NTNU – Trondheim
Norwegian University of
Science and Technology

A Vision-Aided Navigation System based on Optical Flow

Jesper Hosen

Master of Science in Cybernetics and Robotics
Submission date: June 2015
Supervisor: Thor Inge Fossen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Formulation

Thesis Description:

The purpose of this project is to implement a camera-based navigation system based on optical flow and compare the performance with estimates from a conventional integrated navigation system.

The following items must be considered:

1. Define the scope of the thesis and clarify what your contributions are.
2. Implement and test algorithms for optical flow under the assumption of no moving objects.
3. Use optical flow measurements to estimate the motion between the UAV and the ground.
4. Design a vision-aided nonlinear observer, using IMU, camera measurements (optical flow) and GNSS for estimation of attitude and translational motion.
5. Design a multiplicative extended Kalman filter (MEKF) for estimation of attitude (quaternions) using IMU and camera measurements (optical flow), assuming known linear velocity in NED reference frame. Extend the Kalman filter for estimation of linear position and velocity using GNSS aiding.
6. Evaluate the performance of the nonlinear observer with the Kalman filter solution by using experimental data from UAV flights.
7. Compare the results of the two methods and conclude your results.

Start date: 2015-01-12
Due date: 2015-06-08
Thesis performed at: Department of Engineering Cybernetics, NTNU
Supervisor: Professor Thor I. Fossen, Dept. of Eng. Cybernetics, NTNU
Co-Supervisor: Lorenzo Fusini, Dept. of Eng. Cybernetics, NTNU

Abstract

This thesis presents design and comparison of a vision-aided uniformly semi-globally exponentially stable (USGES) nonlinear observer (NO) and a Multiplicative Extended Kalman Filter (MEKF) for estimation of attitude, gyro bias, position and velocity of a fixed-wing Unmanned Aerial Vehicle (UAV). The NO uses measurements from an Inertial Measurement Unit (IMU), a Global Navigation Satellite System (GNSS) receiver, and a video camera. The MEKF and the NO are evaluated with real world experimental data. Moreover two new NO representations are proposed. The proposed NOs have a computer vision (CV) system that is based on epipolar geometry, and hence independent of the depth in the images and the structure of the terrain being filmed. The first proposed NO utilizes a camera and the continuous epipolar constraint and is named Continuous Epipolar Optical Flow Nonlinear Observer (CEOF NO). The second proposed NO uses a camera and the epipolar constraint and is named Epipolar Optical Flow Nonlinear Observer (EOF NO). Experimental data from a UAV test flight show that the vision-aided NO is substantially less computational demanding than the MEKF. It is seen that the NO has similar performance as the MEKF by means of accuracy of the estimates. The NO and the MEKF estimates are compared with an Extended Kalman Filter (EKF) implemented on the onboard autopilot of the UAV. The results illustrate that the estimates of the states converge accurately to the correct values. Moreover simulated data show that the proposed observers have more robust CV than the previous developed vision-aided NO, yielding more accurate and robust performance.

Keywords: Unmanned Aerial Vehicle, Optical Flow, Vision-based Navigation, Mulitplicative Extended Kalman Filter, Nonlinear Observer, Epipolar Geometry, Continuous Epipolar Constraint, Navigation System, Image Processing, Computer Vision, Sensor Fusion, State Estimator

Sammendrag

Denne avhandlingen presenterer design og sammenligning av en ulineær observer og et multiplikativt utvidet Kalman filter for estimering av orientering, gyroskopbias, posisjon og hastighet av et ubemannet fly (UAV). Den ulineære observeren og Kalman filteret benytter seg av målinger fra IMU, et globalt posisjonerings system (GPS) og maskinsyn. Kalman filteret og den ulineære observeren sammenlignes ved bruk av ekte data, samlet i en testflyvning av en UAV. Den ulineære observeren viser seg å ha sammenlignbar ytelse som det konvensjonelle multiplikativt utvidede Kalman filteret. Prinsippene bak maskinsynet i den eksisterende kamera-assisterte ulineære observeren krever at terrenget som filmes av kameraet er flatt og horisontalt. Derfor behøves et nytt prinsipp for maskinsyn i observeren, og det foreslås to nye ulineære observer-representasjoner. Disse representasjonene baserer seg på andre prinsipper for maskinsyn enn den eksisterende kamera-assisterte ulineære observeren. De foreslåtte endringene i observeren gjør at systemet kan benyttes uavhengig av terrenget, og dermed kan brukes ved flyvning over terreng som ikke er flat og horisontalt. Eksperimentelle data, fra en flyvning over et relativt flatt område, viser at de foreslåtte ulineære observerne gir lik ytelse som den eksisterende kamera-assisterte ulineære observeren. Videre simuleres en UAV flyving over ulendt terreng. Resultatene fra simuleringen viser at den eksisterende ulineære observeren gir feilaktige estimater av orientering og gyroskopbias ved flyvning over terreng som ikke er horisontalt flatt. De foreslåtte observerne gir estimater som konvergerer til korrekte verdier uavhengig av strukturen på terrenget.

