

# Globally Stable Velocity Estimation using normalized Velocity Measurement

Elias Bjørne\* Edmund F. Brekke\* Torleiv H. Bryne\* Jeff Delaune † Tor Arne Johansen\*

**Abstract**—The problem of estimating velocity from a monocular camera and calibrated IMU measurements is revisited. For the presented setup, it is assumed that normalized velocity measurement is available from the camera. By applying results from nonlinear observer theory we are able to present a velocity estimation with proven global stability, with defined conditions for convergence, and without the need of observing features for several camera frames. Several nonlinear methods are compared to each other, also against an extended Kalman filter (EKF), where the robustness of the nonlinear methods compared to the EKF are demonstrated in simulations and experiments.

**Index Terms**—Nonlinear observer, course measurement, bearing, simultaneous localization and mapping, ego-motion estimation, sensor fusion

## I. INTRODUCTION

Vision-based navigation is an important topic in robotics. With the reduced size, weight, power and money cost of modern digital cameras, they are a key navigation component in many robotics systems. In the literature the techniques are often referred to as real-time structure from motion (SFM), visual odometry (VO) or visual/bearing only simultaneous localization and mapping (SLAM). Several methods have been developed over the years, and the methods are often divided into graph SLAM and SLAM filters, where the graph SLAM methods are based on nonlinear optimization [1]–[3] also called bundle adjustment (BA). The most efficient approaches store key-frames, making the optimization problem sparse, and hence they are solvable in real time. The SLAM filters are often based on EKF [4], where the estimates are stored in the states and the co-variance, making the computational load grow quadratic with regards to the number of features. A further discussion of the pros and cons between the methods can be seen in [5].

However, an inherent problem when using monocular camera is that the metric scale of the world is not observable. The scale is either given during the initialization through stereo vision, laser scanner, IMU information or by having known features or tags at the initialization, or it is just neglected. For

some applications, especially long time navigation, scale drift can be a limiting factor. There are also several Visual-Inertial Odometry (VIO) methods that are able to estimate the scale with the IMU information [6], [7]. Methods for estimating scale from a camera world to the real world has also been presented in [8], [9], where an EKF setup estimates the scale by comparing the pose measurements from a camera system to IMU data. These methods then need the scale estimate to be close to the real scale at the initialization to guarantee convergence.

Initialization procedures have also been presented, with different kind of assumptions available, such as in [10]–[13]. In [10] a throw and go procedure was demonstrated, however as the scale is estimated through an EKF scheme there is no guarantee of convergence, and the scale is therefore initialized close to the true scale. In [11] a similar procedure was demonstrated for SVO+MSF, where the scale is acquired from an altimeter. In [12] and [13], closed form and optimization schemes are developed, that can initialize the scale and gravity from IMU and bearing measurements. This however relies on features being detected for several time steps or in the case of [12], that the monocular SLAM is initialized. They also lack any condition that could guarantee good initialization.

In [14] an ego-motion estimation using a continuous homography scheme was presented, where estimation of metric velocity was performed with a calibrated IMU and pitch roll estimates. This work differentiates itself from the most common implementations from visual navigation in two ways. As it relies on the homography transform, it only requires matching of features between two images. Thus the requirement of observing features for consecutive images is no longer necessary. The other benefit of the system [14] is inherited from its application of nonlinear observer techniques presented in [15], [16]. Thus the observer acquires global convergence under a defined persistent of excitation (PE) condition. In addition, the system with predefined trajectory can also be tuned similarly to a second order differential equation so that convergence rate and frequency response can be defined for the observer.

However the main drawback for the setup [14], is the necessity of observing a flat surface in order to utilize the homography transformation, which is an assumption that is violated for many applications.

This has given the motivation of developing an observer which only relies on normalized velocity. Thus it would no longer be necessary to observe a flat surface. In fact, this enables the ego motion estimation to be applicable to

\*Elias Bjørne, Edmund F. Brekke, Torleiv H. Bryne and Tor Arne Johansen are with the Center for Autonomous Marine Operations and Systems (NTNU-AMOS) and Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway. (Email: eliasseb@ntnu.no, edmund.brekke@ntnu.no, torleiv.h.bryne@ntnu.no, tor.arne.johansen@ntnu.no)

†Jeff Delaune is with the Computer Vision Group at the Jet Propulsion Laboratory, NASA / California Institute of Technology. (Email: Jeff.H.Delaune@jpl.nasa.gov)

any VO, Visual SLAM or SFM method able to provide a normalized camera velocity. Methods includes two view methods such as, essential matrix [17], two point [3], three point [18] or homography [14], or a combination of these as shown in [2], where homography and the 5 point algorithm is used in parallel. Or VO methods such as SVO [1] and SLAM [2]. Such a method could either be used to initialize the scale of these methods, work in parallel as scale estimator to ensure drift free scale or it could work as a velocity measurement in itself.

### A. Contribution

An observer able to estimate velocity with similar sensor setup as in [14], was presented in [19]. The only difference was that a normalized velocity was assumed to come from the camera instead of a homography transform. Thus globally stable velocity estimation with defined PE condition, and without the need of observing features for several images is possible without the need of observing a flat surface. This article builds on the article [19], where we compare the novel observer to other similar observers, and how they perform when used for the ego-motion estimation.

There are several observer able to perform this estimation, since the ego-motion estimation with normalized velocity measurement is mathematically identical to the problem of estimating the distance to a landmark assuming velocity and bearing measurements are available. This implies that the ego motion estimation assuming normalized velocity measurements can be performed by several observers, but whom originally was designed for the later problem [19]–[22]. In addition, the observers will keep the same stability results. In fact all the nonlinear observers have the same persistence of excitation condition; that the the velocity and acceleration of the vehicle is not parallel. Thus globally stable velocity observers with defined PE conditions guaranteeing convergence is possible without any initialized VO method, observable plane or the need of observing features for several images.

The different observers are compared qualitatively, in simulations and on experimental data. It is also shown how the novel observer presented in [19], can tune its convergence rate. They are also compared to an EKF and the homography based observer [14]. The observers are able to estimate the velocity accurately, where the observer [19] has the best performance in simulations and on experiments. It is also shown how the nonlinear observers have more robust behavior than the EKF.

## II. NOTATION AND PRELIMINARIES

### A. Notation

Scalars are in lower case  $a, x, \omega$ ; vectors are lower case bold  $\mathbf{a}, \mathbf{x}, \boldsymbol{\omega}$ ; sets are upper case  $A, X, \Omega$ ; matrices are bold upper case  $\mathbf{A}, \mathbf{X}, \mathbf{\Omega}$ . The 0 denotes the scalar zero, while  $\mathbf{0}$  is the matrix zero where dimensions are implicitly given by the context. The matrix  $\mathbf{I}$  is the identity matrix, and size is given by context. The accents  $\hat{\bullet}, \tilde{\bullet}, \dot{\bullet}, \bar{\bullet}, \underline{\bullet}$ , denotes estimate, estimate error, time derivative, upper bound and lower bound

respectively. Some common mathematical expressions which will be used are: The Euclidean norm for vectors and Frobenius norm for matrices,  $\|\bullet\|$ , absolute value,  $|\bullet|$  and the transpose,  $\bullet^\top$ . The representation of index sets will be done with  $\{1, \dots, n\} = \{x \in \mathbb{Z} | x \leq n\}$ .

A vector is said to be on the unit sphere  $\mathbf{u}_\bullet \in \mathcal{S}^3 = \{\mathbf{u}_\bullet \in \mathbb{R}^3 | \|\mathbf{u}_\bullet\| = 1\}$ . A vector can be represented in different coordinate systems, the representation is denoted with the superscripts  $\bullet^b, \bullet^c, \bullet^n$  which represents the body-fixed, camera-fixed and earth-fixed (inertial) coordinate systems, and will be called body-frame, camera-frame and inertial frame. Lowercase will denote the indices of a landmark, vector or matrix  $\bullet_i$  and  $\bullet_{ij}$ .

### B. Rotation representation

Rotation is the attitude change between two coordinate systems, and a rotation from coordinate system  $b$  to  $n$  is denoted with subscript  $\bullet_{nb}$ . This can be represented with a rotation matrix

$$\mathcal{R}_{nb} \in \{\mathbb{R}^{3 \times 3} | \mathcal{R}_{nb} \mathcal{R}_{nb}^\top = \mathbf{I}, \det(\mathcal{R}_{nb}) = 1\}$$

which means  $\mathcal{R}_{nb} \in SO(3)$ . The rotational vector transformation is calculated with the rotation matrix  $\mathbf{x}^n = \mathcal{R}_{nb} \mathbf{x}^b$ . The cross product is presented in matrix form  $\mathcal{S}(\mathbf{x}) \mathbf{y} = \mathbf{x} \times \mathbf{y}$ , where  $\mathcal{S}(\bullet)$  is a skew-symmetric matrix

$$\mathcal{S}(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

which gives  $\mathcal{S}(\bullet) = -\mathcal{S}(\bullet)^\top$ ,  $\mathbf{x}^\top \mathcal{S}(\bullet) \mathbf{x} = 0$ ,  $\mathbf{x}^\top \mathcal{S}(\mathbf{x}) = \mathbf{0}$ ,  $\forall \mathbf{x}, \mathcal{S}(\mathbf{x}) \mathbf{y} = -\mathcal{S}(\mathbf{y}) \mathbf{x}$ . Moreover the cross-product gives the difference in angle-axis between two vectors

$$\mathcal{S}(\mathbf{x}) \mathbf{y} = \|\mathbf{x}\| \|\mathbf{y}\| \sin(\theta) \mathbf{u} \quad (1)$$

where  $\theta$  is the angle between the vectors, and  $\mathbf{u}$  is the axis of the rotation with direction following the right hand rule of the cross product, making it orthogonal to the two vectors. Let the rotation matrix  $\mathcal{R}_{nb}$  denote the rotation from the body frame to the earth-fixed frame. The dynamics of the of the rotation matrix is described by

$$\dot{\mathcal{R}}_{nb} = \mathcal{R}_{nb} \mathcal{S}(\boldsymbol{\omega}) \quad (2)$$

where  $\boldsymbol{\omega} = \boldsymbol{\omega}_{nb}^b$  is the angular velocity of the frame  $b$  relative to  $n$  decomposed in  $b$ .

