

Optic Flow Based State Estimation for an Indoor Micro Air Vehicle

M. J. Verveld*

DLR, Lilienthalplatz 7, 38108 Braunschweig, Germany

Q. P. Chu[†], C. De Wagter[‡] and J. A. Mulder[§]

Delft University of Technology, Kluyverweg 1, 2629 HS Delft, the Netherlands

This work addresses the problem of indoor state estimation for autonomous flying vehicles with an optic flow approach. The paper discusses a sensor configuration using six optic flow sensors of the computer mouse type augmented by a three-axis accelerometer to estimate velocity, rotation, attitude and viewing distances. It is shown that the problem is locally observable for a moving vehicle. A Kalman filter is used to extract these states from the sensor data. The resulting approach is tested in a simulation environment evaluating the performance of three Kalman filter algorithms under various noise conditions. Finally, a prototype of the sensor hardware has been built and tested in a laboratory setup.

Nomenclature

$C_{\mathcal{O}}$	Observability condition number
\mathcal{O}	Observability matrix
\mathcal{O}_1	First term of the nonlinear observability matrix
\vec{A}	Specific force vector, as measured by the accelerometer
\vec{f}	Rigid body dynamics function
\vec{h}	Observation function
\vec{n}	Three dimensional unit vector in the sensor viewing direction
\vec{u}	Input vector
\vec{V}	Velocity vector in the body frame of reference
\vec{x}	State vector
\vec{y}	Measurement output vector
d_i	Sensor-obstacle distance of sensor i
dt_s	Sample time step
F	Jacobian matrix of $\vec{f}(\vec{x}, \vec{u})$
H	Jacobian matrix of $\vec{h}(\vec{x})$
P	Error covariance matrix

Symbols

$\underline{\mu}(\mathcal{O})$	Distance to rank deficiency of \mathcal{O}
$\vec{\Omega}_{X_i}$	Optic flow component of sensor i in direction X
$\vec{\Omega}$	Optic flow vector
$\vec{\omega}$	Rotational rate vector in the body frame of reference
$\delta\Omega$	Optic flow step size
$\delta\alpha$	Angular step size
$\vec{\delta}$	Control inputs, determining the aerodynamic moment

*Research Engineer, Unmanned Aircraft Department, Institute of Flight Systems; AIAA Member

[†]Associate Professor, Control and Simulation Division, Faculty of Aerospace Engineering, P.O. Box 5058

[‡]PhD Student, Control and Simulation Division, Faculty of Aerospace Engineering, P.O. Box 5058

[§]Professor, Control and Simulation Division, Faculty of Aerospace Engineering, P.O. Box 5058

The research has been conducted at Delft University of Technology.

I. Introduction

AUTONOMOUS indoor flight by Micro Air Vehicles (MAVs) has become an ever more popular research topic in recent years. It requires technological innovations in many fields such as miniaturization, aerodynamics, control and navigation. Many groups have demonstrated successful vehicle designs based on various rotorcraft and flapping wing configurations. However, solutions to the problem of adding autonomy for indoor flight to these vehicles have been demonstrated only by a few, such as the MIT-MAV team¹ at the IARC 2009 Competition.²

A necessary condition for autonomy is of course sensing the environment and the vehicle state of motion in this environment. While outdoor aircraft typically rely on GPS, an Inertial Measurement Unit (IMU) and a magnetometer for state estimation, this is impossible for indoor MAVs due to the lack of GPS reception and the distortion of Earth's magnetic field inside many buildings. In principle, inertial sensors exist with sufficiently high accuracy to allow inertial navigation for a limited amount of time, but these sensors are too heavy for the class of vehicle considered here and extremely expensive. The only class of inertial sensors suitable for MAVs are Micro-Electro-Mechanical Systems (MEMS). These suffer from significant noise and limited bias stability however.³ Their use in MAV state estimation must therefore rely on the combination with a complementary sensor and appropriate filtering.

Flying insects are capable of avoiding obstacles and navigating through dense unstructured environments without any need for direct distance sensing such as sonar or laser range-finders.⁴ They depend on optic flow sensing processes for these capabilities. Attitude stabilization is achieved through the halteres, which act as gyroscopic sensors⁵ and the dorsal light response, the tendency to align with the up-down gradient of light intensity which is present in most daylight environments.⁶ The huge field of view of the insect's compound eyes is mapped to a layer of Elementary Motion Detector (EMD) neurons. However, different kinds of self-motion may induce the same excitation in an EMD, because locally the corresponding optic flow fields may have the same orientation and magnitude. To extract information for flight control, a set of integrating neurons is wired to the EMDs. Each integrating neuron is sensitive to the input of a subset of EMDs corresponding to an optic flow pattern associated with a particular self-motion component. The firing of such an integrating neuron may then trigger for example a saccadic turn to avoid an obstacle or affect the beat frequency of the wings to adjust the perceived ground speed.⁷

Insects only separate the translatory and rotatory optic flow components and use these directly for flight control. This insect-inspired wide-field integration approach has been demonstrated in autonomous corridor navigation⁸ with optic flow control in the horizontal plane and in urban environments in simulation.⁹ The main advantage of this approach is that it requires little computational effort, although sensor fusion with other sensors may still require filtering. Several partly autonomous indoor optic flow control implementations have been made.^{5,10-13} These feature various combinations of obstacle avoidance and speed control, but only in one plane, the remaining direction being constrained or controlled otherwise.

Optic flow induced by rotation is usually viewed as a disturbance and filtered using gyroscopes. This leaves only the translation induced signal which is then used for proximity detection and/or velocity control. However, in order to apply conventional state control techniques and possibly mapping, it is necessary to obtain separate estimates of body velocity and distances. Therefore, this work aims to extract the full six degrees of freedom motion from optic flow signals. In addition to body motion, pitch and roll angles are also required for flight control. The addition of a 3-axis accelerometer allows for attitude estimation with respect to gravity. Gyroscopes are not required, as the rotational rates may be extracted from the optic flow signals. Solving for these motion components and viewing distances is theoretically possible using six optic flow cameras and three accelerometers. This work investigates the feasibility and performance of this sensor concept.

The remainder of this paper is organized as follows. Section II explains the sensor concept including modeling of the sensor behavior. In order to solve for a given set of state variables, the observability of the sensor equations must be evaluated. Section III defines the necessary tools and analyses the observability problem. To obtain an accurate solution, we perform state estimation using a Kalman filter. Section IV describes the selection of suitable algorithms. A straightforward simulation environment has been created to explore the concept and compare various Kalman filter algorithms. Section V describes the simulation environment. In section VI the sensor hardware is presented which has been used in an experiment. The results from both simulation and experiment are presented and discussed in section VII. Finally, section VIII draws some conclusions about the presented sensor concept and gives recommendations for further work.

II. An Optic Flow Based State Estimation Concept

The research goal described in section I calls for a state estimation method which will enable indoor flight control and obstacle avoidance. Buildings generally have interiors with little room to manoeuvre and bad GPS reception. These properties of the environment makes traditional state estimation based on an IMU/GPS sensor package insufficient for an MAV with an indoor mission requiring some level of autonomy. The aforementioned results of research on flying insects suggests to use the concept of optic flow for indoor state estimation.

The quantity optic flow is defined as the angular rate at which a point on an object moves in relation to an optical sensor. As the sensor lens projects the light within its field of view onto a two dimensional surface, the angular rate of features in the image is two dimensional as well. When the sensor is attached to a body flying through a three dimensional space with stationary objects in it, the optic flow generated by points on the surface of those objects will be a function of the body motion relative to the objects (translations u, v, w and rotations p, q, r) and the distance from the sensor to each point. This assumes that the surface of the objects has adequate texture and is sufficiently illuminated for the sensor to distinguish and track features. In general, optic flow will vary across the field of view of the sensor, giving rise to an optic flow vector field in the spherical coordinates (Ψ and Θ). The type of sensor used in this work outputs only two values which constitute the mean optic flow vector within its field of view. The sensors are modelled as giving the value of the optic flow field along their optical axis. This is a good approximation as long as the field of view is sufficiently narrow. The local optic flow $\vec{\Omega}$ in viewing direction \vec{n} (unit vector) experienced by an observer moving at velocity \vec{V} and rotational rate $\vec{\omega}$ in a stationary environment is:⁶

$$\vec{\Omega} = \frac{\vec{V} - (\vec{V} \cdot \vec{n}) \vec{n}}{d} + \vec{\omega} \times \vec{n}, \quad (1)$$

where d is the distance to the object seen in direction \vec{n} . Equation (1) results in a three dimensional optic flow vector which is orthogonal to the viewing direction \vec{n} . It consists of a linear combination of translational and rotational motion. The fact that the translations are scaled by the viewing distance complicates the matter of extracting the motion state. However, if one has several sensors pointed in different directions, the same motion will determine each optic flow signal. While the number of unknowns increases by one per additional sensor, the number of equations increases by two, reaching 12 equations for 12 unknowns at 6 sensors. By solving the associated observation equations simultaneously, would it be possible to solve for all the motion states and viewing distances? This would serve both the flight control and obstacle avoidance goals. Figure 1 shows a schematic 3D representation of the proposed configuration with 6 optic flow sensors aligned in opposing pairs along the body axes. The corresponding observation equations derive directly from Eq. (1):

$$\begin{array}{lll} X - Y \text{ plane} & X - Z \text{ plane} & Y - Z \text{ plane} \\ \Omega_{X_1} = \frac{u}{d_1} - r & \Omega_{Z_2} = \frac{w}{d_2} + q & \Omega_{Z_1} = \frac{w}{d_1} + p \\ \Omega_{Y_2} = \frac{v}{d_2} - r & \Omega_{X_6} = \frac{u}{d_6} + q & \Omega_{Y_6} = \frac{v}{d_6} - p \\ \Omega_{X_3} = \frac{u}{d_3} + r & \Omega_{Z_4} = \frac{w}{d_4} - q & \Omega_{Z_3} = \frac{w}{d_3} - p \\ \Omega_{Y_4} = \frac{v}{d_4} + r & \Omega_{X_5} = \frac{u}{d_5} - q & \Omega_{Y_5} = \frac{v}{d_5} + p \end{array} \quad (2)$$

Equation (2) assumes that the sensors are aligned with the body axes. In general, this is not necessarily the case, but applying a fixed rotation with respect to the body axes essentially does not change the state

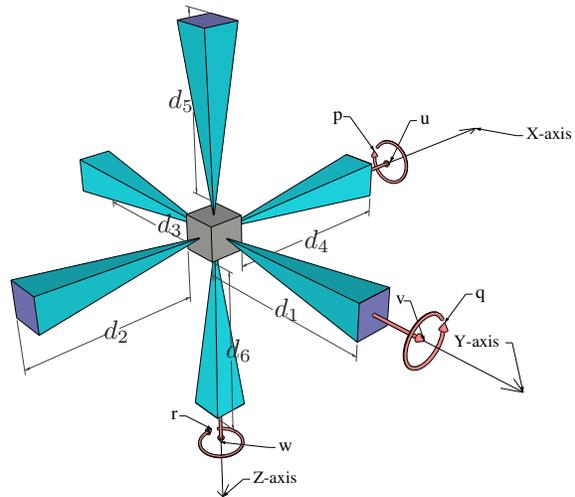


Figure 1. Diagram of the optic flow sensor geometry.

estimation problem.

