}_{\boldsymbol{fs}}(\boldsymbol{\lambda}, \boldsymbol{w}_{rel}).$$

The translation vector $\boldsymbol{T_c}$ in equation (3.10) is not taken into account because the angular motion of the object is analyzed.

The process model integration is performed every period of time Δt by means of the Runge-Kutta 4$^{\text{th}}$ order method.

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 work. This GUI is implemented in order to fine-tune the covariance matrix. The covariance matrix of the process noise $\boldsymbol{Q}$ is started with a diagonal matrix with values equal to 1e-8, then by means of the mentioned GUI, the matrix components are tuned obtaining the next matrix $\boldsymbol{Q}$ as follows:

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

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. On the other hand, the covariance matrix of the observation noise $\boldsymbol{R}$ was determined by performing measurements when the rotating table is static,

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

where σ_{r1}^2=5.4e-9, σ_{r2}^2=1e-8, σ_{r3}^2=1e-7, σ_{r4}^2=1e-4, σ_{r5}^2=3e-4, σ_{r6}^2=5e-5.

Similarly, the state vector **x** is also used for the discrete-time model, its process model is represented by the equation (3.9) and (3.11). The observation model is based on the equation (3.19) as follows:

$$\boldsymbol{\lambda} = \boldsymbol{h}_{\boldsymbol{opt}}(\boldsymbol{\lambda}).$$

The translation vector $\boldsymbol{T_c}$ from equation (3.14) is not taken into account. The covariance matrix of the process noise $\boldsymbol{Q_k}$ is determined by means of tuning process:

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

where $\sigma_{q_k1}^2$=1e-8, $\sigma_{q_k2}^2$=5e-8, $\sigma_{q_k3}^2$=1e-8, $\sigma_{q_k4}^2$=6.29e-6, $\sigma_{q_k5}^2$=4.5e-6, $\sigma_{q_k6}^2$=2.47e-6. The covariance of the observation noise $\boldsymbol{R_k}$ is

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

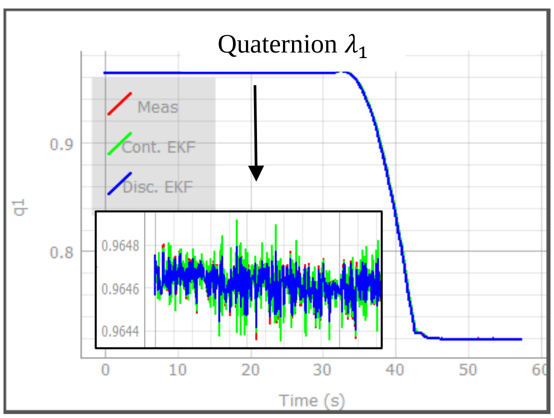
where $\sigma_{r_k1}^2$=5.4e-9, $\sigma_{r_k2}^2$=1e-8, $\sigma_{r_k3}^2$=1e-7.

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 work. This GUI is implemented in order to fine-tune the covariance matrix.

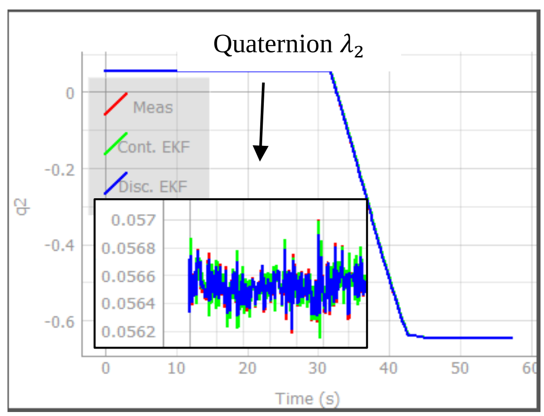
An experiment has been performed where the rotating table rotates in 90° around the axis-Z, see Figure 5. The next pictures shown three graphics defined as follows:

- 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.

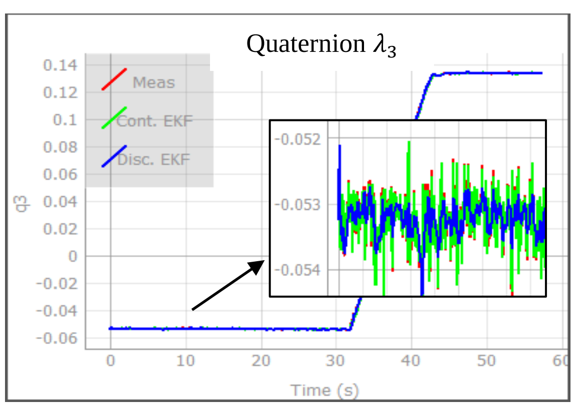
The components of the vector part $\boldsymbol{\lambda}$ of the rotation quaternion are shown in the Figure 6.