For discrete propagation, the Rodriguez formula can be applied as in [23]

$$\tilde{\boldsymbol{\omega}} = \int_t^{t+\Delta t} \boldsymbol{\omega}(s) ds, \quad \bar{\omega} = \|\tilde{\boldsymbol{\omega}}\|, \quad \mathbf{u}_\omega = \frac{\tilde{\boldsymbol{\omega}}}{\|\tilde{\boldsymbol{\omega}}\|} \quad (3)$$

$$\mathcal{R}_{nb}(t + \Delta t) = \mathcal{R}_{nb} \mathcal{R}(\tilde{\boldsymbol{\omega}}) \quad (4)$$

$$\mathcal{R}(\tilde{\boldsymbol{\omega}}) = (\mathbf{I} + \sin(\bar{\omega}) \mathcal{S}(\mathbf{u}_\omega) + (1 - \cos(\bar{\omega})) \mathcal{S}(\mathbf{u}_\omega)^2) \quad (5)$$

where  $\tilde{\boldsymbol{\omega}}$  is the attitude increment.

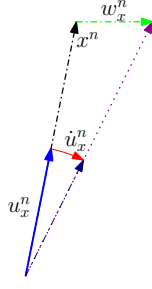


Fig. 1: Vector dynamics in inertial-frame

### C. Unit Vector dynamics

As mentioned, directional measurements of a vector  $\mathbf{x}$  can be represented as a unit vector  $\mathbf{u}_x = \frac{\mathbf{x}}{\|\mathbf{x}\|}$ . These vectors can either be measured in the inertial-frame  $\mathbf{u}_x^n$  or body frame  $\mathbf{u}_x^b$ . The vector corresponding to the directional measurement will have a magnitude, and a time derivative in the inertial-frame

$$z = \|\mathbf{x}^n\| \quad \dot{\mathbf{x}}^n = \mathbf{w}_x^n \quad (6)$$

By combining (2) and (6) we can get the time derivatives of the vector magnitude and unit vector in inertial- and body-frame, visualized in Figure 1,

$$\begin{aligned} z^2 &= (\mathbf{x})^\top \mathbf{x} \\ 2z\dot{z} &= 2(\mathbf{x})^\top \dot{\mathbf{x}} \\ \dot{z} &= (\mathbf{u}_x^n)^\top \mathbf{w}_x^n = (\mathbf{u}_x^b)^\top \mathbf{w}_x^b \end{aligned} \quad (7)$$

These results are then used for deriving the time derivative of the unit vector

$$\begin{aligned} \dot{\mathbf{u}}_x^n &= \frac{\dot{\mathbf{x}}^n}{z} - \frac{\mathbf{x}^n}{z^2} \dot{z} \\ &= \frac{\mathbf{w}_x^n}{z} - \frac{\mathbf{u}_x^n}{z} (\mathbf{u}_x^n)^\top \mathbf{w}_x^n \\ &= \frac{1}{z} (\mathbf{I} - \mathbf{u}_x^n (\mathbf{u}_x^n)^\top) \mathbf{w}_x^n \\ \dot{\mathbf{u}}_x^n &= -\frac{1}{z} \mathbf{S}(\mathbf{u}_x^n)^2 \mathbf{w}_x^n \end{aligned}$$

where it is clear that the unit vector will be maintained on the unit sphere since  $(\mathbf{u}_x^n)^\top \dot{\mathbf{u}}_x^n = 0$ . For the dynamics of the unit vector in body-frame we need to take into account the rotation of the body-frame frame

$$\begin{aligned} \mathbf{x}^b &= (\mathcal{R}_{nb})^\top \mathbf{x}^n \\ &\Downarrow \\ \dot{\mathbf{x}}^b &= -\mathbf{S}(\boldsymbol{\omega}) \mathbf{x}^b + \mathbf{w}_x^b \end{aligned}$$

which gives the unit vector dynamics

$$\begin{aligned} \dot{\mathbf{u}}_x^b &= \frac{\dot{\mathbf{x}}^b}{z} - \frac{\mathbf{x}^b}{z^2} \dot{z} \\ &= \frac{-\mathbf{S}(\boldsymbol{\omega}) \mathbf{x}^b + \mathbf{w}_x^b}{z} - \frac{\mathbf{x}^b}{z^2} (\mathbf{u}_x^b)^\top \mathbf{w}_x^b \\ &= -\mathbf{S}(\boldsymbol{\omega}) \mathbf{u}_x^b + \frac{1}{z} (\mathbf{I} - \mathbf{u}_x^b (\mathbf{u}_x^b)^\top) \mathbf{w}_x^b \\ &= -\mathbf{S}(\boldsymbol{\omega}) \mathbf{u}_x^b - \frac{1}{z} \mathbf{S}(\mathbf{u}_x^b)^2 \mathbf{w}_x^b \end{aligned} \quad (8)$$

This unit vector is also maintained on the unit sphere by the same argument. The inverted magnitude  $d = \frac{1}{z}$  will have the dynamic

$$\begin{aligned} \dot{d} &= -\left(\frac{1}{z}\right)^2 \dot{z} \\ &= -d^2 (\mathbf{u}_x^n)^\top \mathbf{w}_x^n = -d^2 (\mathbf{u}_x^b)^\top \mathbf{w}_x^b \end{aligned} \quad (9)$$

where (7) is used.

### D. Unit vector error and propagation

First, we reorganize the unit vector dynamic (8) into

$$\dot{\mathbf{u}}_x^b = -\mathbf{S}(\boldsymbol{\omega}) \mathbf{u}_x^b - d_i \mathbf{S}(\mathbf{S}(\mathbf{u}_x^b) \mathbf{v}^b) \mathbf{u}_x^b \quad (10)$$

$$= -\mathbf{S}(\boldsymbol{\omega} + d_i \mathbf{S}(\mathbf{S}(\mathbf{u}_x^b) \mathbf{v}^b)) \mathbf{u}_x^b \quad (11)$$

We then see that we also can use Rodriguez formula (5) to propagate the unit vector, by integrating

$$\ddot{\boldsymbol{\omega}} = \int_t^{t+\Delta t} (\boldsymbol{\omega} + d_i \mathbf{S}(\mathbf{S}(\mathbf{u}_x^b) \mathbf{v}^b)) dt \quad (12)$$

$$\mathbf{u}_x^b(t + \Delta t) = \mathcal{R}(\ddot{\boldsymbol{\omega}})^\top \mathbf{u}_x^b(t) \quad (13)$$

It was also seen from the (1), that the cross-product contains the angle information between two vectors. In fact the matrix rotating a unit vector  $\hat{\mathbf{u}}_x^b$  to  $\mathbf{u}_x^b$  can be found by the closed form formula, derived from the Rodriguez formula

$$\mathbf{u}_x^b = \tilde{\mathcal{R}} \hat{\mathbf{u}}_x^b \quad (14)$$

$$\tilde{\mathbf{u}}_x^b = \mathbf{S}(\mathbf{u}_x^b) \hat{\mathbf{u}}_x^b, \quad \tilde{\mathbf{u}}^b = (\hat{\mathbf{u}}_x^b)^\top \mathbf{u}_x^b \quad (15)$$

$$\tilde{\mathcal{R}} = \mathbf{I} + \mathbf{S}(\tilde{\mathbf{u}}_x^b) + \frac{1}{1 + \tilde{\mathbf{u}}^b} \mathbf{S}(\tilde{\mathbf{u}}_x^b)^2 \quad (16)$$

where we see that the first term in the Taylor expansion of this rotation with respect to  $\tilde{\mathbf{u}}_x^b$  is

$$\mathbf{u}_x^b \approx (\mathbf{I} + \mathbf{S}(\tilde{\mathbf{u}}_x^b)) \hat{\mathbf{u}}_x^b \quad (17)$$

## III. EQUIVALENCE BETWEEN BEARING ONLY POSITION ESTIMATION AND VELOCITY ESTIMATION USING NORMALIZED VELOCITY

We first represent the ego-motion estimation similar to what was presented in [14], where the goal is to estimate the velocity of a vehicle provided that a camera and a tactical IMU with roll and pitch estimates are available. We then show that this mathematical system is identical to the problem of filtering velocity and bearing measurements to estimate the distance to a landmark, and thus we can conclude that observers designed for one of the problems is applicable to the other.

To build an observer based on normalized velocity measurements, we need to examine the unit vector kinematics of the normalized velocity. A vehicle has a position  $\mathbf{p}^n$  in the inertial frame, and moves with a velocity

$$\dot{\mathbf{p}}^n = \mathbf{v}^n \quad (18)$$

and also has the acceleration

$$\ddot{\mathbf{p}}^n = \dot{\mathbf{v}}^n = \mathbf{a}^n$$

For simplicity, we assume that the IMU measurements are transformed to the camera frame, thus the acceleration can be measured by an IMU in camera frame, provided that it has an internal gravity estimate  $\mathbf{g}^c$  from filtering

$$\mathbf{a}^c = \mathcal{R}_{nc}^\top \mathbf{a}^n = \mathbf{f}^c - \mathbf{g}^c$$

In addition, we assume that the camera provides a normalized velocity in the camera coordinates

$$\mathbf{u}_v^c = \mathcal{R}_{nc}^\top \mathbf{u}_v^n = \mathcal{R}_{nc}^\top \frac{\mathbf{v}^n}{\|\mathbf{v}^n\|}$$

and we introduce the inverted velocity magnitude  $d_v = \frac{1}{\|\mathbf{v}^n\|}$  which will have similar time derivative as (9)

$$\dot{d}_v = -d_v^2 (\mathbf{u}_v^n)^\top \mathbf{a}^n = -d_v^2 (\mathbf{u}_v^c)^\top \mathbf{a}^c$$

If we then look at the time derivative of  $\mathbf{u}_v^c$  we get

$$\begin{aligned} \dot{\mathbf{u}}_v^c &= \dot{\mathcal{R}}_{nb}^\top \mathbf{v}^n d_v + \mathcal{R}_{nc}^\top \dot{\mathbf{v}}^n d_v + \mathcal{R}_{nc}^\top \mathbf{v}^n \dot{d}_v \\ &= -S(\boldsymbol{\omega}^c) \mathcal{R}_{nb}^\top \mathbf{v}^n d_v + \mathcal{R}_{nc}^\top \mathbf{a}^n d_v - \mathcal{R}_{nc}^\top \mathbf{v}^n d_v^2 (\mathbf{u}_v^c)^\top \mathbf{a}^c \\ &= -S(\boldsymbol{\omega}^c) \mathbf{u}_v^c + \mathbf{a}^c d_v - \mathbf{v}^c d_v^2 (\mathbf{u}_v^c)^\top \mathbf{a}^c \\ &= -S(\boldsymbol{\omega}^c) \mathbf{u}_v^c + d_v (\mathbf{I} - \mathbf{u}_v^c (\mathbf{u}_v^c)^\top) \mathbf{a}^c \\ &= -S(\boldsymbol{\omega}^c) \mathbf{u}_v^c - d_v \mathbf{S}(\mathbf{u}_v^c)^2 \mathbf{a}^c \end{aligned}$$

Which means that we end up with the system

$$\dot{\mathbf{u}}_v^c = -S(\boldsymbol{\omega}^c) \mathbf{u}_v^c - d_v \mathbf{S}(\mathbf{u}_v^c)^2 \mathbf{a}^c \quad (19)$$

$$\dot{d}_v = -d_v^2 (\mathbf{u}_v^c)^\top \mathbf{a}^c \quad (20)$$

where we assume that we can measure  $\mathbf{u}_v^c$ ,  $\mathbf{a}^c$ , and  $\boldsymbol{\omega}^c$ , and we want to estimate the velocity magnitude  $\|\mathbf{v}^c\| = \frac{1}{d_v}$ .