Preface

The work described in this report is carried out in the Department of Engineering Cybernetics, at the Norwegian University of Science and Technology, the spring of 2015. To work with this subject has been educational and has ended in this thesis report. In addition to this I have been co-author of a published paper at the International Conference on Unmanned Aircraft Systems 2015 (ICUAS'15). The paper is written together with PhD. candidate Lorenzo Fusini, MSc. student Håkon H. Helgesen, Professor Thor I. Fossen and Professor Tor A. Johansen. Furthermore I have written a second paper based on the main findings in this thesis together with the aforementioned authors. The paper has been submitted to The 2016 American Institute of Aeronautics and Astronautics Science and Technology Forum and Exposition (AIAA SciTech 2016), and is still waiting for a decision of acceptance or rejection.

I would like to thank my supervisor Professor Thor I. Fossen for his guidance and inspirational meetings, and Lorenzo Fusini for his supervision. Professor Tor A. Johansen deserves a big thanks for proofreading and giving useful feedback on the papers I've worked with. Furthermore, I am grateful to Kristoffer Gryte for useful suggestions and Erik Wilthil for 24/7 white noise and LaTeX support. A huge thanks goes to my fellow MSc. student Håkon H. Helgesen for inspirational collaboration and valuable conversations throughout this year. At last I would like to thank my parents, brothers, girlfriend and friends for supporting me the last five years.

Jesper Hosen
Trondheim, June 2015

Table of Contents

III Methods 73

IV Results and Discussion 91

List of Tables

List of Figures

Abbreviations

6 DOF	= Six Degrees of Freedom
CEOF	= Continuous Epipolar Optical Flow
CSV	= Comma Separated Files
CV	= Computer Vision
DCM	= Direction Cosine Matrix
DOF	= Degrees of Freedom
ECEF	= Earth Centered Earth Fixed
EKF	= Extended Kalman Filter
EOF	= Epipolar Optical Flow
FOV	= Field of View
GNSS	= Global Navigation Satellite System
GPS	= Global Positioning System
GTOF	= Ground Truth Optical Flow
IMU	= Inertial Measurement Unit
INS	= Inertial Navigation System
MEKF	= Multiplicative Extended Kalman Filter
NO	= Nonlinear Observer
OF	= Optical Flow
SIFT	= Scale Invariant Feature Transform
TM	= Template Matching
UAV	= Unmanned Aerial Vehicle
USGES	= Uniformly Semi-Globally Exponentially Stable

Part I

Preliminaries

Chapter 1

Introduction

This thesis seeks to compare a vision-aided nonlinear observer (NO) with a traditional multiplicative extended Kalman filter (MEKF), both utilizing computer vision (CV) to generate body-fixed linear velocity measurements. The thesis compares the performance of the NO and the MEKF based on data from a real world experiment. Moreover this thesis presents a new CV subsystem to the NO, yielding a new observer representation relaxing some of the assumptions used in the previous presented vision-aided NO from Fusini et al. (2014, 2015).

The main objective of this thesis is to:

- Implement and compare the performance of the vision-aided nonlinear observer by Fusini et al. (2014) with a multiplicative extended Kalman filter.

- Propose a new, more robust, vision-aided nonlinear observer, with computer vision that is independent of the structure being filmed.

This chapter provides a short introduction to the research field of vision-aided navigation. Furthermore the contributions of this thesis are highlighted. An outline of the thesis ends the chapter.

1.1 Historical View and Motivation

The use of Unmanned Aerial Vehicles (UAV) has in the last decade gained increasingly attention, and already plays a major role in military use. The field of applications for UAVs will grow even more in the future, and the demands for reliability are considered to be crucial. Robust navigation is one of the most important parts when working with UAVs. A challenge in navigation systems is to maintain accurate estimates of the states when using low-cost measurement units. The output of such low-cost sensors are typically contaminated by noise and bias, hence some filtering is necessary. As it is desirable to have low energy consumption on UAVs, it is desired to find computational light weight navigation systems with good performance. The Kalman filter has been the preferred filter algorithm in many years, but in the later years nonlinear passive observers, like the nonlinear complementary filter has gained increasingly interest (Mahony et al., 2008; Crassidis et al., 2007; Hua, 2010; Grip et al., 2012*a*; Fusini et al., 2014, 2015).