(a) Quaternion component λ_1



(b) Quaternion component λ_2



(c) Quaternion component λ_3

Figure 6. Components of the vector part of the unit quaternion

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

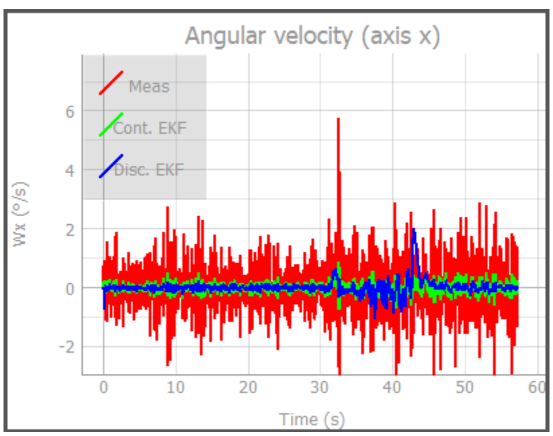
It can be seen that as for the orientation, which is determined by the quaternions, there was no significant improvement, this is because the measurement models $\boldsymbol{h_{fs}}$ and $\boldsymbol{h_{opt}}$ are already accurate for orientation determination.

Table 1

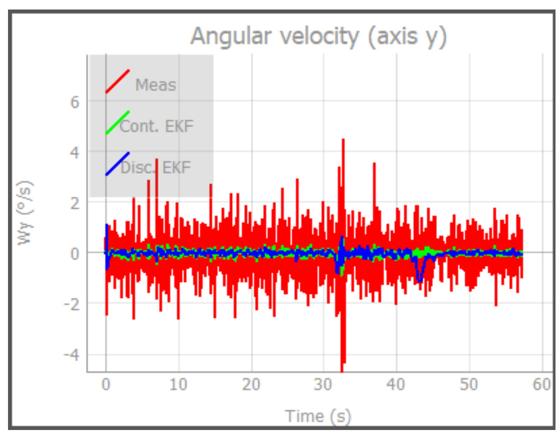
Quaternion measurements for the first 30 seconds for a static rotating table

	Mean (μ)			Standard deviation (σ)		
	μ_{meas}	$\mu_{Cont.EKF}$	$\mu_{Disc.EKF}$	σ_{meas}	$\sigma_{Cont.EKF}$	$\sigma_{Disc.EKF}$
$\boldsymbol{\lambda_1}$	0.9646	0.9646	0.9646	0.00009	0.00009	0.000069
$\boldsymbol{\lambda_2}$	0.0566	0.0566	0.0566	0.0001	0.0001	0.0001
$\boldsymbol{\lambda_3}$	-0.0533	-0.0533	-0.0533	0.0004	0.0004	0.0002

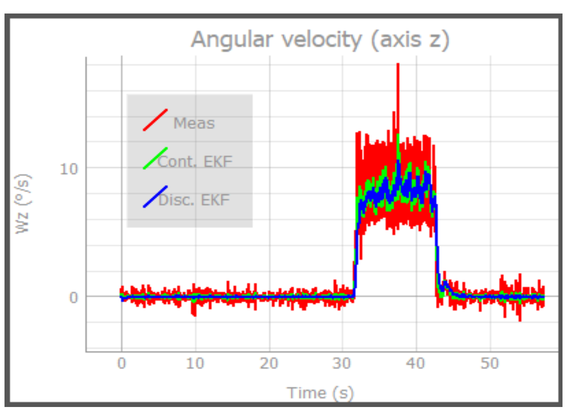
In the Figure 7, the results for the angular velocity estimation are shown. It can be noticed that there is a remarkable increase in precision.



(a) Angular velocity along the axis-X



(b) Angular velocity along the axis-Y



(c) Angular velocity along the axis-Z

Figure 7. Angular velocity with respect to the coordinate system OXYZ

In the Table 2, 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 70.3%. 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	0.00	0.0019	-0.001	0.84	0.15	0.086	82.1	89.7
w_y	0.00	-0.0012	-0.004	0.99	0.13	0.113	86.8	88.5
w_z	0.00	-0.0016	-0.005	0.40	0.12	0.050	70.3	87.5

In the Figure 8 the angle of rotation around the axis-Z is shown, and as it is expected the estimated one is close to 90° with an error of not more than 0.1°.