We then want to compare this to the system with bearing and velocity measurements. The bearing measurement from a landmark are as follows

$$\mathbf{u}_\delta^c = \mathcal{R}_{nc}^\top \frac{\boldsymbol{\delta}_i^n}{\|\boldsymbol{\delta}_i^n\|} = \mathcal{R}_{nc}^\top \frac{\mathbf{p}_i^n - \mathbf{p}^n}{\|\mathbf{p}_i^n - \mathbf{p}^n\|}$$

which is related to the velocity by

$$\dot{\boldsymbol{\delta}}^n = \dot{\mathbf{p}}_i^n - \dot{\mathbf{p}}^n = -\mathbf{v}^n$$

where we use (18) and  $\dot{\mathbf{p}}_i^n = 0$ . We see that  $\boldsymbol{\delta}_i^n$  is the distance vector between the vehicle and the landmark. By repeating the steps done above, the time derivatives of  $\mathbf{u}_\delta^c$  and  $d_\delta = \frac{1}{\|\boldsymbol{\delta}_i^n\|}$  can be calculated to be

$$\dot{\mathbf{u}}_\delta^c = -S(\boldsymbol{\omega}^c) \mathbf{u}_\delta^c + d_\delta \mathbf{S}(\mathbf{u}_\delta^c)^2 \mathbf{v}^c \quad (21)$$

$$\dot{d}_\delta = d_\delta^2 (\mathbf{u}_\delta^c)^\top \mathbf{v}^c \quad (22)$$

where we have the measurements  $\mathbf{u}_\delta^c$ ,  $\boldsymbol{\omega}^c$  and  $\mathbf{v}^c$ , and we want to estimate the distance to the landmark  $\|\boldsymbol{\delta}_i^n\| = \frac{1}{d_\delta}$ . We see that the systems (19)-(20) and (21)-(22) are mathematically identical except a sign difference, since  $\dot{\mathbf{v}}^n = \mathbf{a}^n$  and  $\dot{\boldsymbol{\delta}}^n = -\mathbf{v}^n$ .

If we then want to design an observer for the ego motion estimation (19)-(20) we can use observers that are originally designed for system (21)-(22), where range to a landmark is estimated. This is an extensively studied system in the literature, and several observers have been designed. In the next section we will present the observers that have proven global stability and apply them to the ego motion estimation.

## IV. VELOCITY OBSERVERS

As stated, several observers exist to estimate the distance to a landmark provided a bearing and gyro rate and velocity measurements. However, in this section we apply them to the camera velocity estimation. Where the velocity can be estimated provided normalized velocity, acceleration and gyro rate measurement.

What the nonlinear observers tested have in common is that they have similar measurements and have proof of global or semi-global stability. Even though the observers are structured differently, with PE condition differently defined, they all have in common that they rely on the unit vector measurement to be non stationary in the inertial frame. For the ego motion estimation, this means that the PE conditions are fulfilled when the acceleration and velocity are not co linear. In addition we compare the observers to an EKF inspired by the states used in robust visual inertial odometry [7]. A qualitative comparison between the observers is presented in Table I. In the rest of this section we present the sensor setup for the ego motion estimation and present the observers that will be compared.

### A. Sensor setup

The IMU or AHRS is assumed to be of high quality with an internal filter available to calculate gravity estimates and biases. In addition a camera is available, providing a information about the velocity direction (vehicle course). The camera is calibrated, and the transformation between the camera and the IMU is found through calibration. The following measurements and known values are then:

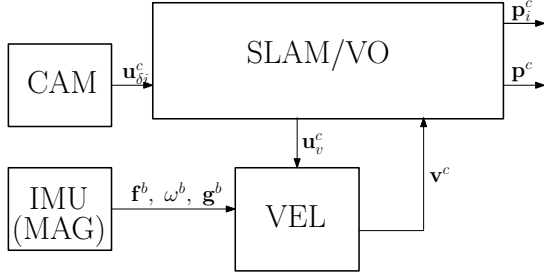
- Constant transformations  $\mathbf{p}_{cb}^c$  and  $\mathcal{R}_{cb}$  between the IMU and the camera.
- $\boldsymbol{\omega}^c = \mathcal{R}_{cb} \boldsymbol{\omega}^b$ , The gyro rate from the IMU in camera frame
- $\mathbf{u}_v^c$ , normalized camera velocity in the camera frame
- $\hat{\mathbf{a}}^c = \mathcal{R}_{cb} (\mathbf{f}^b + \mathbf{g}^b) + \mathbf{S}(\boldsymbol{\omega}^c) \mathbf{p}_{cb}^c$ , perceived acceleration provided by the IMU from the specific force and gravity estimates together with the lever arm effect.

The terms related to jerk  $\dot{\boldsymbol{\omega}}$  is neglected and assumed to be zero for this setup, as done for the sensor setup in [14].

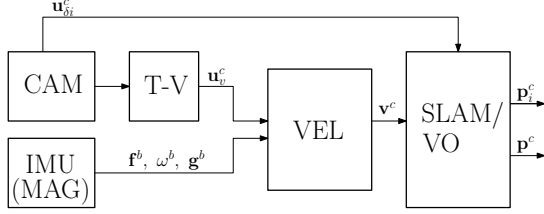
As the observers will be able to estimate velocity from only normalized velocity, the observers can be used either for initializing visual SLAM or VO methods, with the setup shown in Figure 3. The normalized velocity can then come from a scheme as described in [2], where a normalized velocity can be acquired from two-view algorithms such as homography or the 5 point algorithm [17], thus not requiring any initialization. Or in parallel with a visual SLAM or VO as in Figure 2, acquiring the normalized velocity and fusing it with the IMU data and return the metric velocity which can be used for a drift free scale estimate.

### B. Robust Extended Kalman Filter

Inspired by the work in [7], we use the robust EKF for comparison. Where the distance estimate is inverted, and the bearing and inverted range states are separated. As we assume there are no biases in the IMU, and gravity is



**Fig. 2:** Block diagram of a possible setup between the velocity estimation and a SLAM/VO, where the VEL observer gets normalized velocity from the SLAM/VO method, and fuses it with the IMU information which then can be used to maintain a drift free scale for the visual SLAM/VO.



**Fig. 3:** Block diagram a possible initialization setup, where a two-view algorithms as explained in [2], that can provide a normalized velocity without any initialization. The velocity observer can thus combine the normalized velocity with IMU data and initialize the scale of a visual SLAM/VO method, or provide velocity measurements.

assumed to be known. The states are  $\mathbf{x}_{EKF} = [(\mathbf{u}_v^c)^\top, d_v]^\top$ , where  $d_v = \frac{1}{\|\mathbf{v}^c\|}$  is the inverse metric speed, where the dynamics are (19)-(20).

Conversely to the other observers, we show the discrete time implementation for the EKF, and we use the euler method to discretize the system, however we apply the Rodriguez formula (5) as in section II-D for propagating the unit vector

$$\begin{aligned} \boldsymbol{\omega}_u^c[k] &= \boldsymbol{\omega}^c[k] + d_v[k|k] \mathbf{S}(\hat{\mathbf{a}}^c[k]) \mathbf{u}_v^c[k|k] \\ \mathbf{u}_v^c[k+1|k] &= \mathcal{R}(\Delta t \boldsymbol{\omega}_u^c[k])^\top \mathbf{u}_v^c[k|k] \\ &\approx \mathbf{I} - \Delta t (\mathbf{S}(\boldsymbol{\omega}^c[k]) \mathbf{u}_v^c[k|k] - d_v[k|k] \mathbf{S}(\mathbf{u}_v^c[k|k])^2 \hat{\mathbf{a}}^c[k]) \\ d_v[k+1|k] &= d_v[k|k] - \Delta t d_v^2[k|k] (\mathbf{u}_v^c[k|k])^\top \hat{\mathbf{a}}^c[k] \end{aligned} \quad (23)$$

where  $k \in \mathbb{N}$  denotes the  $k$ th iteration step, and  $\bullet[k|l]$  is the estimate of the state at time-step  $k$ th given the  $l$ th measurement. In addition,  $\boldsymbol{\omega}^c[k]$  and  $\hat{\mathbf{a}}^c[k]$  are the averaged sensor values between  $t_k$  and  $t_{k+1}$ . When calculating the Jacobi matrices we utilize (23)

$$\mathbf{F}_{ekf} = \begin{bmatrix} \frac{\partial \mathbf{u}_v^c[k+1|k]}{\partial \boldsymbol{\omega}^c[k|k]} & \frac{\partial \mathbf{u}_v^c[k+1|k]}{\partial d_v[k|k]} \\ \frac{\partial d_v[k+1|k]}{\partial \boldsymbol{\omega}^c[k|k]} & \frac{\partial d_v[k+1|k]}{\partial d_v[k|k]} \end{bmatrix} \quad (24)$$

$$\begin{aligned} \frac{\partial \mathbf{u}_v^c[k+1]}{\partial \mathbf{u}_v^c[k]} &= \mathbf{I} - \Delta t (\mathbf{S}(\boldsymbol{\omega}^c[k] + d_v[k] \mathbf{S}(\hat{\mathbf{a}}^c[k]) \mathbf{u}_v^c[k]) \\ &\quad - d[k] \mathbf{S}(\mathbf{u}_v^c[k]) \mathbf{S}(\hat{\mathbf{a}}^c[k])) \\ \frac{\partial \mathbf{u}_v^c[k+1]}{\partial d_v[k]} &= -\Delta t \mathbf{S}(\mathbf{u}_v^c[k])^2 \hat{\mathbf{a}}^c[k] \\ \frac{\partial d_v[k+1]}{\partial \mathbf{u}_v^c[k]} &= -\Delta t d_v^2[k] (\hat{\mathbf{a}}^c[k])^\top \\ \frac{\partial d_v[k+1]}{\partial d_v[k]} &= \mathbf{I} - \Delta t 2d_v[k] (\hat{\mathbf{a}}^c[k])^\top \mathbf{u}_v^c[k] \end{aligned}$$