Unfortunately, this system is not solvable. What fails is one scaling factor to break up the quotients of velocities and distances. In this system, a given room-velocity combination would yield the same optic flow signals when one were to scale all room dimensions and the velocity vector by the same factor.

This observability problem may be solved by augmenting the sensor set with a 3-axis accelerometer and including the vehicle dynamics into the state estimation problem. The presence of gravity enables and requires the estimation of its direction with respect to the body frame which serves flight control as well. This brings the state vector to be estimated to 14 elements: the body velocity (u, v, w) , body rotations (p, q, r) , roll and pitch angles $(\varphi$ and $\theta)$, and the viewing distances $(d_1 - d_6)$. The problem has the following state-space form:

$$\begin{aligned} \dot{\vec{x}} &= \vec{f}(\vec{x}, \vec{u}) + G\vec{w} \\ \vec{y} &= \vec{h}(\vec{x}) \end{aligned}, \quad (3)$$

where $\vec{x} = [u \ v \ w \ p \ q \ r \ \theta \ \varphi \ d_1 \ \dots \ d_6]^T$, $\vec{u} = [A_x \ A_y \ A_z \ \mathcal{M}_x \ \mathcal{M}_y \ \mathcal{M}_z]^T$, $\vec{y} = [\Omega_{X_1} \ \dots \ \Omega_{Y_6}]^T$ and \vec{w} is a white noise vector. G is a diagonal gain matrix to map the process noise. \vec{h} contains the optic flow sensor equations in Eq. (2) and \vec{f} the vehicle dynamics for a general rigid body with known inertia and one plane of symmetry:

$$\vec{f}(\vec{x}, \vec{u}) = \begin{bmatrix} rv - qw - g \sin \theta + A_x \\ pw - ru + g \cos \theta \sin \varphi + A_y \\ qu - pv + g \cos \theta \cos \varphi + A_z \\ \frac{\mathcal{M}_x I_z - qr I_z^2 + qr I_z I_y - J_{xz} p q I_y + J_{xz} p q I_x - J_{xz}^2 r q + J_{xz} \mathcal{M}_z + J_{xz} p q I_z}{-J_{xz}^2 + I_x I_z} \\ \frac{\mathcal{M}_y - (I_x - I_z) r p - J_{xz} (p^2 - r^2)}{I_y} \\ \frac{\mathcal{M}_z I_x - p q I_x I_y + p q I_x^2 - J_{xz} r q I_z + J_{xz} r q I_y - J_{xz}^2 p q + J_{xz} \mathcal{M}_x - J_{xz} r q I_x}{-J_{xz}^2 + I_x I_z} \\ q \cos \varphi - r \sin \varphi \\ p + q \sin \varphi \tan \theta + r \cos \varphi \tan \theta \\ 0 \\ \vdots \end{bmatrix}. \quad (4)$$

The state \vec{x} contains the viewing distances for which the derivative is unknown since it not only depends on the vehicle velocity, but also on the unknown geometry of the environment. This has been modeled as a random walk, i.e. the corresponding elements of \vec{f} have been set equal to zero and the variance is allowed for in the noise term $G\vec{w}$.

An alternative modelling could be to pose that all obstacles are planes which remain perpendicular to each sensor. In that case, the derivatives of the distances would depend only on the body velocity component in the viewing direction. This assumption has been tried in simulation, but the results did not show clear improvement of the prediction in the general case. Rotations tend to cause the largest distance changes, swamping the predicted change. Also, any nonplanar surfaces and edges make the translation-induced distance changes unpredictable. Therefore this modelling has not been used later on.

The input vector \vec{u} contains the accelerometer measurements (A_x, A_y, A_z) and the aerodynamic moments $(\mathcal{M}_x, \mathcal{M}_y, \mathcal{M}_z)$ about the body axes. This assumes that, for a specific MAV, the aerodynamic model has been identified. However, since indoor flight effectively restricts the flight envelope to quasi-hover, a relatively simple model may usually suffice.

III. Observability Analysis

Section II defines a new sensor concept based on optic flow and specific force. In order to extract the desired states from these sensors, observability is a necessary condition to be fulfilled. For Linear Time Invariant (LTI) systems, the definition of observability is well known: an LTI system is called observable if the initial state $\vec{x}(t_0)$ can be reconstructed from knowledge of the input \vec{u} and output \vec{y} on the interval $[t_0, t_1]$ for any

finite t_1 and t_0 with $t_1 > t_0$. The test for observability in the linear case consists of checking the observability matrix for full column rank.

For the problem presented in this paper the concept of observability must be extended to the class of nonlinear state space systems. The corresponding definitions and the rank test for nonlinear systems are included in appendix A. It is important to note here that observability becomes a local property in the nonlinear case: it may vary over the state space.

III.A. A Simplified Test

A practical problem encountered when using the observability rank test (Theorem 2 on page 18) is the complexity of the expressions of the higher order Lie derivatives. The definition of the observability matrix requires an analytical evaluation. But even symbolic math programs may have trouble with a highly nonlinear system in more than a few dimensions due to the rapid expansion of terms. Therefore a simplified test would be of interest when analyzing real-world problems. Equation (11) may be expanded, yielding an approximation to $\mathcal{O}(\vec{x}, \vec{u})$ in terms of the Jacobians of \vec{f} and \vec{h} . Using $d(\vec{a} \cdot \vec{b}) = \vec{a}^T \frac{\partial \vec{b}}{\partial \vec{x}} + \vec{b}^T \frac{\partial \vec{a}}{\partial \vec{x}}$, an arbitrary row of $\mathcal{O}(\vec{x}, \vec{u})$ can be written as:

$$\begin{aligned} dL_f^k(h_i) &= d \left[\frac{\partial}{\partial \vec{x}} \left\{ L_f^{k-1}(h_i) \right\} \vec{f} \right] = dL_f^{k-1}(h_i) \frac{\partial \vec{f}}{\partial \vec{x}} + \vec{f}^T T \frac{\partial (dL_f^{k-1}(h_i))^T}{\partial \vec{x}} \\ &= d \left[\frac{\partial}{\partial \vec{x}} \left\{ L_f^{k-2}(h_i) \right\} \vec{f} \right] \frac{\partial \vec{f}}{\partial \vec{x}} + \vec{f}^T T \frac{\partial (dL_f^{k-1}(h_i))^T}{\partial \vec{x}} \\ &= dL_f^{k-2}(h_i) \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^2 + \vec{f}^T T \frac{\partial (dL_f^{k-2}(h_i))^T}{\partial \vec{x}} \frac{\partial \vec{f}}{\partial \vec{x}} + \vec{f}^T T \frac{\partial (dL_f^{k-1}(h_i))^T}{\partial \vec{x}} \\ &\vdots \\ &= dh_i \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^k + \sum_{j=1}^k \vec{f}^T T \frac{\partial (dL_f^{j-1}(h_i))^T}{\partial \vec{x}} \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^{k-j}, \end{aligned} \quad (5)$$

where $k = 0 \dots n-1$ and $i = 1 \dots p$.

Let

$$\begin{aligned} F(\vec{x}, \vec{u}) &= \frac{\partial \vec{f}(\vec{x}, \vec{u})}{\partial \vec{x}}, \\ H(\vec{x}) &= \frac{\partial \vec{h}(\vec{x})}{\partial \vec{x}}. \end{aligned}$$

Then

$$\mathcal{O}(\vec{x}, \vec{u}) = \begin{bmatrix} H(\vec{x}) \\ H(\vec{x}) F(\vec{x}, \vec{u}) \\ \vdots \\ H(\vec{x}) \{F(\vec{x}, \vec{u})\}^{n-1} \end{bmatrix} + \sum_{j=1}^{n-1} \begin{bmatrix} \emptyset^{jp \times n} \\ E_j(\vec{x}, \vec{u}) \\ \vdots \\ E_j(\vec{x}, \vec{u}) \{F(\vec{x}, \vec{u})\}^{n-1-j} \end{bmatrix}, \quad (6)$$

where

$$E_j(\vec{x}, \vec{u}) = \begin{bmatrix} \vec{f}^T T \frac{\partial (dL_f^{j-1}(h_1))^T}{\partial \vec{x}} \\ \vdots \\ \vec{f}^T T \frac{\partial (dL_f^{j-1}(h_p))^T}{\partial \vec{x}} \end{bmatrix}.$$

The first term in the last expression in Eq. (5) is equal to the i^{th} row in $H(\vec{x}, \vec{u}) \{F(\vec{x}, \vec{u})\}^k$ and the second term is equal to the i^{th} row in $\sum_{j=1}^k E_j(\vec{x}, \vec{u}) \{F(\vec{x}, \vec{u})\}^{k-j}$.

By truncating the expansion of \mathcal{O} in Eq. (6) after the first term, we obtain the following approximation in terms of the Jacobians F and H :

$$\mathcal{O}(\vec{x}, \vec{u}) \approx \mathcal{O}_1(\vec{x}, \vec{u}) = \begin{bmatrix} H(\vec{x}) \\ H(\vec{x}) F(\vec{x}, \vec{u}) \\ \vdots \\ H(\vec{x}) \{F(\vec{x}, \vec{u})\}^{n-1} \end{bmatrix}. \quad (7)$$

In order to use this approximation to determine full local observability for a given problem, the following proposition must hold:

Proposition For any $x_0 \in M$ and some constant permissible input u^*

$$\text{rank } \mathcal{O}_1(\vec{x}_0, \vec{u}^*) \leq \text{rank } \mathcal{O}(\vec{x}_0, \vec{u}^*) . \quad (8)$$

This would make $\text{rank } \mathcal{O}_1$ a lower bound, which is an important condition for its use as it would ensure that a system is locally observable if \mathcal{O}_1 has full rank.