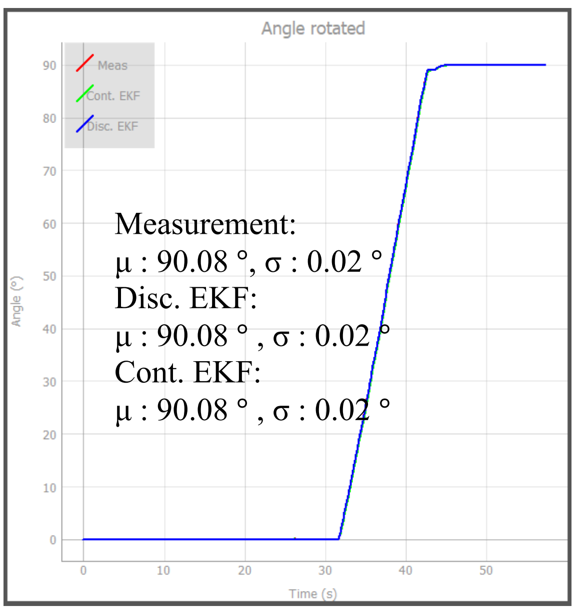


Figure 8. Angle rotated (°) around the axis-Z

The rotation angle is not calculated by angular velocity integration, but it is measured taking into account the first rotation quaternion to the current rotation quaternion. As for the rotation quaternion, the angle rotated there is no significant improvement because the measurement models $\boldsymbol{h}_{fs}$ and $\boldsymbol{h}_{opt}$ are already accurate.

Conclusion

This work is dedicated to the problem of estimating the orientation quaternion of an object and its angular velocity at the same time by image processing. The apporach based on the integration of the measurement model, adapted for the use of quaternions, to the system camera – object, where the the EKF is implemented for the orientation quaternion and angular velocity estimation.

The EKF has been implemented taking into account the rotation quaternion and angular velocity as state variables, the results showed a significatly accuracy increase for angular velocity estimation. As for the rotation quaternion, there was no significant improvement, this is because the measurement is already accurate.

Reference

[1] Zhen S., Jie Y., Peng Y., Z. Cheng Angular velocity estimation in gyroscope-free inertial measurement system based on unscented kalman filter // World Congress on Intelligent Control and Automation (WCICA), Jinan, China, July, 7-9, 2010.

[2] Dehghani M., Kharrati H., Seyedarabi H., Baradarannia M. Improvement of angular velocity and position estimation in gyro-free inertial navigation based on vision aid equipment // IET Computer Vision, 2018, Vol. 12, No. 3, P. 261-275.

[3] Lei L., Shuguang S., Zijian Ch., Gang J., Mai J. Application for solving angular velocity with adaptive kalman filtering in GFSINS // Shenzhen, China, June, 6-8, 2011.

[4] Szczęsna, A., Pruszowski, P. Model-based extended quaternion Kalman filter to inertial orientation tracking of arbitrary kinematic chains // SpringerPlus 5, November, 14, 2016.

[5] Sabatini, A.M. Kalman-Filter-Based Orientation Determination Using Inertial/Magnetic Sensors: Observability Analysis and Performance Evaluation // Sensors 2011, 11, 9182-9206.

[6] Bras S., Cunha R., Vasconcelos J. F., Silvestre C., Oliveira P. A Nonlinear Attitude Observer Based on Active Vision and Inertial Measurements // IEEE Transactions on Robotics, 2011, Vol. 27, No. 4, P. 664-667.

[7] Anirudha V., Suraj P., Ravi K., Somanathan A., An Improved Real-Time Approach for Video based Angular Motion Detection and Measurement // IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, India, December, 13-15, 2018.

[8] Zhang Y., Fathian K., Gans N., VEst: An Efficient Solution to the Camera Velocity Estimation from Minimal Feature Points // American Control

Conference (ACC), Denver, USA, USA, July, 1-3, 2020.

[9] Gardner M., Jia Y. Estimating the Linear and Angular Velocities of an Object in Free Flight // IEEE International Conference on Robotics and Automation, Montreal, Canada, May, 20-24, 2019.

[10] Kim T., Kim Y., Bang H. Improvement of the attitude determination accuracy for nanosatellites using a camera sensor // European Navigation Conference (ENC), Lausanne, Switzerland, May, 9-12, 2017.

[11] Volpe R., Palmerini G. B., Sabatini M. A passive camera based determination of a non-cooperative and unknown satellite's pose and shape // Acta Astronautica, 2018, Vol. 151, P. 805-817.

[12] Pirat C., Ankersen F., Walker R., Gass V. Vision Based Navigation for Autonomous Cooperative Docking of CubeSats // Acta Astronautica, 2018, Vol. 146, P. 418-434.

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

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

[15] Molina Saqui J.C., Tkachev S.S., Testbench calibration technique for testing attitude determination algorithms by video processing, KIAM Preprint, 2020, № 73, 32 p.

[16] Simon D. Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches, 1st ed, Wiley-Interscience, 2006, 552c.

[17] Mashtakov Y., Ovchinnikov M, Woske F., Rievers B. Attitude determination and control system design for gravity recovery mission like GRACE // Acta astronautica, 2020, Vol. 173, P. 172-182.

[18] Ivanov D., Ovchinnikov M. Analytical study of microsatellite attitude determination algorithms //Acta astronautica, 2015, Vol. 116, P. 339-348.

Contents