The covariance matrix is then propagated according to

$$\mathbf{P}[k+1|k] = \mathbf{F}_{ekf} \mathbf{P}[k|k] \mathbf{F}_{ekf}^\top + \mathbf{R}_{ekf}$$

where  $\mathbf{R}_{ekf}$  are approximated through linearization

$$\mathbf{R}_{ekf} = \mathbf{V} \begin{bmatrix} \text{cov}(\boldsymbol{\omega}^c[k]) & 0 \\ 0 & \text{cov}(\hat{\mathbf{a}}^c[k]) \end{bmatrix} \mathbf{V}^\top + \mathbf{R}_{reg} \quad (25)$$

where  $\mathbf{R}_{reg}$  is a regularization matrix and

$$\mathbf{V} = \begin{bmatrix} \frac{\partial \mathbf{u}_v^c[k+1|k]}{\partial \boldsymbol{\omega}^c[k]} & \frac{\partial \mathbf{u}_v^c[k+1|k]}{\partial d_v[k|k]} \\ \frac{\partial d_v[k+1|k]}{\partial \boldsymbol{\omega}^c[k]} & \frac{\partial d_v[k+1|k]}{\partial d_v[k|k]} \end{bmatrix}$$

$$\begin{aligned} \frac{\partial \mathbf{u}_v^c[k+1|k]}{\partial \boldsymbol{\omega}^c[k]} &= \Delta t \mathbf{S}(\mathbf{u}_v^c[k|k]) \\ \frac{\partial \mathbf{u}_v^c[k+1|k]}{\partial \hat{\mathbf{a}}^c[k]} &= -d_v[k|k] \mathbf{S}(\mathbf{u}_v^c[k|k])^2 \\ \frac{\partial d_v[k+1|k]}{\partial \boldsymbol{\omega}^c[k]} &= \mathbf{0} \\ \frac{\partial \hat{\mathbf{a}}^c[k]}{\partial d_v[k|k]} &= \Delta t d_v^2[k|k] (\mathbf{u}_v^c[k|k])^\top \end{aligned}$$

where the  $\text{cov}(\boldsymbol{\omega}^c[k])$  and  $\text{cov}(\hat{\mathbf{a}}^c[k])$  can be calculated from the data sheet from the IMU, or found empirically. The most challenging task is to find  $\text{cov}(\hat{\mathbf{a}}^c[k])$ , as it is both the combination of the gravity estimate and the specific force measurement. The unit  $\mathbf{z}_{ekf} = \mathbf{u}_v^c(t_{k+1})$  velocity vector is measured at time  $t_{k+1}$ , the jacobian of the measurement is thus trivial

$$\mathbf{J}_{ekf} = [\mathbf{I} \quad \mathbf{0}_{3 \times 1}]$$

The Kalman gain is then calculated using

$$\mathbf{K}_{ekf} = \mathbf{P}[k|k] \mathbf{J}_{ekf}^\top (\mathbf{J}_{ekf} \mathbf{P}[k|k] \mathbf{J}_{ekf}^\top + \text{cov}(\mathbf{z}_{ekf}))^{-1} \quad (26)$$

The the estimate update is as follows

$$\mathbf{x}_{EKF}[k+1|k+1] = \begin{bmatrix} \mathbf{u}_v^c[k+1|k+1] \\ d_v[k+1|k+1] \end{bmatrix} \quad (27)$$

$$= \begin{bmatrix} \mathbf{u}_v^c[k+1|k] \\ d_v[k+1|k] \end{bmatrix} + \mathbf{K}_{ekf} (\mathbf{z}_{ekf} - \mathbf{u}_v^c[k+1|k]) \quad (28)$$

$$\mathbf{P}[k+1|k+1] = (\mathbf{I} - \mathbf{K}_{ekf} \mathbf{J}_{ekf}) \mathbf{P}[k+1|k] \quad (29)$$

As the magnitude parameter  $d_v$  is inverted, the observer is said to be able to be initialized without any special procedure [7]. However as the filter is based on the EKF, there are no guarantees of convergence, and the requirements for having the observer converge is not clear. Never the less we use this observer as the state of the art that we would like to compare the nonlinear observers with global stability when they have their PE condition satisfied.

### C. Sensor-centric observer - KF

The sensor-centric bearing-only SLAM is presented in [20]. By augmenting the states, and transforming the outputs, they are able to formulate the bearing only SLAM problem as a linear system. This means it can be solved with a Kalman filter using the Riccati equation and we, therefore, give it the abbreviation *KF*. For a single bearing measurement the states to be estimated are  $\mathbf{x}_{KF} = [v^c, s]^\top$ , the input  $\mathbf{w}_{KF} = \hat{\mathbf{a}}^c$ . Where  $v^c$  is the metric velocity of the camera, and  $s$  is the speed; the resulting system dynamics are thus

$$\dot{\mathbf{x}}_{KF} = \mathbf{A}_{KF}(t)\mathbf{x}_{KF} + \mathbf{B}_{KF}(t)\mathbf{w}_{KF} \quad (30)$$

$$\mathbf{y}_{KF} = \mathbf{C}_{KF}(t)\mathbf{x}_{KF} \quad (31)$$

where the time varying system matrices are partially built up by sensor data. The matrices are then

$$\mathbf{A}_{KF}(t) = \begin{bmatrix} -\mathbf{S}(\boldsymbol{\omega}^c(t)) & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \quad \mathbf{B}_{KF}(t) = \begin{bmatrix} \mathbf{I} \\ \mathbf{u}_v^c(t)^\top \end{bmatrix} \quad (32)$$

$$\mathbf{C}_{KF}(t) = [\mathbf{I} \quad -\mathbf{u}_v^c(t)] \quad (33)$$

where the virtual measurement  $\mathbf{y}_{KF} = \mathbf{0}$  works as a constraint on the system. The system is proved to be Uniformly Completely Observable (UCO) if there exist a time when  $\|\dot{\mathbf{u}}_v^n\| > 0$ , implying that the KF can be designed to be globally exponentially stable (GES) [24]. However, as the set-up will provide noise that is not Gaussian because of the unit vector parametrization gives a banana shaped distribution, the optimality of the Kalman filter can not be expected. In addition the system state space model is clearly not minimal, and the noise is state dependent. This may compromise the performance of this observer for more noisy measurements.

### D. Position Observer - PO

The position observers presented in [22] and [25] have measurements rotated in the inertial frame. However since the comparison between the observers is made in the body-frame, we transform the observers to work with measurements in the body-frame. As the transformation from inertial to body-frame can be regarded as a similarity transformation and hence preserves the observability properties of the system, the observer in body-frame will have the same properties as the observer in the inertial frame, which for both observers are GES.

In [22] a similar framework as described in IV-C is presented. However, the range is not explicitly estimated, but implicitly estimated in the relative position estimate. It is also shown how velocity bias estimation is possible. For

one landmark with estimate  $\mathbf{x}_{PO} = v^c$  in camera-frame, and without velocity bias estimation  $\mathbf{w}_{PO} = \hat{\mathbf{a}}^c$ , the system matrices are

$$\mathbf{A}_{PO}(t) = [-\mathbf{S}(\boldsymbol{\omega}^c)] \quad \mathbf{B}_{PO} = [\mathbf{I}] \quad (34)$$

$$\mathbf{C}_{PO}(t) = [\mathbf{S}(\mathbf{u}_v^c(t))^2] \quad (35)$$

where the measurements are  $\mathbf{y}_{PO} = \mathbf{0}$ . The observer will have the same pros and cons as the observer presented above, although it can be argued that the states of the system are minimal. A similar fixed gain version was presented in [25], which is also able to estimate velocity bias. The simple observer with one landmark measurement and without bias estimation, in body-frame, is then

$$\dot{\hat{\mathbf{x}}}_{PO} = -\mathbf{S}(\boldsymbol{\omega}^c)\hat{\mathbf{v}}_{PO}^c + \hat{\mathbf{a}}^c - k_{PO}\mathbf{S}(\mathbf{u}_v^c)^2\hat{\mathbf{v}}_{PO}^c \quad (36)$$

In addition, as it is argued in [26], the difference between finding gains through the Riccati solution compared to using fixed gains are just a matter of tuning. And as it is noted in [22], by designing the Kalman filter with certain covariance matrices, the observer (36) is recovered. Hence we will use the fixed gain observer when we compare this setup to the other observers. The observer is also proven to have GES, assuming that a PE condition is satisfied. For a system with one bearing measurement the PE condition is similar to the KF above; there exist a time when  $\|\dot{\mathbf{u}}_v^n\| > 0$ .

### E. Magnitude Observer - MO

The last observer was originally presented in [19], with proof of semi-global asymptotically and locally exponentially stability. The observer utilizes the dynamic of a unit direction vector and how it relates to the magnitude of the vector corresponding to the unit vector measurement. From this, the inverse magnitude of the vector is estimated, leading to  $\mathbf{v}_{MO}^c = [\frac{\mathbf{u}_x^c}{d_v}]$ . Thus the state representation is identical to the robust EKF in section IV-B. The nonlinear observer is driven by the input  $\hat{\mathbf{a}}^c$  and  $\boldsymbol{\omega}^c$ , and is as follows

$$\dot{\hat{\mathbf{u}}}_x^b = -\mathbf{S}(\boldsymbol{\omega}^c - k_{MO}\tilde{\mathbf{u}}^c)\hat{\mathbf{u}}_v^c + \hat{d}_v\mathbf{S}(\hat{\mathbf{u}}_v^c)\mathbf{S}(\mathbf{u}_v^c)\hat{\mathbf{a}}^c \quad (37)$$

$$\dot{\hat{d}}_v = \text{proj}_d(\hat{d}_v^2(\mathbf{u}_v^c)^\top \hat{\mathbf{a}}^c - \gamma_{MO}(\hat{\mathbf{a}}^c)^\top \mathbf{S}(\mathbf{u}_v^c)^2 \mathbf{S}(\hat{\mathbf{u}}_v^c)\tilde{\mathbf{u}}^b) \quad (38)$$

$$\tilde{\mathbf{u}}^c = \mathbf{S}(\mathbf{u}_v^c)\hat{\mathbf{u}}_v^c \quad (39)$$

the system is proven to have global stability when a projection operator is applied in the estimation, and  $\|\dot{\hat{\mathbf{u}}}_x^n\| > 0$ . This means that there also is a upper bound of what value the estimate  $\hat{d}_v$  can take to guarantee convergence for the system. It is then natural to think that a similar bound should be set on the EKF as well to increase the robustness. Another aspect with the MO observer is that it has the skew-symmetric structure similar to what is discussed in [15]. This means that the observer can be tuned similarly to a second order system by using the gains  $k_{MO} = 2\sqrt{\alpha}\lambda$  and  $\gamma_{MO} = \frac{\alpha}{\|\tilde{\mathbf{B}}_u\|}$ , where  $\tilde{\mathbf{B}}_u = (\hat{\mathbf{a}}^c)^\top \mathbf{S}(\mathbf{u}_x^b)^2 \mathbf{S}(\hat{\mathbf{u}}_x^b)$ . Where the convergence rate will

increase linearly with  $\alpha$ , and  $\lambda$  is the damping ratio of the observer.