Vision-based navigation is a globally prioritized research field. The use of cameras for navigational purposes is expected to grow fast since video cameras are lightweight, energy efficient and the prices are constantly decreasing. Cameras with great quality can be acquired at low cost. As magnetometers are very sensitive to disturbances, such as electromagnetic fields (Euston et al., 2008), cameras might be a good alternative or complementary to the magnetometer in a navigation system. The camera images can be used to calculate the body-fixed velocity of a UAV, which can provide valuable information to the navigation system.

1.2 Previous Work

The problem of estimating attitude has been the focus of many researchers (Batista et al., 2011*b*,*a*; Grip et al., 2012*b*; Guerrero-Castellanos et al., 2013; Hua et al., 2014; Mahony et al., 2008, 2009, 2011; Salcudean, 1991; Thienel and Sanner, 2003). Integration of inertial navigation systems, magnetometers and GNSS to estimate navigation states of vehicles have been done by other researchers (Maybeck, 1979; Phillips and Schmidt, 1996; Grewal et al., 2001; Vik and Fossen, 2001; Groves, 2013).

Nonlinear Observer

Mahony et al. (2008) and Mahony et al. (2009) proposed a vector-based NO for estimation of attitude based on low cost inertial measurement sensors, in particular an accelerometer, gyroscope and magnetometer. Hua (2010) expanded this observer to include GNSS velocity measurements. Batista et al. (2011*b*) and Batista et al. (2011*a*) built globally exponentially stable (GES) attitude estimators based on multiple time-varying reference vectors or a single persistently exiting vector. A similar observer, including also gyro bias estimation and GNSS integration, was developed in Grip et al. (2012*a*) and Grip et al. (2015). Grip et al. (2013) replaced the rotation matrix with the unit quaternion for representing attitude, considered Earth rotation and curvature and a non-constant gravity vector. Fusini et al. (2014)

presented a vision-aided NO utilizing body-fixed velocity measurements from CV instead of magnetometer measurements. Fusini et al. (2014) proved that the vision-aided NO is Uniformly Semi-Globally Exponentially Stable (USGES), as defined in Grotli et al. (2008), in the origin of the error dynamics. An experimental validation of the vision-aided NO, showing that the system works with real world experimental data, was presented in Fusini et al. (2015). The CV in the NO by Fusini et al. (2014, 2015) used optical flow (OF) and assumed ground truth optical flow (GTOF). With GTOF it is assumed known distance to all features corresponding to a OF vector, making it possible to recover the ego-motion of a camera from the relationship between theoretical OF and camera motion. Requiring known distance to any feature in the image is a fundamental restriction, imposing different restrictive assumptions for the CV.

Computer Vision in Navigation

In this thesis CV will be used to aid inertial navigation. The camera is commonly used in navigation. Low weight, low power consumption, and a wide range of CV software make it a viable choice for navigation purposes. Some drawbacks are its dependence on lighting and weather conditions, which directly affect the ability to extract significant features from the images to be used in image processing, and the difficulty in separating camera motion from moving objects in complex non-stationary environments. The CV of the observers in this thesis uses OF as a measurement. OF is how features in the image plane moves in time, caused by relative motion between the camera and the object being depicted. Several methods exists for determining the OF of a series of images (Lucas and Kanade, 1981; B.K.P.Horn and B.G.Schunk, 1981; Lowe, 1999; Bay, Tuytelaars and Van Gool, 2008). Different OF algorithms are evaluated in Mammarella et al. (2012*a*); Chao et al. (2014) by the means of estimating UAV velocity. CV has been used for different applications in UAV navigation including indoor manoeuvring (Zingg et al., 2010; Shen et al., 2011), linear and angular velocity estimation (Dusha, 2007; Mammarella et al., 2012*a*; Weiss et al., 2013; Kehoe et al., 2006; Moore et al., 2012) and obstacle avoidance (Zufferey and Floreano, 2005; Hrabar et al., 2005; Zingg et al., 2010; Merrell et al., 2004*a*; Conroy et al., 2009; Ruffier and Franceschini, 2004) as well as height above the ground estimation in (Merrell et al., 2004*b*). Bibuli et al. (2007); Ahrens et al. (2009) have used camera as sensor for navigating in GPS-denied environments.