As of yet a proof has not been found. The difficulty is that in general the rank of the sum of two matrices may be lower than the rank of one of them. In this case however, the terms are related to one another. A strategy for proving this proposition would be to formulate the conditions on $dh_i \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^k$ and $\sum_{j=1}^k \vec{f}^T \frac{\partial (dL_f^{j-1}(h_i))^T}{\partial \vec{x}} \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^{k-j}$ such that a linear combination between two rows of \mathcal{O} implies a linear combination of two rows of \mathcal{O}_1 .

Applying the observability rank condition to both \mathcal{O}_1 and \mathcal{O} of a two dimensional version of the optic flow concept presented in this paper has yielded results in support of the proposition. For the remainder of this paper, Eq. (8) is assumed to hold.

III.B. Analysis

To investigate the local observability of the optic flow sensor concept, a large number of points within a realistic part of the state space have been sampled in a grid fashion with an additional focus on zeros. Zero values appear to be the only cases where the observability rank drops. Table 1 lists the findings for the case where the sensor package is aligned with the body frame and positioned in the center of gravity and for the case with a fixed rotation and displacement of the sensor package. Both the rank of the Jacobian of the observations, $H(x)$, and the local observability matrix $\mathcal{O}_1(x, u)$ is listed.

Table 1. Local observability results where particular states have been set to zero while the others have arbitrary (but physically realistic) values.

number of zero states	\vec{V} $\vec{\omega}$	0 0 0	1 1 1	2 2 2	3 3
		0 2 3	0 2 3	0 1 3	0 3
rank condition aligned sensors	rank H rank \mathcal{O}_1	14 14 13	13 13 13	12 12 11	8 8
		14 14 13	14 14 13	12 12 11	8 8
rank condition rotated sensors	rank H rank \mathcal{O}_1	14 14 13	14 14 13	14 14 13	8 8
		14 14 13	14 14 13	14 14 13	8 8

The first conclusion from Table 1 is that, in general, motion is required to observe all the states. At least one component of both \vec{V} and $\vec{\omega}$ should be nonzero in the case of the rotated sensor package. The fact that the aligned sensor package has lower observability rank in some cases can be explained by considering that the alignment with the body axes means that some of the measurements are exactly zero. The rotated sensor package also has critical directions of motion where the observability rank drops. Those directions are not tested in this table.

III.C. Condition Numbers

Although the results from the observability analysis indicate that the problem is observable in the general case with both translational and rotational motion, this does not give information on how “difficult” the problem is to solve. To measure how well a system is observable, the observability condition number \mathcal{C}_O may be used.^{14,15} It is derived from the singular value decomposition of the observability matrix. For nonlinear, time-invariant systems Σ , this will depend on \vec{x} and \vec{u} , i.e. $\mathcal{C}_O(\vec{x}, \vec{u})$ is a local quantity for nonlinear systems.

Definition 1 Let a nonlinear state space system Σ be given as in Eq. (10) on page 18 and its local observability matrix $\mathcal{O}(\vec{x}, \vec{u})$ as in Eq. (11). Then the observability condition number $\mathcal{C}_{\mathcal{O}}$ is defined as the ratio between the maximum and minimum nonzero singular values of $\mathcal{O}(\vec{x}, \vec{u})$. The singular values σ_i are ordered such that

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0, \quad \sigma_r > \sigma_{r+1} = 0,$$

where $r = \text{rank}(\mathcal{O}(\vec{x}, \vec{u}))$. Then $\mathcal{C}_{\mathcal{O}} = \frac{\sigma_1}{\sigma_r}$.

A closely related measure, proposed by Hong et al.,¹⁶ indicates the smallest perturbation in \mathcal{O} which makes \mathcal{O} rank deficient:

Definition 2 Let $\mathcal{O}, \Delta \in \mathbb{R}^{np \times n}$, then

$$\underline{\mu}(\mathcal{O}) \triangleq \min_{\text{rank}(\mathcal{O}-\Delta) < n} \|\Delta\|_2.$$

The following theorem¹⁷ provides a means to compute $\underline{\mu}(\mathcal{O})$:

Theorem 1 Let $\mathcal{O}, \Delta \in \mathbb{R}^{np \times n}$ and σ_i as in definition 1 then,

$$\underline{\mu}(\mathcal{O}) = \sigma_n.$$

These two observability condition measures may also be based on $\mathcal{O}_1(\vec{x}, \vec{u})$. This assumes that $\mathcal{C}_{\mathcal{O}_1} \approx \mathcal{C}_{\mathcal{O}}$. $\mathcal{C}_{\mathcal{O}_1}$ is used in section VII to analyze the observability condition development over time with hardware data. The results in Fig. 10 on page 15 show that $\mathcal{C}_{\mathcal{O}_1}$ behaves as expected, giving empirical grounds for the above assumption.

IV. Kalman filter Algorithm Selection

The state estimation problem posed in section II has been addressed using three variations of the Kalman filter algorithm. A performance comparison between these filters is presented in section VII. As a reference to the discussion below, appendix B includes a summary of these algorithms.

In the physical reality, the vast majority of processes have nonlinearities in their behavior. Therefore many attempts have been made to develop suitable Minimum Mean Squared Error (MMSE) estimators for nonlinear problems. These are generally suboptimal solutions however, because the optimal solution requires that the complete conditional probability density is known, which may require an unbounded number of parameters.¹⁸ The original Kalman filter is the optimal MMSE estimator for linear stochastic problems. It is therefore not unexpected that many approaches have focused on extending the basic Kalman filter algorithm. By far the most commonly applied algorithm is the Extended Kalman Filter (EKF). The straightforward idea behind it is to linearize all the nonlinear models so that the conventional Kalman filter may be applied to the resulting linearized problem.

Depending on the problem, the EKF may not always converge. Specifically, its performance is sensitive to the choice of the initial state estimate. If it is not close enough to the true state at that time, the filter may not converge. This also means that the estimate may diverge at a later time, if the measurements are momentarily of poor quality. This limits the practical usefulness of the EKF.

A method to reduce the effect of measurement function nonlinearity, thereby improving filter performance and convergence interval, is due to J. V. Breakwell,¹⁹ although first published by Denham and Pines.²⁰ Called the Iterated Extended Kalman Filter (IEKF), it comprises local iteration of the measurement update (Eq. (17)), re-linearizing about the updated state η_i to compute a new updated state η_{i+1} which is presumably closer to the true state. The iterative process may continue until the difference between two consecutive iteration steps is below a threshold ε (Eq. (18)). The algorithm is largely identical to the EKF.

Breakwell has pointed out that this local iteration produces a biased estimate. However, as the error covariance becomes smaller, so does the bias in the estimate. This presumes that the filter is converging. Jazwinski²¹ reports significant performance gains of the IEKF over the EKF for some simulated nonlinear systems. The example of a highly nonlinear reentry simulation shows that most performance improvement is achieved with the first two iterations.

To address the inaccuracies arising from the fundamental first order approximation inherent to the (I)EKF implementation, Julier and Uhlmann²² have introduced the concept of Unscented Transforms (UT) and extended it to the problem of recursive estimation. The result is known as the Unscented Kalman Filter (UKF).

The UKF is based on the idea that “*it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation.*”²³ The algorithm propagates a carefully selected set of state vector variations, called sigma points, through the system nonlinear dynamics and then approximates the first two moments of the distribution through weighted sample mean and covariance calculations. No linear approximation of the nonlinear transformation is applied and the order of the probability distribution which is captured by the approximation can be scaled by choosing the number of sigma points. Furthermore, the UKF does not require the calculation of any Jacobian or Hessian matrices, not only simplifying implementation, but also making it suitable for black box applications, e.g. in a filtering toolbox and applications involving non-differentiable functions. When using the conventional number of sigma points, the accuracy of the UKF can be compared to the second-order Gauss filter and the computational order is comparable to the EKF. However, when used with continuous dynamics equations requiring numerical integration, the UKF does take significantly longer to compute. In the case of this work, about a factor of six.

The Kalman filter has an observer structure consisting of a prediction and a correction part. Although in general multiple prediction steps may be performed between correction updates, the hardware provides both accelerometer and optic flow at the same rate. So a prediction based on accelerometer data is always followed by a correction with optic flow from the same recording time in our implementation.

This structure of the Kalman filter lends itself to a simple combination of two filters. The prediction part yields a one-step-ahead state prediction and predicted covariance which can then be processed with the correction part of another algorithm.

As mentioned above, there is a trade-off between accuracy and computation time when comparing the (I)EKF with the UKF. The UKF does give better results however, so it may be interesting to see whether the performance gain is achieved mainly in the prediction part or in the correction part. In order to investigate this, the Hybrid Kalman Filter (HKF) algorithm is introduced. It uses the UKF prediction combined with the IEKF correction step.

If the performance of this filter is much like the UKF, it may be concluded that the largest performance gain is due to the use of the UT in the prediction part. If, on the other hand, the HKF performs much like the IEKF, then the correction part is the most affected by the use of the UT. It has to be noted that these results will have limited generalization potential as research results in literature show a large dependence of algorithm performance on the specific problem they are solving.

V. Simulation

The first goal of the simulations is to explore the problem in general. Defining the sensor configuration has to be done based on which states should be estimated and the sensitivity of the observability of those states to properties of the environment and the sensors themselves. This includes e.g. the geometry and scale of the room, sensor noise characteristics and camera field of view. The simulation should also provide a convenient and fast way to test various algorithms for filtering the required state variables from the sensor data. Finally, the simulations can be used to choose an experiment setup. It should give an idea whether the experiment will be able to show the performance differences between the filter algorithms.

The algorithms developed for filtering can later be used in the hardware-in-the-loop case. To this end, it is important that the simulated data resembles the hardware as closely as possible. This has been done by concurrent development of the simulation and the hardware. In the first stage the simulation was used to define the concept and later on the simulation has been refined using knowledge about e.g. the sensor properties gained from building the hardware.

The overall structure of the simulation is shown in Fig. 2a. *Data generation* provides simulated sensor output \vec{y}_d and controls \vec{u} for the *Kalman filter* plus truth data \vec{x} to compare with the estimated state $\hat{\vec{x}}$ in the *performance gauge*. It has been implemented in Simulink^a using an S-function to accelerate the Kalman filter. To process hardware datasets from the experiment, the *data generation* subsystem could easily be

^aThe MathWorks, Inc.