*Remark 1:* The PE condition discussed in this section are a bit different, however they all depend on some excitation of the normalized velocity in the inertia frame  $\|\mathbf{u}_v^n\| > 0$  at some time. However when uncertainty and noise is added to the system, the requirement for the PE becomes larger resulting in that the normalized velocity measurement  $\|\mathbf{u}_v^n\| > 0$  should be excited regularly to get good convergence.

## V. SIMULATION RESULTS

The scenario presented in this section is a vehicle travelling in a circle in 3D-space at constant velocity  $\mathbf{v}^b = [0.5, 0, 0][m/s]$ . The trajectory of the vehicle can be seen in Figure 4, which is the same trajectory used in the experiment. The simulator is implemented using Euler integration, having step length  $h = 0.025[s]$ .

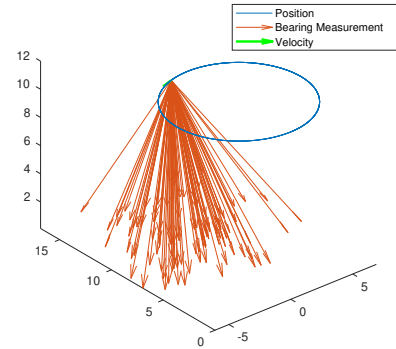
When comparing different observers, how they perform is often related to the tuning of the observers. In this section, we will also compare how easy it is to change the convergence rate of the observers when the observers are implemented on a continuous system without noise. We will also try to find the tuning parameters that give the lowest root mean square error (RMSE) for the selected trajectory and noise parameters presented. We will regard this tuning as optimal, and the mean and variance of the RMSE of consecutive Monte Carlo (MC) simulations is a metric of how well the observers are performing.

The choice of circular trajectory was also made so that the system matrices would be close to constant, making it easier to find what we call the optimal tuning, and thus the comparison was made simpler.

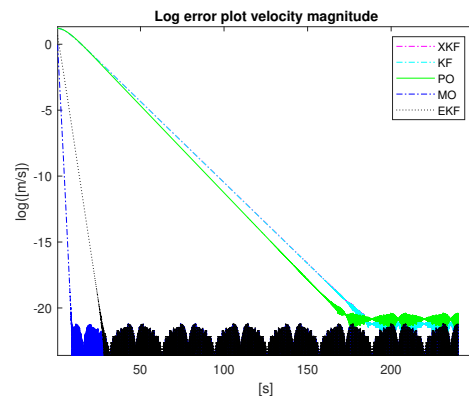
In addition to the observers presented in section IV, we introduce a velocity observer, where the direction measurements are first filtered in the MO observer and later used as measurements for the sensor-centric KF, the MO and KF are thus in cascade and we named the filter XKF as it is inspired by [27]. The motivation for this was to reduce the apparent noise sensitivity of the sensor-centric KF which was seen in the results of [28], where also the semi-global stability of this setup is proven.

### A. Simulation without noise

To verify that the observers had the exponential stability claimed by the authors, simulations were performed without noise. In addition it was investigated how the convergence rate could be tuned for the different observers. For the MO observer, we were able to change the convergence rate of the observer by utilizing the tuning presented in section IV-E, which was confirmed through simulations. For the PO observer with a single gain, the convergence rate was related to the gain  $k_{PO}$  (36), however, there seemed to be a minimum value of the convergence rate, and choosing a gain for faster convergence resulted in oscillations without faster convergence. It was tested if the numerical implementation played a role in the oscillations, however reducing the step length of the discretization did not affect the oscillations. As



**Fig. 4:** The figure shows the trajectory of the vehicle flying in a circle over a plane with a camera pointing downwards. The green arrow represents the normalized velocity measurement; red arrows represent the bearing measurements from features that can be used for velocity estimation using for instance homography



**Fig. 5:** Logplot of the velocity magnitude estimation error. The observers except of the MO was tuned to be as fast as possible. The XKF can not be seen as it lies behind the KF line.

for the sensor-centric observer KF, the observer was tuned by changing the diagonal entries for the measurement and process noise covariance matrices. However, changing these did not affect the rate of convergence of the observer. The EKF was tuned with a combination of diagonal matrices and changing the values of  $\text{cov}(\omega^c)$  and  $\text{cov}(\hat{a}^c)$  in (25). The results can be seen in Figure 5, where the MO observer can be tuned to have arbitrary fast convergence, the PO observer has approximately similar convergence rate as the KF and XKF, when not oscillating. And the KF and EKF did not change convergence rate by changing the diagonal entries of the covariance matrix. In [29], the notion of tuneability and how it is connected to observability is presented. Whether an observability analysis can explain the lack of tunable convergence rate for the sensor-centric KF, XKF and PO observers needs to be investigated further. The notion of tunable convergence rate becomes increasingly important when the observer is working with a controller. As it allows the control designer to design the state feedback controller to meet transient response specification and/or constraint by increasing the convergence rate of the observer.

TABLE I: Qualitative comparisons of the observers presented

Observers	States	Observer Innovation	Tuning parameters	Stability	PE condition
EKF	$[\mathbf{u}_v^c, d_v]$	$\tilde{\mathbf{y}} = \mathbf{u}_v^c - \hat{\mathbf{u}}_v^c$	$\mathbf{Q}_{EKF}, \mathbf{R}_{EKF}$	Unknown	Unknown
KF	$[\mathbf{v}^c, \varrho]$	$\tilde{\mathbf{y}} = \mathbf{0} - \hat{\mathbf{v}}^c - \hat{\mathbf{s}}\mathbf{u}^c$	$\mathbf{Q}_{KF}, \mathbf{R}_{KF}$	GES	$\dot{\mathbf{u}}^n > 0$
PO	$\mathbf{v}^c$	$\tilde{\mathbf{y}} = \mathbf{0} + \mathbf{S}(\mathbf{u}^c)^2 \hat{\mathbf{v}}^c$	$k_{PO}$	GES	$\int_t^{t+T} \mathbf{S}(\mathbf{u}^n(\tau))^2 d\tau > \mu \mathbf{I}, \ \mathbf{v}^c\  > 0$
MO	$[\mathbf{u}_v^c, d_v]$	$\tilde{\mathbf{y}} = \mathbf{S}(\mathbf{u}^c)\hat{\mathbf{u}}^c$	$k_{MO}, \gamma_{MO}$	USGAS LES	$\dot{\mathbf{u}}^n > \mu, \ \mathbf{v}^c\  > c_\delta$

### B. Noisy simulation and tuning

For the noisy simulations, realistic values of the sensor noise were chosen. The IMU measurements were corrupted by white noise with standard deviation  $\sigma_\omega = 0.02\mathbf{I}$  [rad/s] and  $\sigma_f = 0.02\mathbf{I}$  [m/s<sup>2</sup>], which is meant to resemble a low cost MEMS IMU. The white noise in the bearing measurements was  $\sigma_u = 0.00314\mathbf{I}$  [rad], resembling a pixel error for a camera with 90° field of view and 500 pixels image height/width; the AHRS white noise was  $\sigma_{\mathcal{R}} = 0.0116\mathbf{I}$  [rad] giving a 3σ value of 2°; the velocity direction had a white noise of  $\sigma_v = 0.1060\mathbf{I}$  [rad], which is the mean error from a homography with the presented setup [14]. The bearing noise is orthogonal to the bearing  $\mathbf{u}_n = \mathbf{S}(\mathbf{u}_x^n)\mathbf{w}_u$ , in which the noise  $\mathbf{w}_u$  is a white noise vector  $\mathbf{w}_u = \mathcal{N}(0, \sigma_u)$ , and the same is applied to the noise of the normalized velocity. As the unknown for the observers is the magnitude of the velocity, we utilize the velocity magnitude estimation error to compare the observers, and its RMSE as a performance metric.

The four observers and the EKF we are comparing have different tuning parameters. The MO observer has two parameters  $k_{MO}$  and  $\gamma_{MO}$ ; the PO has one  $k_{PO}$ , this can also be expanded to a matrix  $\mathbf{K}_{PO} \in \mathbb{R}^{3 \times 3}$  although this expansion did not lead to improved performance. The sensor-centric KF has two matrices, corresponding to the process and measurement noise covariance noise matrices,  $\mathbf{Q}_{KF} \in \mathbb{R}^{4 \times 4}$  and  $\mathbf{R}_{KF} \in \mathbb{R}^{3 \times 3}$ . As the XKF is a cascade of the MO filter and the KF the tuning parameters are inherited from the MO observer and the KF observer, and the EKF has the tuning covariance matrices  $\mathbf{Q}_{EKF}$  and  $\mathbf{R}_{EKF}$ .

The observers were tuned using Monte Carlo (MC) simulations, where the tuning parameters were chosen randomly. For the candidate MO and PO observers, as there were few parameters to choose from, choosing the pool of possible tuning parameters was straightforward. The results from these MC simulations can be seen in Figure 6 and 7. From these simulations, it is evident that the MO observer is able to estimate the velocity from a large set of tuning parameters (yellow area), and the PO observer also seems to have an optimal tuning.