An effective principle for recovering the ego-motion of a camera is epipolar geometry and the epipolar constraint (Hartley and Zisserman, 2003). Epipolar geometry relates motion of a camera with the projective motion in the images, independent of the depth in the images. The principle of epipolar geometry have been used together with Kalman filtering in navigation (Diel et al., 2005; Weiss, 2012; Weiss et al., 2013; Mercado et al., 2013; Bazin et al., 2010; Meingast et al., 2004; Sanfourche et al., 2012; Davison et al., 2007; Brockers et al., 2012; Herisse et al., 2008). Ma et al.

(2001) presented the epipolar constraint in the continuous case. Grabe, Bulthoff and Robuffo Giordano (2012); Grabe, Bulthoff and Giordano (2012) have used the continuous epipolar constraint together with an extended Kalman filter to recover the velocity of a UAV.

1.3 Contributions

This thesis seeks to compare two different navigation filters: the nonlinear observer (NO) with the Multiplicative Extended Kalman Filter (MEKF). Moreover two new NO representations are proposed. The main contributions of this thesis are:

M-matrix expressed with attitude quaternion (Section 3.3.1) The M-matrix from Fusini et al. (2014, 2015) describes the relationship between theoretical OF and 6DOF motion. By expressing the M-matrix by quaternions, singularity issues are avoided.

CEOF nonlinear observer (Section 5.5) A new proposed NO representation is presented. This system enables CV based on the continuous epipolar constraint (Ma et al., 2001; Hartley and Zisserman, 2003) and is named Continuous Epipolar Optical Flow Nonlinear Observer (CEOF NO). The CEOF NO has, in contradiction to the GTOF observer by Fusini et al. (2014, 2015), CV that is independent of the structure of the terrain. The computer vision of the CEOF NO is independent of the depth in the images, yielding more relaxed assumptions, and in turn a more robust observer representation. The GTOF observer by Fusini et al. (2014, 2015) has some strict requirements of having measurements of the attitude and altitude of the UAV. In addition it requires the terrain being filmed to be flat and horizontal. These requirements are removed with the CEOF NO. A stability proof is presented showing that the new proposed CEOF NO representation has the same stability properties as the GTOF observer by Fusini et al. (2014, 2015). That is a USGES equilibrium point at the origin of the error dynamics.

EOF nonlinear observer (Section 5.6) A second proposed NO representation is presented. This system uses CV based on the discrete epipolar constraint (Hartley and Zisserman, 2003) and is named an Epipolar Optical Flow Nonlinear Observer (EOF NO). The motivation behind the EOF NO is that the OF-algorithms are discrete in their nature and measure only the displacement of features from image to image. That is the discrete OF. As the discrete epipolar constraint is derived by assuming discrete OF and that computer implementation implies discretization, the EOF NO is argued to have a more appropriate CV. A stability proof is presented showing that the new proposed EOF NO representation has, under some assumptions, the same stability properties as the proposed CEOF NO and the GTOF observer by Fusini et al. (2014, 2015). The EOF NO is like the CEOF NO independent of the structure being filmed.

UAV flight test (Chapter 6) A UAV flight test is conducted in order to collect real sensor data that can be used to evaluate the different observers.

UAV with optical flow sensor simulator (Chapter 7) A simulator is implemented. A UAV with optical flow sensor is simulated flying over an coastline environment. The simulator produces IMU and GNSS data as well as simulated OF vectors, which in turn will be used to evaluate the performance of the different NO representations.

Experimental testing of observers (Chapter 8) Different nonlinear observers and MEKF are implemented and tested. An experimental comparison of performance of a MEKF, the GTOF NO proposed by Fusini et al. (2014), the proposed CEOF observer and the proposed EOF observer, using real world data collected from a UAV flight test, is presented. Experimental validation of the observer proposed by Fusini et al. (2014) including feedback of attitude estimates to CV is also presented.

Comparison of GTOF, CEOF and EOF nonlinear observers (Chapter 9) A comparison of performance of the GTOF NO (Fusini et al., 2014, 2015), the proposed CEOF and EOF NOs using simulated data. A UAV flight is simulated over rugged terrain. The simulation is presented to highlight the advantages of the proposed observers compared to the GTOF observer.

Paper submitted to AIAA SciTech 2016 (Appendix A) I have been first author of a paper presenting the proposed CEOF NO, and comparing the CEOF observer with the GTOF observer from Fusini et al. (2014, 2015). The paper was submitted for The AIAA Science and Technology Forum and Exposition conference. Any decision on acceptance is not yet taken, but is expected before 25th of August 2015.