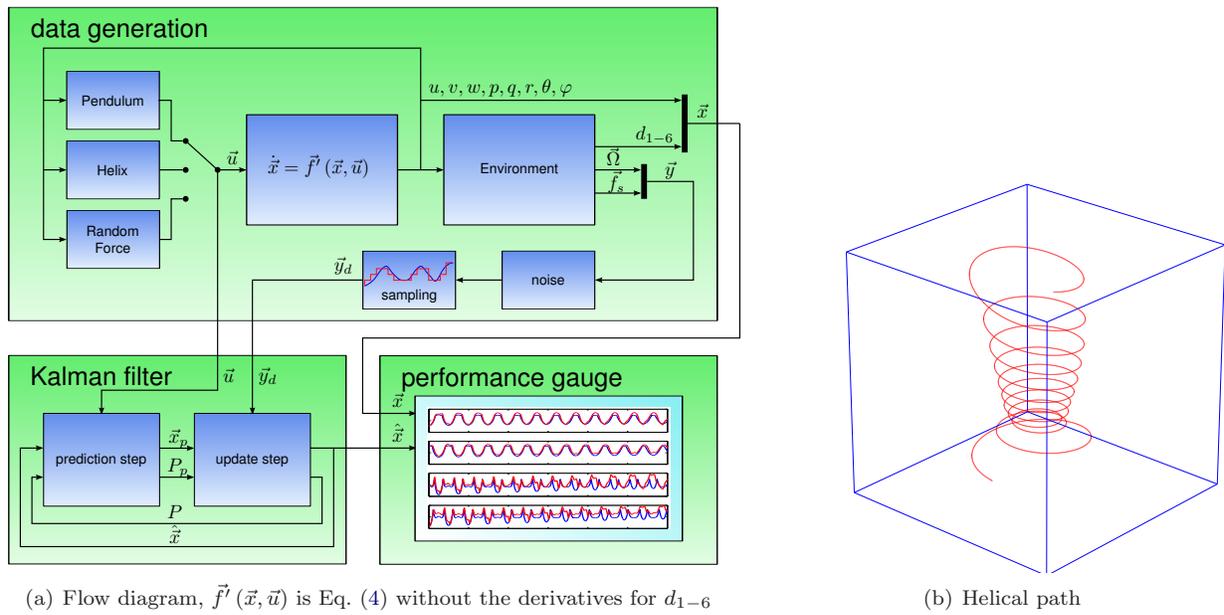


Figure 2. Simulation details.

exchanged for a serial interface.

The *data generation* includes three options for different trajectory generation. The helical path, shown in Fig. 2b, has varying orbital radius and descends from ceiling to floor with uncoordinated body rotations, whereas the pendulum resembles the hardware experiment. This includes the sensor board attitude with respect to the pendulum and its distance to the pendulum center of gravity as explained in subsection VI.B. Since the work is conceptual in nature, no specific MAV has been chosen. Rather, only the inertias have been defined and control forces and moments are generated directly without aerodynamic model.

After the vehicle state has been generated, the optic flow sensors are simulated using Eq. (1) and a model of a convex rectangular room to determine the viewing distances. Here assumptions regarding sufficient texture and lighting are made. Also, the optic flow along the optical axis has been taken without regarding the sensor field of view. This is an accurate approximation of the mean optic flow value generated by the real sensors as long as the field of view is sufficiently narrow and no severe distance discontinuities appear in it. The field of view angle defined by the hardware configuration is 10.9° .

Measurement noise has been modeled by adding a (band-limited) Gaussian noise term to the signals. The magnitude of the noise has been set at 10% of the signal power.

Another important mechanism altering the ideal signals is the fact that the sensors have a digital output, which, in the case of the optic flow cameras, originates in the array of photoreceptors and the digital signal processor (DSP). The accelerometers generate continuous signals which are subsequently sampled in an analog-to-digital converter (ADC). Both types of sensor output can be simulated by quantizing the continuous signals. The rounding should be towards the zero. The discrete signals are encoded using a finite number of bits. This means signal saturation may occur.

Finally, the simulated sensor sample rates (25 - 50 Hz) are lower than the simulation has been run at to ensure accurate numerical integration (3000 Hz). No low pass filtering has been applied prior to downsampling in the simulation, because no indication has been found that the hardware does apply an analog low pass filter before sampling. This means that high frequency signal content may show up in the sampled signal as low frequency oscillations which are not present in the real process. This effect is known as aliasing and it is an additional disturbance which has been intentionally simulated, because it is expected to be present in the hardware as well.

VI. Hardware

A hardware experiment in a controlled environment is an invaluable tool to better understand the properties of the sensor concept and evaluate its performance. Simulation has the advantage of knowing the “truth” states, however one can not be sure whether all relevant details of a system have been captured by the model. Especially in this case, where very limited prior knowledge about the optic flow sensors existed.

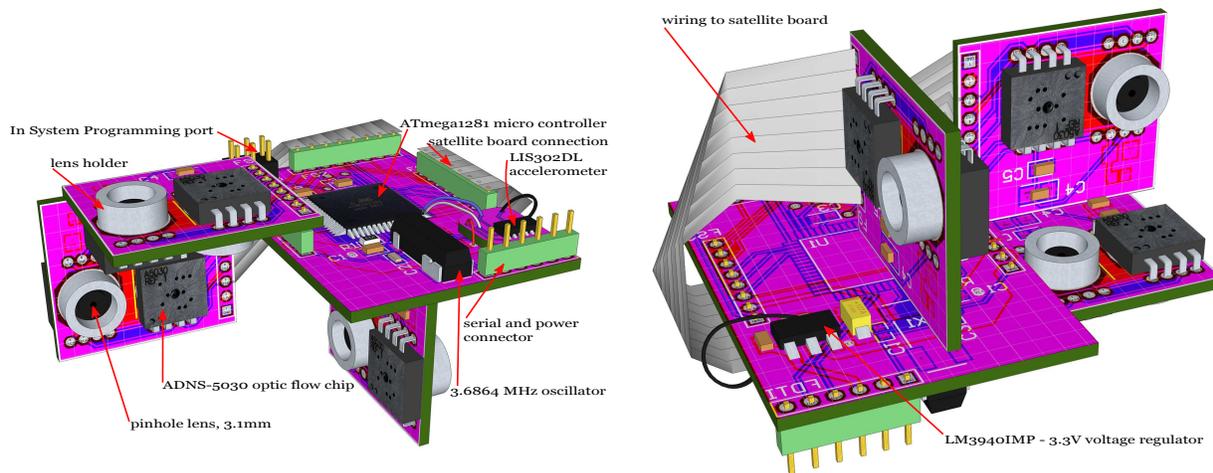


Figure 3. 3D render of the sensor board with 6 optic flow sensors and a 3-axis accelerometer

VI.A. Hardware Description

The hardware, which was built for the experiment, is shown in Fig. 3. Its overall dimensions are 66 mm × 51 mm × 41 mm. There are four double sided copper circuit boards consisting of two copper layers with an insulating polymer substrate in between them. Three of them are identical and contain the optic flow cameras. The fourth connects the sensors to the micro-controller (μC), the μC to an external computer via serial link and it provides a clock signal for the μC and a DC voltage conversion. In addition, a small board containing the 3-axis accelerometer has been glued onto the main board such that the accelerometer axes are parallel to the optical axes of the optic flow cameras. The accelerometer is connected to two pins of the μC using an I²C serial interface.

The central component of the system is an AVR ATmega1281 μC with an 8-bit Reduced Instruction Set Computer (RISC) architecture. There is almost no capacity for on board processing and the μC is only used for collecting measurement data and forwarding this to the host PC. It is possible to do this at precise intervals using the on board counters which can act as timers and run independently of the CPU.

The six ADNS-5030 optic flow ICs are mounted on three satellite boards. The configuration is designed such that the sensor pairs on each board look in opposite directions and the satellite boards point in three mutually orthogonal directions. They are connected to the μC using a 4-wire serial bus using the SPI protocol. To get the optic flow data from all six sensors takes 4.18 milliseconds. Strictly, data are not recorded at the same time instant, but the small time delay between the sensors has been neglected in subsequent processing.

The ADNS-5030 output consists of two additive registers recording the total angular displacements since

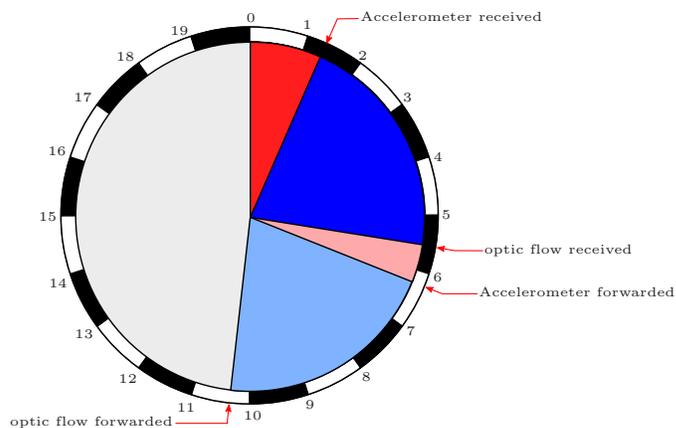


Figure 4. Sampling cycle pie chart, the numbers represent time in milliseconds from the start of the cycle.

they were last read. The resulting optic flow step size may be expressed in terms of the angular step size $\delta\alpha$ and the sample time step dt_s as follows:

$$\delta\Omega = \frac{\delta\alpha}{dt_s} \quad (9)$$

For the chosen optics - sensor combination $\delta\alpha = 5.18 \cdot 10^{-3}$ rad e.g. choosing $dt_s = 0.02$ s gives $\delta\Omega = 0.274$ rad/s = $15.7^\circ/s$, while choosing $dt_s = 0.05$ s gives $\delta\Omega = 0.110$ rad/s = $6.28^\circ/s$. $\delta\Omega$ is the smallest measurable optic flow quantity. Clearly, there is a trade-off between high sample rates and good sample resolution. This trade-off has been investigated by running the pendulum experiment at several sample rates.