To investigate how the sensor-centric KF and XKF observer could be tuned properly, the covariance matrix of the process noise and innovation was found empirically. This was done by simulating the system in parallel with and without noise, and the covariances were estimated from the difference between these two simulations. As the observer was implemented by Euler discretization, the covariance

estimate was of the discrete system, where  $\Delta\mathbf{x} = h\mathbf{f}(\mathbf{x}, \mathbf{u})$

$$\begin{aligned} \hat{\mathbf{Q}} &= cov(\Delta\mathbf{x} - \Delta\mathbf{x}_m) \\ &= h^2 cov((\mathcal{A}_{KF}(t_k) - \mathcal{A}_{KF}(t_k)_m)\mathbf{x} \\ &\quad + \mathcal{B}_{KF}(t_k)\mathbf{u}(t_k) - \mathcal{B}_{KF}(t_k)_m\mathbf{u}(t_k)_m) \end{aligned}$$

where  $h$  is the discretization interval, and the  $t_k$  is the discrete time and  $\mathcal{A}_{KF}(t_k)$  is the system matrix from (33) with true sensor input, while  $\mathcal{A}_{KF}(t_k)_m$  has noisy input, the same applies to  $\mathcal{B}_{KF}(t_k)$  and  $\mathbf{u}(t_k)$  and their corresponding noisy version. For estimating the measurement noise the covariance was estimated from

$$\begin{aligned} \hat{\mathbf{R}} &= cov(\mathbf{y} - \mathbf{y}_m) \\ &= cov(\mathbf{0} - \mathcal{C}_{KF}(t_k)_m\mathbf{x}) \end{aligned}$$

The covariance was estimated by running 10000 simulations and averaging the covariance estimates from these runs. The matrices were then tested by calculating the normalized error (NE)

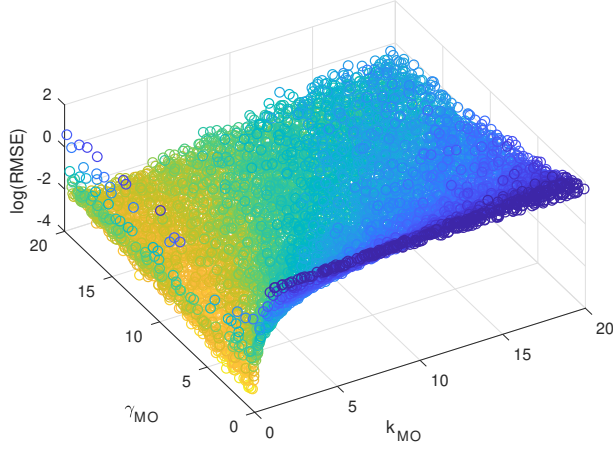
$$\begin{aligned} NE_Q(t_k) &= \\ &= \frac{1}{N} \sum [\Delta\mathbf{x}(t_k) - \Delta\mathbf{x}(t_k)_m]_i^\top \hat{\mathbf{Q}}^{-1} [\Delta\mathbf{x}(t_k) - \Delta\mathbf{x}(t_k)_m]_i \end{aligned}$$

where  $[\Delta\mathbf{x}(t_k) - \Delta\mathbf{x}(t_k)_m]_i$  is the discrete process noise of the  $i$ th MC simulation at time  $t_k$ . From this test we could verify that the covariance found was right by verifying that  $E[NE_Q(t)] \approx 4$ , as  $\dim(\hat{\mathbf{Q}}) = 4$ .

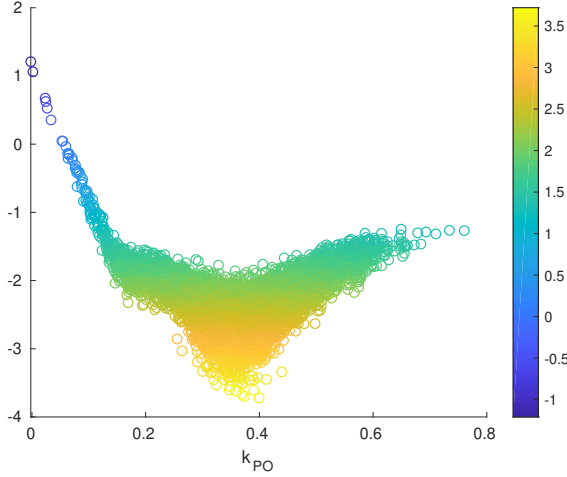
Theses covariance matrices were then used as starting point for the tuning of the KF and XKF. It was tried to add a diagonal regularization matrix to these, witch worked for the sensor-centric XKF. For the KF the tuning variant giving the best results was multiplying the covariance matrices by a gains  $\mathbf{Q}_{KF} = q_g \hat{\mathbf{Q}}_{KF}$  and  $\mathbf{R}_{KF} = r_g \hat{\mathbf{R}}_{KF}$ . The results of the KF simulations can be seen in figure 8. The initial covariance  $P(0)$  was initialized with its true value.

For the sensor-centric observers the KF achieves good results when the covariance matrix is found empirically, conversely, the XKF version has better results when the tuning covariance matrices had larger diagonal entries. In Figure 9 we see how the tuning parameters of the MO part of the XKF affect the observer. Both the tuning parameters of the MO and KF are varied, however, the most significant parameter for the performance of the XKF seems to be the  $k_{MO}$ . The authors believe this is related to the sensitivity of the KF part of the cascade has to noisy measurements, and





**Fig. 6:** Plot of the velocity log(RMSE) error of the MO from 10000 MC simulations with varying tuning parameter

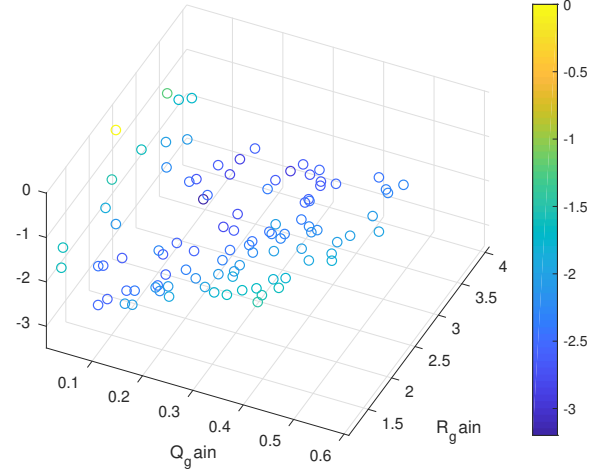


**Fig. 7:** Plot of log(RMSE) of the MO from 10000 MC simulations with varying tuning parameters

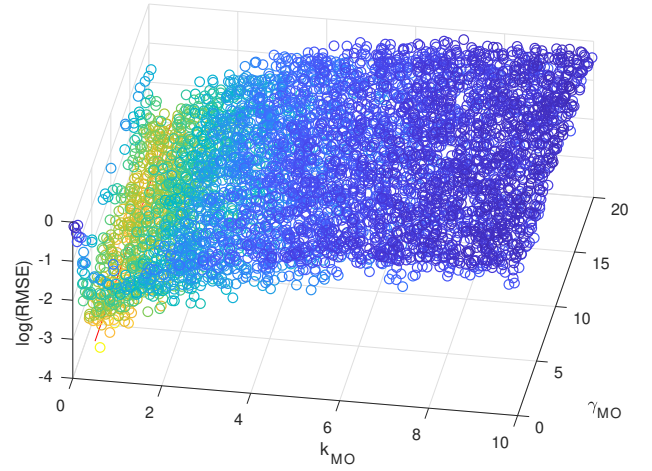
how noisy the measurement the KF gets, is highly related to the parameter  $k_{MO}$ . Hence from the figure, we see that the gain of the MO observer should be  $k_{MO} = 0.5$ , and thus the MO observer will work as a low-pass filter for the bearing measurements. When this was the case, the XKF seemed to have a much larger set of tunings parameters that gave good results compared to the KF.

In Figure 8 we see the result from the MC simulations for the sensor-centric KF, where the best result with lowest RMSE was  $q_g = 0.22$  and  $r_g = 4$ . Why the empirical covariance did not perform the best is probably because the noise is state dependent and the empirical noise was found for actual state estimates.

The EKF was similarly tuned, where the covariance of both  $\hat{a}^c$  and  $u_v^c$  were found through 10000 MC simulations, and later a tuning parameter  $r_{EKF}$  and  $q_{EKF}$  are used so



**Fig. 8:** Scatter plot of the log(RMSE) error of the KF for different gains multiplied to the covariance matrices. The simulations was run 30 000 times, and tuning parameters were repeated over 300 times and the log(RMSE) shown is the average of 300 runs.



**Fig. 9:** Scatter plot of the log(RMSE) of the XKF observer versus the tuning parameters of the MO observer. The results are from 10000 MC simulations, where the diagonal entries of the parameters of the KF are varied from  $[0.2, 1.5]10^{-3}$  for the process noise covariance matrix  $Q$ , for the measurement matrix  $R$  the parameters varied from  $\text{diag}([0.0015, 0.01, 0.01])$  to  $\text{diag}([0.01, 0.02, 0.02])$

that

$$\begin{aligned} Q_{EKF} &= V \text{cov}([\omega^c; \hat{a}^c]) V^T + I q_{EKF} \\ R_{EKF} &= \text{cov}(u_v^c) r_{EKF} \end{aligned}$$

where the best values were found to be  $q_{EKF} = 2.6 \cdot 10^{-3}$  and  $r_{EKF} = 3.6$ , by several MC simulations.

From the MC simulations, we thus found what we regarded as the best tuning. To test how the observers performed with the chosen tuning, 10000 MC simulations were run. The mean and variance of the RMSE can be seen in Table II, where we see that the observers performing best is the MO observer followed by the XKF. As these observers have the combination of low mean and variance

**TABLE II:** Table of the resulted mean and variance of the RMSE for 10000 MC simulations for the selected tuning of the observers. The columns with B ending is for normal noise described in section V-B, while the C ending is from the high noise scenario described in section V-C

Observer	Mean RMSE B	VAR RMSE B	Mean RMSE C	VAR RMSE C
(MO)	0.046	$1.47 \cdot 10^{-4}$	0.192	$5.7 \cdot 10^{-3}$
(PO)	0.076	$4.49 \cdot 10^{-4}$	0.324	$1.8 \cdot 10^{-3}$
(KF)	0.061	0.030	0.638	$6.6 \cdot 10^{-4}$
(XKF)	0.051	$1.349 \cdot 10^{-4}$	0.414	0.015
(EKF)	0.067	$2.473 \cdot 10^{-4}$	2.592*	4.74*

\* As 45% of the EKF had problems converging and ended up with an RMSE that was saturated to 5. None of the other observers had similar problems.

of their RMSE. We see that the variance of the sensor-centric KF is significantly larger than the other observers, and by examining the histogram of the RMSE values from the MC simulation we see that for about 0.61% of the runs the KF fails to converge properly. The relationship between the noise and the estimate can also explain why this filter fails to converge for certain runs. For the XKF this problem was not seen, possibly because the noise was significantly reduced by the MO filter.

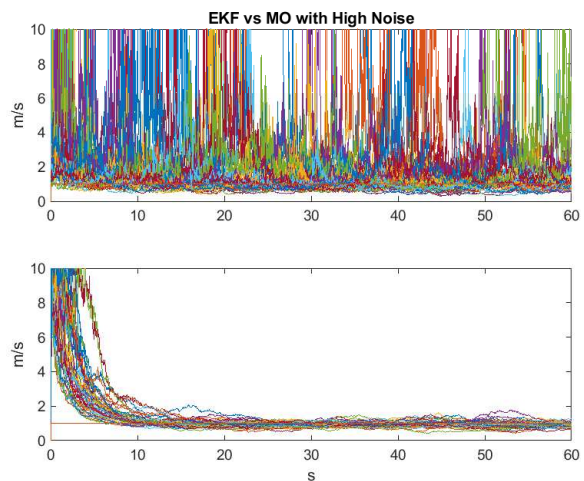
### C. High Noise simulation

To see how robust the methods are to noisy measurements, a MC simulation was run where the variance of all the noise parameters are increased by three times. The tuning was kept at similar to the previous simulation to see how the observers would react to the change. The results can be seen in Table II in columns Mean RMSE C and VAR RMSE C, where it is seen how the different observers react to the higher noise. The PO, KF and XKF gets a substantial bias because of the noise with varying variance. The MO also gets substantial worse results however it is also has the best performance for this simulation. The EKF fails on almost half of the simulations which can be seen in Figure 10. The biggest surprise was the degeneration of the XKF, as this has previous shown resilience toward noise. However, a re-tuning of the observer would make it perform better, which is expected to be true for all the observers, especially the EKF.

## VI. EXPERIMENTAL VERIFICATION

### A. Experimental setup

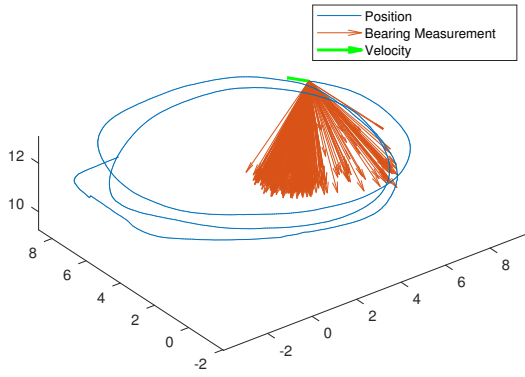
The experimental verification was carried out by using an Octocopter flying in a circle with an autopilot for 150 seconds. The data set was recorded from a payload consisting of SenTiStack, which is built up of a SenTiBoard, a uBlox GNSS receiver and a STIM300 tactical grade IMU [30]. The SenTiBoard was also connected to the flash signal of a uEye UI-3140CP camera, time stamping the flash from the camera so that accurate timing from the images were available. The sensor data and images were stored using an Odroid UX4. Before the flight, an IMU, camera and temporal calibration was performed using the Kalibr toolbox [31], [32], finding IMU biases, as well as the time delay and coordinate transformation between the camera and the IMU. In the data set presented in this article, the image



**Fig. 10:** The velocity magnitude estimate of 50 trajectories of the MO vs EKF in the high noise scenario described in section V-C. The upper plot shows the estimates from the EKF while the lower shows the MO estimates. The 50 trajectories were chosen randomly from the 10000 MC trajectories

time delay estimated was less than 3 ms, and as the timing of the image was related to the flash signal hence it was independent of the kernel load and thus assumed to be constant. As the IMU was tactical grade, the accelerometer and gyro biases were assumed constant for the duration of the experiment and respectively  $\mathbf{b}_a = [0.03, 0.005, 0.085]^T$  [m/s<sup>2</sup>] and  $\mathbf{b}_\omega = [0.007, -0.0002, 0.0017]^T$  [rad/s]. The gravity vector was also estimated using this scheme and was propagated using the gyro for the rest of the flight and used as a gravity measurement, which was possible as the gyro-bias was stable for the time frame recorded. The data set used, hence consists of synchronized and calibrated IMU and camera measurements, in addition, gravity direction estimate in body is correctly initialized; GNSS is also available as navigation reference.

We also wanted to verify the velocity observers to the homography observer presented in [14]. The nonlinear homography observer will be denoted HOM. As we wanted to compare the observers to the one based on homography, we flew over a flat field, so that we could use homography transformation between two images to get the camera velocity we needed. For more detail on how this is done see [14]. Although we emphasize that for the observers using normalized velocity, the camera velocity can come from many other computer vision algorithms. For feature extraction we used the Kanade-Lucas-Tomasi feature tracker [33], the homography matrix was found with a 4-point direct linear transformation (DLT) [34] and outlier rejection was done using RANSAC. The velocity was extracted from the Homography using techniques based on [35]. As discussed in [14], the homography can be a limitation for several scenarios, nevertheless, our experience is that the method is superior to the other methods based on epi-polar geometry when the features are observed on a close to planar surface [36]. We therefore also chose to use the homography to acquire the camera velocity, and this velocity was then nor-



**Fig. 11:** The figure shows the trajectory of the vehicle for the experimental verification. The green arrow represents the normalized velocity measurement; the red arrows represent the bearing measurements from the camera.

malized to fit into the unit vector framework. Alternatively we could also apply the two-view framework in [2], where homography and the 5 point algorithm is used in parallel and the velocity is chosen through a weighting scheme. This also shows the flexibility of the unit velocity framework, by how easily the unit velocity measurement is acquired from either the homography or essential matrix.

The velocity reference magnitude in Figure 12 was the ublox GNSS velocity. For the reference velocity in Figure 13 a RTK and camera aided inertial navigation system (INS) was utilized; based on the unit quaternion and a multiplicative extended Kalman filter (MEKF) [37].

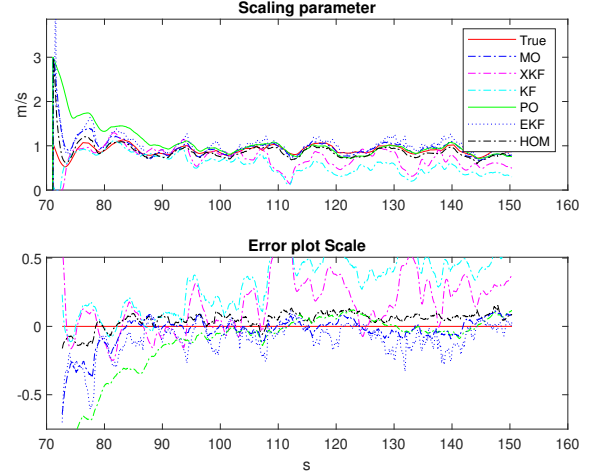
### B. Experimental Results

The scenario presented in this section is a vehicle traveling in a circular pattern in 3D-space, with a camera looking downwards. The trajectory of the vehicle can be seen in Figure 11, where the GNSS measurements, velocity and bearing measurements at the last frame are shown.

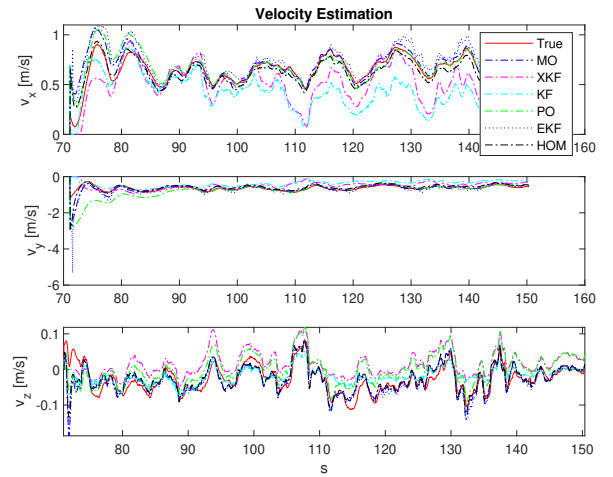
For the experimental data set, the MO and PO observers were tuned similarly to the simulations. The XKF, KF and EKF had to be re-tuned, which was done as in the simulations using MC simulations.

The continuous-time observers were also discretized using forward Euler at every image time stamp, where the measurements from the IMU were integrated between the images. The observers were tested on the data with two different initial condition, and the results can be seen in Table III. From Figure 12, we see how the different observers perform on the recorded data set when the velocity was initialized with magnitude  $\|\hat{v}^c\| = 3m/s$ . The MO observer has comparable results to the HOM observer, and performs superior to the other unit vector observers, with fast convergence and low RMSE error. The EKF starts out with a poor transient, that was not seen when the observer was initialized closer to the true value. However the EKF is able to get satisfactory results with a bit more error than the MO and HOM. The PO, KF and XKF filter are able to stabilize their estimates, though with varying error. The PO has slow convergence, though, it gets good accuracy after convergence. The benefit of filtering the unit vector with the MO observer is also apparent as the

sensor-centric XKF performs better than the sensor-centric KF also on the experimental data. In Figure 13, we see the estimated velocities compared to the estimated velocity from the GNSS and camera aided INS, and especially MO, HOM and eventually PO are able to provide accurate velocity estimates for the experimental setup.

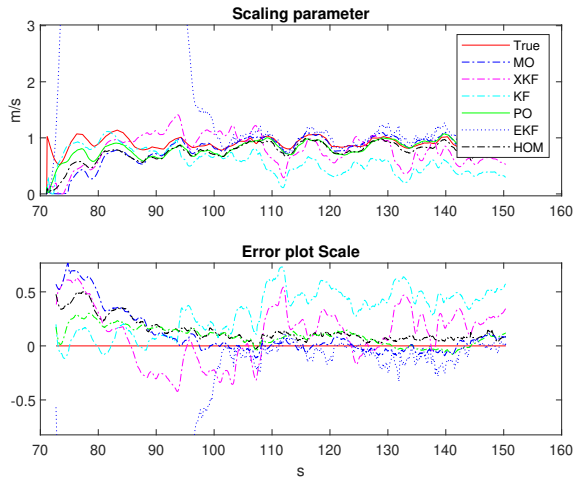


**Fig. 12:** The upper plot shows the velocity magnitude estimate combined with the norm of the GNSS velocity, while the bottom plot shows the velocity error



**Fig. 13:** The figure show the estimates of the velocity of the camera compared to the reference velocity.

As some of the observers use the inverted velocity magnitude as state, it is natural to examine what happens when the inverted estimate gets to large. The observers was therefore initialized with a small velocity. For a velocity smaller than  $\|\hat{v}^c\| \leq 0.11m/s$  the EKF had problems converging. A test was therefore performed just above this limit with initial value  $\|\hat{v}^c\| = 0.115m/s$  where the result can be seen in Figure 14. We see that all the nonlinear observers are able to converge nicely, even the MO which relies on inverted magnitude estimate had no problem, even without the projection activated. The EKF however seems to get



**Fig. 14:** The upper plot shows the velocity magnitude estimate combined with the norm of the GNSS velocity, while the bottom plot shows the velocity error.

**TABLE III:** RMS error from the experimental results in Figure 12

	MO	PO	XKF	KF	EKF	HOM
$1:70 \leq t \leq 110$	0.308	0.585	0.270	0.294	0.381	0.215
$110 \leq t \leq 150$	0.0542	0.0599	0.307	0.477	0.107	0.0820
$2:70 \leq t \leq 110$	0.346	0.205	0.336	0.260	47.7	0.281
$110 \leq t \leq 150$	0.0543	0.0762	0.282	0.244	0.107	0.0855

trouble when the inverted estimate becomes small enough, and it overshoots the estimates and uses tens of seconds to converge. Once they converged all have similar accuracy as the previous run, as expected.