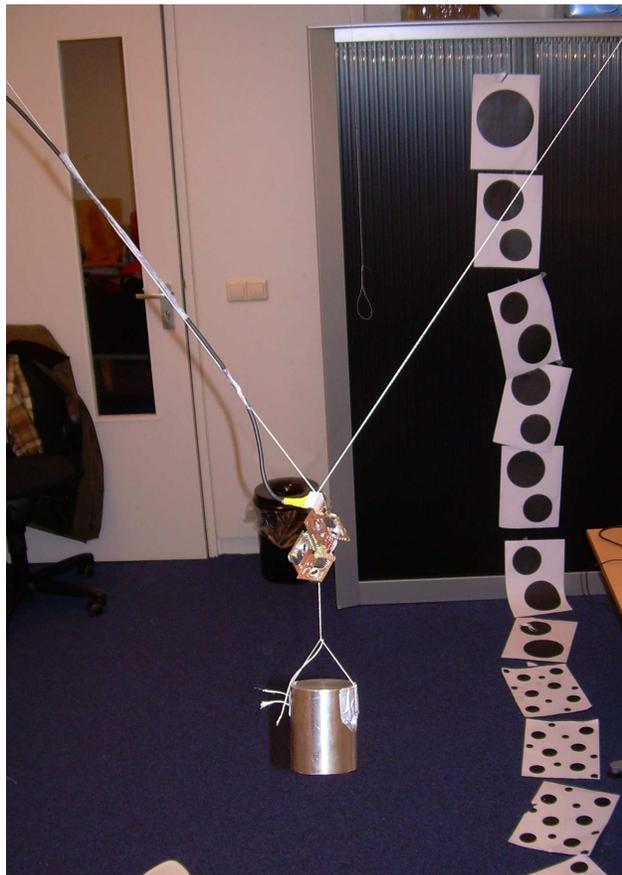


Figure 5. The experiment setup.

The 3-axis accelerometer on the sensor board is the LIS302DL. It is a compact (3 mm × 5 mm × 0.9 mm) MEMS device with built-in ADCs. This means that the output is stored in digital registers as signed 8-bit integers, like in the case of the ADNS-5030. There are two sensitivity settings available, namely ± 2 g or ± 8 g. These are actually minimum values and in reality the values are typically ± 2.3 g or ± 9.2 g, respectively. For this application, ± 2 g is the most appropriate setting, because a high sensitivity is critical for good filter performance. The signed 8-bit integer encoding means that the sampling step size is about 18 mg.

Also, as explained in section V, aliasing may be an issue with this sensor, as there is no mention of any low pass filter being applied. Only a high pass filter may be enabled, but this does not solve aliasing and is not desirable for this application. An I²C interface connects the accelerometer with the μ C. In this case, the I²C has been implemented in software using two general purpose I/O pins on the μ C. It is generally slower since the CPU has to handle all operations, while a hardware peripheral implementation can run parallel to other CPU tasks. That is not a problem however, since the data rate is limited by the consideration of optic flow resolution ($\delta\Omega$) which leaves enough time to handle software I²C for

the accelerometer.

As mentioned before, the pendulum experiment has been run at several values of sample timestep dt_s . The shortest possible dt_s is determined by the time it takes to get the data from the sensors and to send it to the PC. This takes on average 38227 clock cycles which at 3.6864 MHz amounts to 10.37 ms or a sample frequency of 96 Hz. However, as explained above, the optic flow resolution ($\delta\Omega$) depends on the sample step dt_s . It is therefore important to have control over the time between readouts of the sensors. To achieve this, one of the on board counters has been used. Fig. 4 shows the situation when $dt_s = 0.02$ s. Each segment amounts to the time spent on one task with respect to the duration of the entire sampling cycle.

VI.B. Experiment Setup

The experiment's aim is to investigate the optic flow sensor concept through a hardware implementation in a realistic environment. The environment is an office room. Its dimensions are approximately 3 m × 4 m × 2.6 m and it is lit by TL tubes. The motion pattern was chosen to be a pendulum for the following reasons:

- It has to be a well-reproducible motion with easily identifiable pattern and parameters for validation purposes. The pendulum produces a sinusoidal motion pattern with known period and decreasing amplitude. This can be easily recognized in the estimated states and it can be simulated to provide a comparison.
- The pendulum is autonomous, i.e. it requires no external input after the motion has started. This means that the moments acting on the pendulum mass are only a function of the state and thus do not have to be measured, which simplifies the experiment and eliminates a noise source.
- The pendulum provides changing velocities and rotation in the range of a typical indoor MAV with momentary zero velocity points at the extremes of the pendulum swing arc. This should provide interesting data to test the Kalman filters and check the observability.

Figure 5 shows the setup. The pendulum consists of two strings attached to the ceiling at one end and to each other at the lower end. The angle between the two strings ensures that the motion stays within one plane. A short string connects the triangle with a mass. This mass is a solid steel cylinder weighing 1.13 kg. It ensures that the pendulum keeps oscillating long enough to record a good dataset. The effective pendulum arm is 1.45 m.

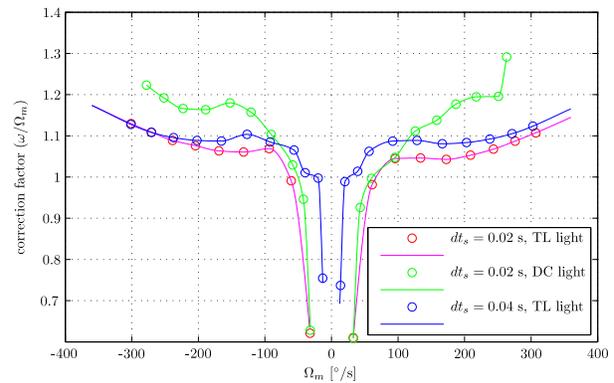
The sensor board is connected to the pendulum at the string junction and also to the data cable, which runs along one of the strings to the ceiling and then to the laptop PC. Its attitude with respect to the pendulum is fixed.

As can be seen in Fig. 5, high contrast texture has been applied along the camera trajectories. This was necessary to stimulate the optic flow sensors. An immediate conclusion which may drawn from this is that this particular optics - sensor combination is not suited for use in a general indoor environment. However, the goal of this work is to explore a concept and future work could substitute other sensors.

VI.C. Sensor Calibration



(a) Calibration setup.



(b) Resulting curve.

Figure 6. The ADNS-5030 optic flow calibration.

The value of $\delta\alpha$ mentioned in subsection VI.A has been calculated using the datasheet²⁴ and the geometry of the custom lens assembly and may only serve as guideline. To calibrate the optic flow sensors, a calibration procedure was performed using a single axis calibration turning table as shown in Fig. 6a. The sensor has been placed on the axis of rotation such that the optic flow is purely rotational. A texture of black dots on a white background is presented to provide optimal feature conditions.

Turning the table at a series of constant rates ω has yielded a graph of the calibration factor ω/Ω_m to be applied to correct the measured, uncalibrated optic flow output Ω_m over the range of optic flow rates which is expected to occur during the experiment. The results are shown in Fig. 6b. Ω_m multiplied by the correction factor yields the calibrated optic flow. It includes three conditions: fluorescent tube lamp (TL) light with two sample rates (50 Hz and 25 Hz) and a DC light source at 50 Hz sample rate. The TL light is the main light source in the office room used for the pendulum experiment. The DC light was included, because the sensor might be affected by the AC frequency of 50 Hz. However, the sensors have a much higher adaptive internal processing frequency which cannot be set or read by the micro controller. This results in

a more complex behavior than originally expected. It is not clear whether the sensor suffers from a 50 Hz light source, as more frequencies would have to be measured to get a consistent picture as to the response to light intensity variations. That was however not the goal of the calibration procedure and thus has not been investigated further.

The results from the constant intensity light source are of lower quality because the surface was not sufficiently lit. This illustrates the fact that low light levels and high optic flow values cause the sensor to not count some image movement. Performance has turned out to be mainly dependent on effective mean light intensity and texture contrast at the scale of the sensor image elements. The sensor shows a steep increase in optic flow scale factor for low Ω , which is directly proportional to the sample rate. Presumably this is an artifact of the quantization of the signal, but since the image processing inside the ADNS-5030 is proprietary, it is hard to say with certainty what causes this nonlinearity.

For the accelerometer a calibration procedure based on the local gravitational acceleration has been applied before each experiment session to determine the current sensitivity and bias for each of the three axes. It is assumed that the sensitivity and bias values remain constant during the experiment session following a calibration. Therefore the data processing uses these values as constants and does not attempt to estimate them from the data itself.

VII. Results

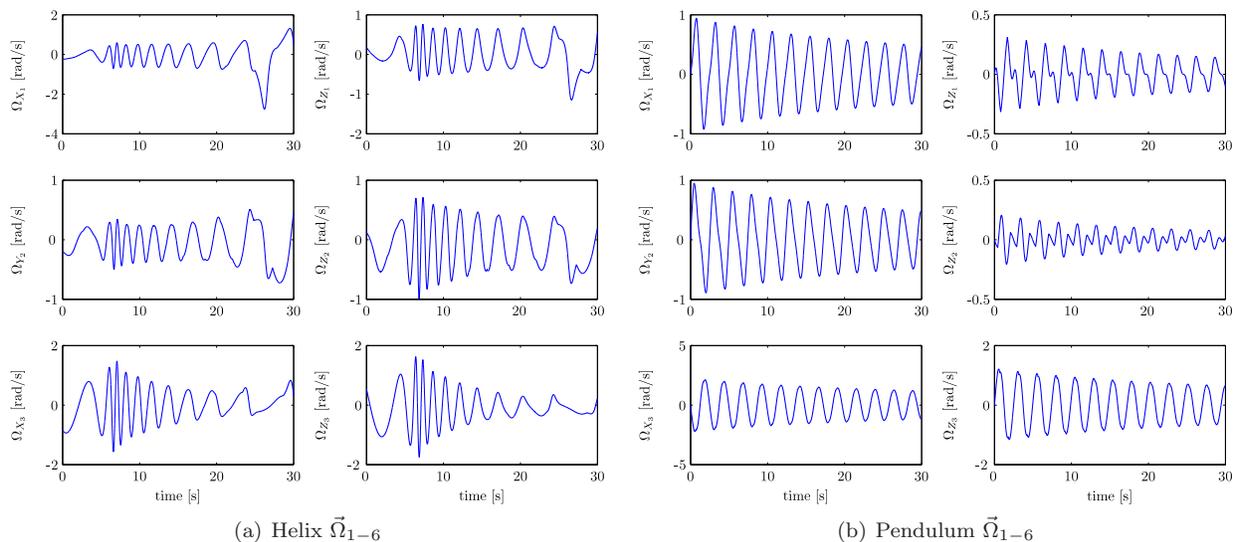


Figure 7. Simulated optic flow sensor signals before noise and downsampling.

Three filter algorithms have been tested using simulated data: IEKF, HKF and UKF as described in section IV. Two input conditions, namely the helical path and the pendulum motion have been applied to the filters.