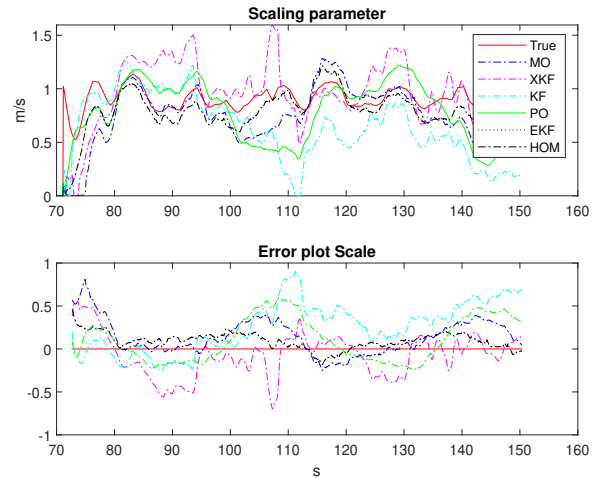
### C. Error in Gravity

A possibly optimistic assumption for the presented setup is that an accurate gravity estimate is available through rotation of the gravity vector in NED/ENU coordinates using AHRS measurements. However, there exist several industrial AHRSs with sub degree error in pitch and roll. To evaluate how an error in gravity could affect the observers, the data set was ran with an time-varying error of the gravity direction oscillating between  $0.2^\circ$ - $0.4^\circ$ . The resulting performance can be seen in Figure 15, with the corresponding RMS errors presented in Table IV. The velocity estimates were initialized at  $\|\hat{v}^c\| = 0.115m/s$ .

**TABLE IV:** RMSE error from the experimental results in Figure 15 with the error in gravity oscillating from  $0.2^\circ$ - $0.4^\circ$

	MO	PO	XKF	KF	EKF	HOM
$70 \leq t \leq 110$	0.324	0.297	0.348	0.300	NaN	0.212
$110 \leq t \leq 150$	0.192	0.277	0.173	0.463	NaN	0.101

We see that the nonlinear observers are not destabilized, and are variably affected by the error introduced. Though the EKF diverged in this scenario the initial estimate  $\hat{d}_v$  was too large. Of the nonlinear observers, the worst affected is the PO observer, which has large oscillations, while the least affected is the HOM observer that has almost the same RMSE, but in this scenario the error is oscillating. The MO observer is also affected by the error, but less than the PO,



**Fig. 15:** The upper plot shows the velocity magnitude estimate combined with the norm of the GNSS velocity, while the bottom plot shows the velocity error, for the data set run with an error in gravity oscillating from  $0.2^\circ$ - $0.4^\circ$

this can be due to the slow convergence of the PO observer. A surprising result was that the XKF in this scenario had actually better results than without the error. Why that was the case is uncertain, but it could maybe be because the velocity estimates of the XKF were shifted upwards as a result of the added gravity error, which can be seen by comparing Figure 12 and 15.

From the experimental results, we thus see that the setup presented is able to estimate the velocity of the vehicle, with global convergence of the velocity estimate, by only having calibrated IMU/AHRS and camera measurements. This also shows that the ego-motion estimation can be performed by the observers presented. The setup presented does not need the camera to observe a plane nor observe features over several images. As it uses normalized velocity as measurement, it can get its camera velocity estimate from VO or visual SLAM methods. It can also be used to scale a converged visual SLAM solution, if a sufficiently accurate IMU is present. It was also shown how the nonlinear observers had more robust behavior than the equivalent robust EKF implementation.

It was also seen how the fixed gain observers were easier to tune, and could reuse their tuning on the experimental data set. The proposed setup can either be used for initializing a VO or VIO algorithms if a calibrated IMU/AHRS is available or it can provide globally stable velocity estimated in-itself for globally stable bearing-only navigation methods such as [20], [25], [38], paving the way for a globally stable SLAM solution with calibrated IMU, camera and AHRS measurements.

## VII. CONCLUSION

We presented a comparative study of globally stable observers applied to the ego motion estimation. The solutions give globally stable velocity estimates provided that camera, IMU and AHRS measurements are available. The compari-

son was shown both in simulations and on experimental data, with varying results for the different observers. The tuning of the different observers was discussed and performed using Monte Carlo simulations and by estimating the measurement and process noise of some of the observers. In simulations the MO observer performed the best, followed by the XKF observer, where it was also shown how the MO observer could work as a low-pass filter for unit vector measurements. Also on experimental data, the MO observer was shown to have the best overall performance of the velocity direction observers, with fast convergence and small RMSE for steady state results. The nonlinear observers were also compared to an EKF, and the observers were shown to have a more robust performance than the EKF.

Nevertheless, the results of this article confirms that ego-motion estimation with camera is possible with global convergence of the velocity estimate, provided that an IMU, AHRS and camera is available, which becomes a more realistic setup as IMU and AHRS sensors improve their performance in the future.

#### ACKNOWLEDGEMENT

This work was supported by the Research Council of Norway through the Centres of Excellence funding scheme, project number 250725, and Centre for Autonomous Marine Operations and Systems, project number 223254. Thanks to Pål Kvaløy and Lars Semb for providing the test flights, and also thanks to Robert Rogne, Sigurd Albrektsen and Frederik Leira for help with the payload.

#### REFERENCES

- [1] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22, IEEE, 2014.
- [2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [3] L. Kneip, M. Chli, and R. Y. Siegwart, "Robust real-time visual odometry with a single camera and an imu," in *Proceedings of the British Machine Vision Conference 2011*, British Machine Vision Association, 2011.
- [4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1052–1067, 2007.
- [5] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual slam: why filter?," *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.
- [6] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, (Rome, Italy), pp. 3565–3572, IEEE, apr 2007.
- [7] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 298–304, IEEE, 2015.
- [8] S. Weiss, R. Brockers, and L. Matthies, "4DoF drift free navigation using inertial cues and optical flow," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 4180–4186, 2013.
- [9] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *International Conference on Intelligent Robots and Systems (IROS) IEEE/RSJ*, pp. 3923–3929, 2013.
- [10] S. Weiss, R. Brockers, S. Albrektsen, and L. Matthies, "Inertial optical flow for throw-and-go micro air vehicles," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 262–269, IEEE, 2015.
- [11] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1722–1729, IEEE, 2015.
- [12] R. Mur-Artal and J. D. Tardos, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [13] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, "Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 18–25, 2017.
- [14] V. Grabe, H. H. Bühlhoff, D. Scaramuzza, and P. R. Giordano, "Nonlinear ego-motion estimation from optical flow for online control of a quadrotor UAV," *The International Journal of Robotics Research*, vol. 34, no. 8, pp. 1114–1135, 2015.
- [15] R. Spica and P. R. Giordano, "A framework for active estimation: Application to structure from motion," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 7647–7653, 2013.
- [16] R. Spica, P. R. Giordano, and F. Chaumette, "Active structure from motion: Application to point, sphere, and cylinder," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1499–1513, 2014.
- [17] D. Nistér, "An Efficient Solution to the Five-Point Relative Pose Problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
- [18] S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart, "Versatile distributed pose estimation and sensor self-calibration for an autonomous MAV," in *International Conference on Robotics and Automation (ICRA)*, pp. 31–38, 2012.
- [19] E. Bjørne, J. Delaune, and T. A. Johansen, "Estimating vector magnitude from its direction and derivative, with application to bearing-only SLAM filter problem," in *57th IEEE Conference on Decision and Control*, IEEE, 2018.
- [20] P. Lourenço, P. Batista, P. Oliveira, and C. Silvestre, "A globally exponentially stable filter for bearing-only simultaneous localization and mapping with monocular vision," *Robotics and Autonomous Systems*, vol. 100, pp. 61–77, feb 2018.
- [21] F. Le Bras, T. Hamel, R. Mahony, and C. Samson, "Observers for position estimation using bearing and biased velocity information," in *Lecture Notes in Control and Information Sciences*, vol. 474, pp. 3–23, 2017.
- [22] T. Hamel and C. Samson, "Position estimation from direction or range measurements," *Automatica*, vol. 82, pp. 137–144, aug 2017.
- [23] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.
- [24] B. D. O. Anderson, "Stability properties of Kalman-Bucy filters," *Journal of the Franklin Institute*, vol. 291, no. 2, pp. 137–144, 1971.
- [25] F. Le Bras, T. Hamel, R. Mahony, and C. Samson, "Observers for position estimation using bearing and biased velocity information," in *Sensing and Control for Autonomous Vehicles*, pp. 3–23, Springer, 2017.
- [26] B. Ekstrand, "Some aspects on filter design for target tracking," *Journal of Control Science and Engineering*, vol. 2012, p. 10, 2012.
- [27] T. A. Johansen and T. I. Fossen, "The exogenous kalman filter (XKF)," *International Journal of Control*, vol. 90, no. 2, pp. 161–167, 2017.
- [28] E. Bjørne, T. A. Johansen, and E. F. Brekke, "Cascaded Bearing Only SLAM with Uniform Semi-Global Asymptotic Stability," in *2017 22th International Conference on Information Fusion (Fusion)*, IEEE, jul 2019.
- [29] V. Andrieu, G. Besançon, U. Serres, and G. Besançon, "Observability necessary conditions for the existence of observers (long version)," tech. rep., 2013.
- [30] S. Albrektsen and T. Johansen, "User-configurable timing and navigation for uavs.," *Sensors*, vol. 18, no. 8, 2018.
- [31] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 1280–1286, IEEE, 2013.
- [32] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2088–2095, IEEE, 2012.
- [33] C. Tomasi and T. Kanade, "Detection and tracking of point features," 1991.
- [34] A. Agarwal, C. Jawahar, and P. Narayanan, "A survey of planar homography estimation techniques," *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*, 2005.

- [35] E. Malis and M. Vargas, *Deeper understanding of the homography decomposition for vision-based control*. PhD thesis, INRIA, 2007.
- [36] S. Šegvic, G. Schweighofer, and A. Pinz, "Performance evaluation of the five-point relative pose with emphasis on planar scenes," in *Proceedings of the workshop of the Austrian Association for Pattern Recognition, Schloss Krumbach, Austria*, pp. 33–40, 2007.
- [37] F. L. Markley, "Attitude error representations for kalman filtering," *Journal of guidance, control, and dynamics*, vol. 26, no. 2, pp. 311–317, 2003.
- [38] E. Bjerne, T. A. Johansen, and E. F. Brekke, "Redesign and analysis of globally asymptotically stable bearing only SLAM," in *2017 20th International Conference on Information Fusion (Fusion)*, pp. 1–8, IEEE, jul 2017.