The data from the two motion cases has been corrupted by measurement noise. It is band-limited white noise, generated by Simulink and the standard deviation is 10% of each signal power. Four sampling cases have been defined: case A uses a very high sampling frequency of 3000 Hz and double precision numerical format (64 bits), case B has been down-sampled to 50 Hz with double precision, case C is 50 Hz and 8-bit precision and case D is 20 Hz and 8-bit. The last two cases use the same sampling rates and precision used by the ATmega1281 microcontroller and can therefore serve as a test to compare the simulated pendulum case with the hardware experiment. The initial conditions have been chosen as a generic estimate assuming some idea of the dimensions of the environment and the vehicle motion.

The resulting sensor output generated by the helix and pendulum simulations is shown in Fig. 7. They show the clean sensor output without any noise or downsampling applied. The helix motion provides mea-

sured moments as inputs, while the pendulum is an autonomous motion, i.e. there is no external input once the motion has started. With the pendulum, the Kalman filters have to work without moments being supplied as known quantities. However, two moments (\mathcal{M}_x and \mathcal{M}_z) are zero and \mathcal{M}_y follows from the states and has been included in Eq. (4). This facilitates the experiment, simplifying measurements and repeatability. Other than concerning the moments, the filter algorithms have not been modified to account for planar motion in the pendulum case. The out-of-plane attitude of the sensor board with respect to the pendulum plane of motion results in sufficient signals on all optic flow components and specific forces.

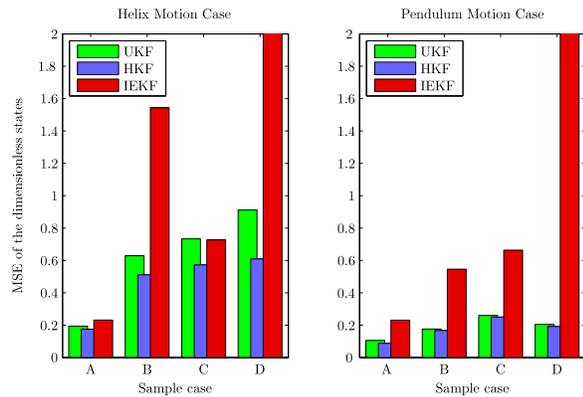
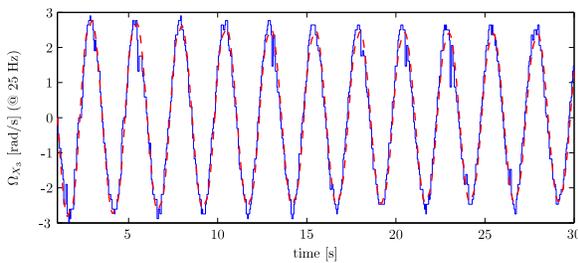


Figure 8. Simulation results summarized as Mean Squared Errors.

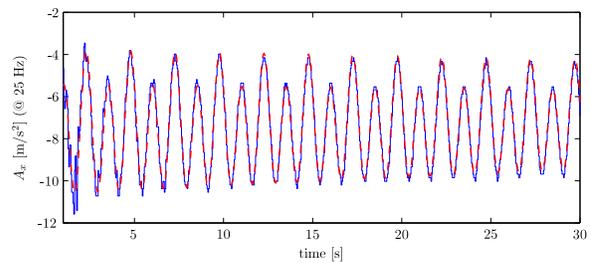
Figure 8 provides an overview of the performance of the filters in terms of mean squared error (MSE) of the dimensionless signals. All estimated states have been made dimensionless through division by the mean square of the corresponding simulated states. The mean values presented in the figure are composed of all elements of the state vector. This condenses the information, but each individual value has little meaning as such. Rather, the idea is to compare the MSEs between filters and sample cases. This gives a quick indication of the relative performance of the filters under the various conditions.

The first observation is that the IEKF tends to produce much larger errors than UKF/HKF. The presented results represent a single dataset for each sample case. Other data sets with different noise realizations and initial conditions show a similar picture.

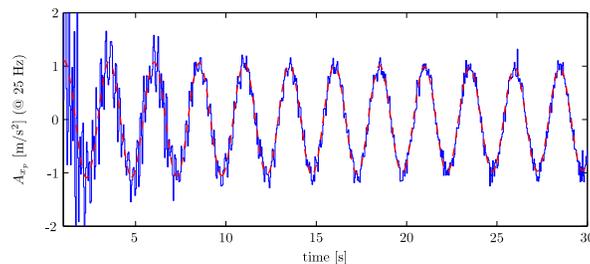
There is an anomaly in the IEKF performance for the helix sample case C however. This is an interesting result which may be explained as follows. As mentioned in section IV, the IEKF is known to have a narrow convergence domain due to the errors caused by linearization. Between datasets, and especially for the helical path, the IEKF has shown strongly varying performance whereas the other two algorithms are much more consistent. Case C represents a dataset where the IEKF happened to come close enough to the true states to remain within its convergence zone. Although this case most likely represents a statistically outlying result, it has been included as an example of the more inconsistent behavior of the IEKF.



(a) Optic flow of sensor 3 in sensor board X-direction.



(b) Specific force component in the sensor board X-direction.



(c) Specific force component in the pendulum X-direction.

Figure 9. Sample of the hardware output. The hardware signals (solid lines) are compared to simulated data (dashed).

The MSE suggests an answer to the question posed in section IV, namely which part of the UKF filter provides the largest contribution to the performance gain observed between IEKF and UKF. The HKF has similar MSE as the UKF whereas the IEKF has a much larger MSE. The major difference is the way in which the predicted covariance matrix $P(k+1|k)$ is calculated: the EKF makes one function evaluation and uses that in the Jacobian of the dynamics F_x to calculate $P(k+1|k)$. This ignores the second and higher order terms of the Taylor series expansion of $f(\vec{x}, \vec{u})$ with respect to the state \vec{x} , thus linearizing the relation. The UKF on the other hand evaluates the dynamics for a well defined set of perturbations of the state vector. The resulting predictions for the perturbed states form a transformed set from which $P(k+1|k)$ can be reconstructed. This method is much more computationally expensive as it requires many numerical integrations per filter timestep, but it yields a more accurate one-step-ahead prediction. This must explain the difference in filter performance as it is the only difference between the IEKF and HKF. The addition of the UT in the update part of the UKF does not improve the results, rather the HKF shows a slightly better performance. The iterations in the update part may be the cause of this improvement. It is apparent that the nonlinear nature of the observation equations does not provide a performance advantage to the UT based update part. This is an unexpected finding. Julier and Uhlmann²³ describe a benefit for systems involving coordinate transformations. The observation equations in this work also involve a transformation from body frame to the rotated sensor frame. Moreover, the prediction does not provide any information on the distances, which are the hardest states to estimate. So these have to be estimated entirely by the update part. Nevertheless, the iterated linearized algorithm of the IEKF, using the predicted state and covariance from the UKF, does this even a little better than the UT based algorithm. Whether the small performance difference is significant cannot be concluded from these results however.

An example of the filtered and calibrated hardware output is shown in Fig. 9. It shows an optic flow component and specific force components. The hardware output is compared to data (without noise) from the simulations. Figure 9b shows the accelerometer output in its X-direction. The oblique sensor board attitude causes an alternating peak pattern which is replicated by the simulation. As expected, this effect of the geometrical configuration cannot be observed on A_{x_p} in Fig. 9c, which is a component of specific force rotated to the pendulum plane of motion using the measured sensor board attitude. This comparison between the experiment and the simulation permits to validate the simulation results.

Due to the lack of “truth” data in the pendulum experiment it is hard to apply a quantitative performance metric to the resulting state estimates. Some qualitative observations can be made however. In general, the period of the pendulum motion is well recognizable in the results. The IEKF becomes unstable however and has clearly diverged in some cases. It has a strong tendency to produce negative distance estimates and the addition of a condition that the distances can only be positive did not solve the instability. Each time a distance value becomes arbitrarily small, the filter innovation becomes very inaccurate because the predicted optic flow components are calculated by dividing velocity components by distances, producing very large corrections through the innovation. These have a destabilizing effect on the solution.

Comparing the UKF and HKF, the first observation is that they produce very similar results as has been noted in the results discussion of the simulations as well. The common difference with the IEKF is the prediction step, so this part of the filter appears to have the larger influence on the state estimates. The UKF uses the UT for the correction part of the filter as well, but this does not produce a big performance gain over the HKF as has also been noted and explained in the simulation results discussion.

In order to get some insight into the observability condition of the hardware data, the condition number \mathcal{C}_{O_1} , which is defined in section III, has been plotted together with the HKF estimate for u in Fig. 10. The

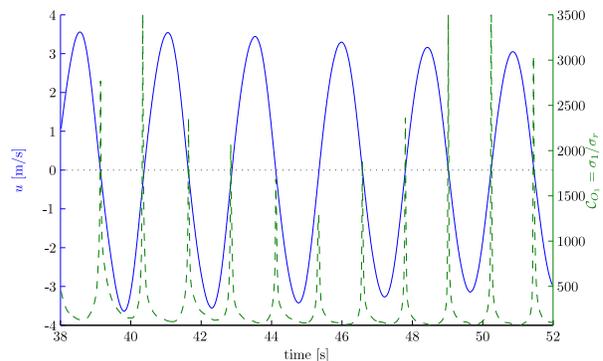


Figure 10. HKF velocity estimate (solid) and observability condition \mathcal{C}_{O_1} (dashed) from the hardware experiment.

best observability condition occurs when $\mathcal{C}_{\mathcal{O}_1} = 1$ and higher means harder to observe. The peaks in the $\mathcal{C}_{\mathcal{O}_1}$ -plot coincide with the moments where the velocity crosses zero and the pendulum reverses direction. Theoretically those points are not observable. In practice, the sensors are never sampled at the exact moment of the direction reversal. Moreover, the measurements are too noisy and inaccurate to find a value of the rank of \mathcal{O}_1 lower than 14 (full rank). So the rank condition for observability does not give much information in practice. Therefore the peaks in $\mathcal{C}_{\mathcal{O}_1}$ are a good indicator to signal possible observability problems. To be sure that the rank condition is met, the other condition number presented in section III, namely $\underline{\mu}(\mathcal{O}_1)$, is useful.

If $\mathcal{C}_{\mathcal{O}_1}$ would stay above a certain threshold for a longer time, the Kalman filter might well diverge. In this case, temporarily not updating the state could be a way to improve filter stability. The state would only be predicted based on accelerometer input during this time.

VIII. Conclusions

It has been the goal of this work to investigate a state estimation concept for indoor MAVs driven by the requirements of low weight and independence of GPS. Also, it should be straightforward to build hardware for a laboratory experiment. In this way, the theory and simulation results have been compared to performance in an experiment. Although the experiment lacked “truth” data for the estimated states, similarity with the simulated outputs gives a basis for qualitative comparison.

Observability analysis, under the assumption that Eq. (8) holds, indicates that the concept with 6 orthogonally mounted optic flow sensors and 3 accelerometers should provide sufficient output to observe the platform motion (u, v, w, p, q, r) , the sensor-obstacle distances $(d_1 - d_6)$ and two attitude angles (θ, φ) as long as the platform has some translational and rotational motion relative to a stationary environment. Theoretically, the motion may also not be exactly aligned with the sensor board. In practice, this does not have to be a problem when some external disturbances ensure enough change in direction. Also, a momentary low observability condition (high $\mathcal{C}_{\mathcal{O}_1}$) is not a problem for convergence of the solution as shown by the pendulum case.

Both the simulation and the hardware experiment results clearly show that $\mathcal{C}_{\mathcal{O}_1}$ peaks (indicating a hard to observe motion state) when the velocity approaches zero. The error of the pendulum velocity estimate may be driven by the observability condition. The standard deviations show a correlation with $\mathcal{C}_{\mathcal{O}_1}$, indicating that filter performance is impacted by the observability condition. This property of $\mathcal{C}_{\mathcal{O}_1}$ could be used as a criterion for deciding whether to proceed to the correction step. It might prevent sensor data from deteriorating the state estimates in moments of insufficient observability.

The result of the simulation clearly indicates that the use of the Unscented Transform (UT) in a Kalman filter yields better estimates for this problem. The largest effect is achieved in the prediction step. The use of the UT in the update step, as is implemented in the UKF, does not provide clear performance benefits. This is an unexpected finding as the nonlinear nature of the observation equations, including coordinate transformation and velocities and distances only appearing as ratios, would suggest that an algorithm with higher order probability distribution capture would produce better results. The HKF, using the iterated linearized update step with UT-based predicted state and covariance, appears to yield the best results.

The convergence zone of the IEKF is smaller than that of the other algorithms, so the initial conditions have to be chosen closer to the true values to get converging results. The cost of the improved accuracy of the UT-based filters is about 6 times longer computation time. Processing of the sensor output is done off board on a PC and it would require a relatively powerful on board computer. This is a drawback of the full six degree of freedom state estimation approach based on optic flow.

Theory indicates that the problem is observable given enough motion. However, the theoretical statement that an estimation problem is observable only provides a starting point. In principle, the solution can be reconstructed from the observations, given perfect signals and an exact process model. It says nothing about the achievable quality of the data. In this case, both the optic flow sensors and accelerometers are low-end consumer-grade devices. The optic flow sensors have been designed for the rather constrained environment of computer mice and they turned out not to be well suited for a general indoor environment. This is mainly due to the widely varying lighting conditions and textures. To yield useful results, the texture presented to the optic flow sensors had to have high contrast and contain suitably sized features. This is partly due to the choice of the custom lens. A larger aperture and longer focal length will likely improve performance, at the expense of increased overall size and weight.

Better quality sensors may raise the quality of the estimated state to useful levels. In principle, the UT based Kalman filter has shown to be capable of estimating the motion state of an indoor MAV using optic flow sensors and accelerometers. To address the observability in low motion cases, one option would be the addition of direct distance measurement through e.g. infrared-, sonar- or laser-based sensors. Instead nature suggests to adapt the guidance and control side such that the vehicle maintains enough optic flow input to remain in a well-observable state.

Research on flying insects indicates that a lot is possible. Their completely different data processing structure with highly parallel neuron network would be an interesting research topic. However, insects do not know their absolute velocities²⁵ and that is where this work has diverged from nature. It turned out to be difficult to estimate the velocity and distances as individual states. A promising research path is therefore to look at control concepts which can handle translational optic flow directly.

References

- ¹Achtelik, M., Bachrach, A., He, R., Prentice, S., and Roy, N., "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments," Vol. 7332, SPIE, 2009, p. 733219.
- ²IARC, "International Aerial Robotics Competition 2009 Website," <http://iarc.angel-strike.com/>, 2009.
- ³Barbour, N. M., "Inertial Navigation Sensors," *RTO-EN-SET-116*, NATO Research and Technology Organisation, May 2009.
- ⁴Aubépart, F. and Franceschini, N., "Bio-inspired Optic Flow Sensors based on FPGA: Application to Micro-Air-Vehicles," *Microprocessors and Microsystems*, Vol. 31, 2007, pp. 408 – 419.
- ⁵Zufferey, J. C. and Floreano, D., "Toward 30-gram Autonomous Indoor Aircraft: Vision-based Obstacle Avoidance and Altitude Control," *Proceedings of 2005 IEEE International Conference on Robotics and Automation*, IEEE, Barcelona, Spain, April 2005, p. 2594 – 2599.
- ⁶Neumann, T. R. and Bulthoff, H. H., *Insect Inspired Visual Control of Translatory Flight*, Springer-Verlag Berlin Heidelberg, 2001.
- ⁷Krapp, H. G. and Hengstenberg, R., "Estimation of Self-motion by Optic Flow Processing in Single Visual Interneurons," *Nature*, Vol. 384, No. 6608, December 1996, pp. 463 – 466.
- ⁸Conroy, J., Gremillion, G., Ranganathan, B., and Humbert, J. S., "Implementation of Wide-Field Integration of Optic Flow for Autonomous Quadrotor Navigation," *Autonomous Robots*, Vol. 27, No. 3, October 2009, pp. 189 – 198.
- ⁹Hyslop, A. M. and Humbert, J. S., "Autonomous Navigation in Three-Dimensional Urban Environments Using Wide-Field Integration of Optic Flow," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, January – February 2010, pp. 147 – 159.
- ¹⁰Ruffier, F. and Franceschini, N., "Optic Flow Regulation: the Key to Aircraft Automatic Guidance," *Robotics and Autonomous Systems*, Vol. 50, 2005, pp. 177 – 194.
- ¹¹Serres, J., Ruffier, F., and Franceschini, N., "Two Optic Flow Regulators for Speed Control and Obstacle Avoidance," *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Vol. 2006, No. 1639180, 2006, pp. 750 – 757.
- ¹²Zufferey, J.-C., Klaptocz, A., Beyeler, A., Nicoud, J.-D., and Floreano, D., "A 10-gram Vision-based Flying Robot," *Advanced Robotics*, Vol. 21, No. 14, 2007, pp. 1671 – 1684.
- ¹³Franceschini, N., Ruffier, F., and Serres, J., "A Bio-Inspired Flying Robot Sheds Light on Insect Piloting Abilities," *Current Biology*, Vol. 17, February 2007, pp. 329 – 335.
- ¹⁴Chen, Z., "Local Observability and its Application to Multiple Measurement Estimation," *IEEE Transactions on Industrial Electronics*, Vol. 38, No. 6, December 1991, pp. 491 – 496.
- ¹⁵Driels, M. R. and Pathre, U. S., "Significance of Observation Strategy on the Design of Robot Calibration Experiments," *Journal of Robotic Systems*, Vol. 7, No. 2, April 1990, pp. 197 – 223.
- ¹⁶Hong, S., Chun, H. H., Kwon, S. H., and Lee, M. H., "Observability Measures and their Application to GPS/INS," *IEEE Transactions on Vehicular Technology*, Vol. 57, No. 1, January 2008, pp. 97 – 106.
- ¹⁷Golub, G. H. and Loan, C. F. V., *Matrix Computations*, The Johns Hopkins Univ. Press, Baltimore, MD, 3rd ed., 1996.
- ¹⁸Kushner, H. J., "Dynamical Equations for Optimal Nonlinear Filtering," *Journal of Differential Equations*, Vol. 3, No. 2, April 1967, pp. 179 – 190.
- ¹⁹Breakwell, J. V., "Estimation with Slight Non-Linearity," Unpublished communication, TWR Systems.
- ²⁰Denham, W. F. and Pines, S., "Sequential Estimation when Measurement Function Nonlinearity is Comparable to Measurement Error," *AIAA J.*, Vol. 4, No. 6, 1966, pp. 1071 – 1076.
- ²¹Jazwinski, A. H., *Stochastic Processes and Filter Theory*, Vol. 64, Academic Press, 1970, Mathematics in Science and Engineering series.
- ²²Julier, S. J. and Uhlmann, J. K., "A New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of SPIE*, Vol. 3068, 1997, pp. 182 – 193.
- ²³Julier, S. and Uhlmann, J. K., *Handbook of Multisensor Data Fusion*, chap. 13: Data Fusion in Nonlinear Systems, CRC Press LLC, 2001.
- ²⁴Avago, "ADNS-5030 Datasheet," microcontroller manual, Avago Technologies Ltd., September 2008, <http://www.avagotech.com/docs/AV02-0113EN>.
- ²⁵Srinivasan, M. V., "Small Brains, Smart Computations: Vision and Navigation in Honeybees, and Applications to Robotics," *International Congress Series*, Vol. 1291, 2006, pp. 30 – 37.

²⁶Hedrick, J. K. and Girard, A., *Control of Nonlinear Dynamic Systems: Theory and Applications*, chap. 6, Lecture Notes ME 237, University of California at Berkeley, 2005.

²⁷Hermann, R. and Krener, A., “Nonlinear Controllability and Observability,” *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 5, October 1977, pp. 728 – 740.

²⁸Julier, S. J. and Uhlmann, J. K., “Unscented Filtering and Nonlinear Estimation,” *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004, pp. 401 – 422.

Appendix

A. Nonlinear Observability Concepts

This appendix section describes the concept of observability for nonlinear state space systems:

$$\Sigma : \begin{cases} \dot{\vec{x}} = \vec{f}(\vec{x}, \vec{u}) \\ \vec{y} = \vec{h}(\vec{x}) \end{cases}, \quad (10)$$

where $\vec{u} \in L$, a subset of \mathbb{R}^m , $\vec{x} \in M$, the state space with dimension n , $\vec{y} \in \mathbb{R}^p$ and \vec{f} and \vec{h} are vector functions of appropriate dimensions.

Whereas in the linear case the observability condition holds for the entire domain of \vec{x} , for nonlinear systems observability is state dependent and has to be determined locally.²⁶ First, the definition of *distinguishability* is required:

Definition 3 Given Σ . Two states \vec{x}_0 and \vec{x}_1 are distinguishable if and only if there exists an input function \vec{u}^* such that: $\vec{h}(\vec{x}_0) \neq \vec{h}(\vec{x}_1)$.

Local observability can then be defined as follows:

Definition 4 Σ is locally observable at \vec{x}_0 if and only if there exists a neighborhood of \vec{x}_0 such that every \vec{x} in that neighborhood other than \vec{x}_0 is distinguishable from \vec{x}_0 .

To test whether a system is locally observable at \vec{x}_0 , an observability matrix similar to the linear case can be constructed. This is defined in terms of the Lie derivative of a scalar function g in \vec{x} with respect to a vector field \vec{f} also in \vec{x} , which is denoted as $L_f(g) = \frac{\partial g}{\partial \vec{x}} \vec{f}$. Higher order Lie derivatives are defined by recursive application of the first order Lie derivative. Finally, $L_f^0(g) \equiv g$. The matrix to test for local observability of Σ at \vec{x}_0 for some constant control \vec{u}^* is defined by

$$\mathcal{O}(\vec{x}_0, \vec{u}^*) \triangleq \left[\begin{array}{c} dL_f^0(h_1) \\ \vdots \\ dL_f^0(h_p) \\ \vdots \\ dL_f^{n-1}(h_1) \\ \vdots \\ dL_f^{n-1}(h_p) \end{array} \right]_{\vec{x}=\vec{x}_0, \vec{u}=\vec{u}^*}, \quad (11)$$

where d is the gradient operator with respect to \vec{x} , $\vec{h}(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ and h_i is the i^{th} element of \vec{h} . Σ is said to satisfy the observability rank condition at \vec{x}_0 if the dimension of the image of $\mathcal{O}(\vec{x}_0, \vec{u}^*)$ is equal to n . The following theorem from Ref. 27 establishes how to test for local observability using this condition.

Theorem 2 If and only if Σ satisfies the observability rank condition at \vec{x}_0 then Σ is locally observable at \vec{x}_0 .

B. Kalman Filter Algorithms

Extended Kalman Filter

1. The one step ahead prediction:

$$\hat{x}(k+1|k) = \hat{x}(k|k) + \int_{t_k}^{t_{k+1}} f(\hat{x}(t|t_k), \bar{u}(t), t) dt . \quad (12)$$

2. The error covariance matrix of the prediction:

$$P(k+1|k) = \Phi(k) P(k|k) \Phi^T(k) + Q_d , \quad (13)$$

where

$$\begin{aligned} \Phi(k) &= \exp(F_x(k) \cdot (t_{k+1} - t_k)) , \\ F_x(k) &= \frac{\partial}{\partial x} f(\hat{x}(k|k), u(k), k) , \\ Q_d &= \Gamma Q \Gamma^T, \quad \Gamma = \frac{\partial f}{\partial u} . \end{aligned}$$

3. The Kalman gain matrix:

$$\begin{aligned} K(k+1) &= P(k+1|k) H_x^T(k+1) \\ &\times \{H_x(k+1) P(k+1|k) H_x^T(k+1) + R\}^{-1} , \end{aligned} \quad (14)$$

where

$$H_x(k+1) = \frac{\partial}{\partial x} h(\hat{x}(k+1|k), u(k+1), k+1) .$$

4. The measurement update equation:

$$\begin{aligned} \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + K(k+1) \\ &\times \{z(k+1) - h(\hat{x}(k+1|k), u(k+1), k+1)\} . \end{aligned} \quad (15)$$

5. The error covariance matrix of the state estimate

$$\begin{aligned} P(k+1|k+1) &= [I_n - K(k+1) H_x(k+1)] P(k+1|k) \\ &\times [I_n - K(k+1) H_x(k+1)]^T + K(k+1) R K^T(k+1) . \end{aligned} \quad (16)$$

Iterated Extended Kalman Filter

The Iterated Extended Kalman Filter (IEKF) is very similar to the EKF, the difference being that a local iteration of the measurement update is performed, re-linearizing about the updated state candidate η_i to compute a new updated state candidate which is presumably closer to the true state. The iteration may continue until the difference between two consecutive iteration steps is below a threshold ε or a maximum number of iterations is reached. The algorithm is obtained by replacing EKF step 3 to 4 by steps 3 to 4 below. At each time step, initialize $i = 1$ and $\eta_1 = \hat{x}(k+1|k)$.

3. Apply Equation (14) with

$$H_x(k+1) = \frac{\partial}{\partial x} h(\eta_i, u(k+1), k+1) .$$

- 4a. Let η_i be the i th iteration to the updated state, then

$$\begin{aligned} \eta_{i+1} &= \hat{x}(k+1|k) + K(k+1) \\ &\times [z(k+1) - h(\eta_i, u(k+1), k+1) - H_x(k+1, \eta_i) \{\hat{x}(k+1|k) - \eta_i\}] . \end{aligned} \quad (17)$$

- 4b. The stopping criterion is:

$$\frac{\|\eta_{i+1} - \eta_i\|_{\infty}}{\|\eta_i\|_{\infty}} < \varepsilon . \quad (18)$$

If Eq. (18) is not met then increment i and go back to step 3. Else $\hat{x}(k+1|k+1) = \eta_{i+1}$ and perform EKF step 5.

Unscented Kalman Filter

1. The state estimate and covariance are augmented with the mean and covariance of the process noise:

$$x^a(k|k) = \begin{bmatrix} \hat{x}^T(k|k) & E[w^T(k+1)] \end{bmatrix}^T ,$$

$$P^a(k|k) = \begin{bmatrix} P(k|k) & 0 \\ 0 & Q \end{bmatrix} .$$

2. A set of $2L + 1$ prediction sigma points is derived from the augmented state and covariance where L is the dimension of the augmented state:

$$\chi_{p,0}(k|k) = x^a(k|k) ,$$

$$\chi_{p,i}(k|k) = x^a(k|k) + \left(\sqrt{(L+\lambda)P^a(k|k)} \right)_i , \text{ for } i = 1 \dots L ,$$

$$\chi_{p,i}(k|k) = x^a(k|k) - \left(\sqrt{(L+\lambda)P^a(k|k)} \right)_{i-L} , \text{ for } i = (L+1) \dots 2L ,$$

where

$$\left(\sqrt{(L+\lambda)P^a(k|k)} \right)_i$$

is the i th column of the matrix square root of

$$(L+\lambda)P^a(k|k)$$

using the definition: The matrix square root A of B satisfies $B \equiv AA^T$.

3. The prediction sigma points are propagated through the equations of motion:

$$\chi_{p,i}(k+1|k) = f(\chi_{p,i}(k|k)) , \text{ for } i = 0 \dots 2L .$$

4. The propagated sigma points are recombined to produce the predicted state and covariance:

$$\hat{x}(k+1|k) = \sum_{i=0}^{2L} W_s(i) \chi_{p,i}(k+1|k) ,$$

$$P(k+1|k) = \sum_{i=0}^{2L} W_c(i) [\chi_{p,i}(k+1|k) - \hat{x}(k+1|k)] [\chi_{p,i}(k+1|k) - \hat{x}(k+1|k)]^T ,$$

where the weights for the state and covariance are given by:

$$W_s(0) = \frac{\lambda}{L+\lambda} ,$$

$$W_c(0) = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) ,$$

$$W_s(i) = W_c(i) = \frac{1}{2(L+\lambda)} ,$$

$$\lambda = \alpha^2(L+\kappa) - L .$$

Values for α , β and κ have to be chosen to tune the prediction step. The constant α determines the spread of the sigma points around $x^a(k|k)$ and is usually set to small positive values less than one (typically in the range 0.001 to 1). The secondary scaling parameter κ is usually set to either 0 or $3 - L$. When κ is set to 0, weights of the sigma points are directly related to L . When $\kappa = 3 - L$, the 4th order moment information is mostly captured in the true Gaussian case. β is used to incorporate prior knowledge of the distribution of x in the computation of $W_c(0)$. In the case of the Gaussian distribution, the optimum value is $\beta = 2$. Some guidelines to choose these constants for a particular problem are given in Ref. 28.

5. For the update step the predicted state and covariance are augmented with the mean and covariance of the measurement noise:

$$x^a(k+1|k) = \left[\hat{x}^T(k+1|k) \quad E[v^T(k+1)] \right]^T ,$$

$$P^a(k+1|k) = \begin{bmatrix} P(k+1|k) & 0 \\ 0 & R \end{bmatrix} .$$

6. A set of $2L+1$ update sigma points is derived from $x^a(k+1|k)$ and $P^a(k+1|k)$ where L is the dimension of the augmented state:

$$\chi_{u,0}(k+1|k) = x^a(k+1|k) ,$$

$$\chi_{u,i}(k+1|k) = x^a(k+1|k) + \left(\sqrt{(L+\lambda)P^a(k+1|k)} \right)_i , \text{ for } i = 1 \dots L ,$$

$$\chi_{u,i}(k+1|k) = x^a(k+1|k) - \left(\sqrt{(L+\lambda)P^a(k+1|k)} \right)_{i-L} , \text{ for } i = (L+1) \dots 2L .$$

7. Alternatively, the prediction sigma points propagated through the equations of motion (step 3) can be used directly:

$$\chi_{u,i}(k+1|k) = \left[\chi_{p,i}^T(k+1|k) \quad E[v^T(k+1)] \right]^T \pm \sqrt{(L+\lambda)R^a} ,$$

where

$$R^a = \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix} .$$

8. The update sigma points are fed to the observation function h :

$$\gamma_i(k+1|k) = h(\chi_{u,i}(k+1|k)) , \text{ for } i = 0 \dots 2L .$$

9. The result is recombined to yield the predicted measurement and predicted measurement covariance:

$$\hat{z}(k+1|k) = \sum_{i=0}^{2L} W_s(i) \gamma_i(k+1|k) ,$$

$$P_{zz} = \sum_{i=0}^{2L} W_c(i) [\gamma_i(k+1|k) - \hat{z}(k+1|k)] [\gamma_i(k+1|k) - \hat{z}(k+1|k)]^T .$$

10. the UKF Kalman gain is computed as:

$$K_{k+1} = P_{xz} P_{zz}^{-1} ,$$

where the state-measurement cross-covariance matrix is expressed as:

$$P_{xz} = \sum_{i=0}^{2L} W_c(i) [\chi_{u,i}(k+1|k) - \hat{x}(k+1|k)] [\gamma_i(k+1|k) - \hat{z}(k+1|k)]^T .$$

11. The familiar state update equation is:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K_{k+1} (z(k+1) - \hat{z}(k+1|k)) .$$

12. Finally, the updated covariance is:

$$P(k+1|k+1) = P(k+1|k) - K_{k+1} P_{zz} K_{k+1}^T .$$