Paper accepted to ICUAS15 (Appendix B) I have been co-author on a paper giving an experimental validation of the GTOF nonlinear observer by Fusini et al. (2014), which was submitted and accepted for The 2015 International Conference on Unmanned Aircraft Systems (ICUAS'15). The paper was presented in Denver June 2015.

1.4 Outline

This thesis is organized in six parts, with twelve chapters and eight appendices:

Part I: Preliminaries

- Chapter 1: Introduction. An introduction to the research field is presented, main contributions are listed, and an outline of the thesis is displayed.

- Chapter 2: Nomenclature. Definitions of variables used in this thesis.

Part II: Theory

- Chapter 3: Computer Vision. Gives an introduction to CV, and how it has been used in navigation. Description of OF and OF algorithms. Different principles for recovering ego-motion from OF is presented.
- Chapter 4: Multiplicative Extended Kalman Filter for UAV Navigation. An observer utilizing IMU, GNSS and CV is designed and presented.
- Chapter 5: Nonlinear Observer for UAV Navigation. A modified version of the vision-aided NO by Fusini et al. (2014) is presented. Moreover two new NO representations are proposed.

Part III: Method

- Chapter 6: Experiment. A UAV test flight is conducted and described in this chapter. Case studies for evaluating the different observers with real sensor data are presented.
- Chapter 7: Simulation. A simulator is implemented and presented. Furthermore a case study of evaluating different observers with the simulated data is described.

Part IV: Results and Discussion

- Chapter 8: Experiment Results. The different NO representations and MEKF are tested and evaluated with real world data from the flight.
- Chapter 9: Simulation Results. The proposed NO representations are compared to the GTOF NO by Fusini et al. (2014) when simulating a flight over rugged terrain.
- Chapter 10: Discussion. A discussion of the main findings in the results.

Part V: Closing Remarks

- Chapter 11: Conclusion. Presents a conclusion of the thesis and the main findings in this thesis.
- Chapter 12: Future Work. Summarizes what should be done in future work, and forms a description of ideas for future work and challenges that need to be solved.

Part VI: Appendices

- Appendix A: Submitted paper to AIAA SciTech 2016. A paper is written based on the results and findings in this thesis.
- Appendix B: Accepted paper to the ICUAS'15. A paper on experimental validation of the GTOF NO from Fusini et al. (2014).
- Appendix C: Derivation of the M-matrix. M is a matrix relating optical flow to ego-motion.
- Appendix D: Software Implementation. OF algorithms, MEKF and NO are implemented. General implementation considerations are assessed.
- Appendix E: Payload Setup. Description of the UAV payload module used to record data.
- Appendix F: Reference Frames. Explanation of the reference frames used in this thesis.
- Appendix G: Attitude Representations. Explanation of the attitude representations used in this thesis.
- Appendix H: Sensor Models. Measurement equations for the sensors used in the MEKF and NO design.

Chapter 2

Nomenclature

This chapter lists the variables and reference frames used in this thesis.

The reference systems used in this thesis are illustrated in Figure 2.1. {b} is the body-fixed coordinate system, {c} is the camera frame, {m} is the image plane and {n} is the earth fixed North-East-Down reference system. {n} is assumed inertial. A more detailed description of the reference frames is attached in Appendix F.

Euler Angels, Direction Cosine Matrices and unit quaternions are used to represent attitude in this thesis. Explanation of different attitude representations are described in Appendix G.

In this thesis vectors are bold. Matrices are represented by upper-case letters. Scalars are lower-case letters. The superscript is used to denote which frame a vector is expressed in. Some vectors have subscript showing which frames the vectors are relative to. For rotation matrices sub- and superscripts are used to show which frames a vector is rotated from and to, respectively. Some examples are shown to clarify:

$\boldsymbol{v}^n_{b/n}$ = The linear velocity of {b} relative to {n} expressed in {n}

R^n_b = The rotational matrix relating {b} to {n}

λ = A scalar value

Moreover the subscripts are also used to indicate what sensor a measurement come from. Throughout this thesis it is assumed that IMU and GPS measures a state of the UAV body {b} w.r.t {n} , meaning $\boldsymbol{p}^n_{\text{GPS}}$ indicates that it is a measurement from the GPS of the {b} w.r.t {n} given in {n} -frame. Equivalently $\boldsymbol{\omega}^b_{\text{imu}}$ is a

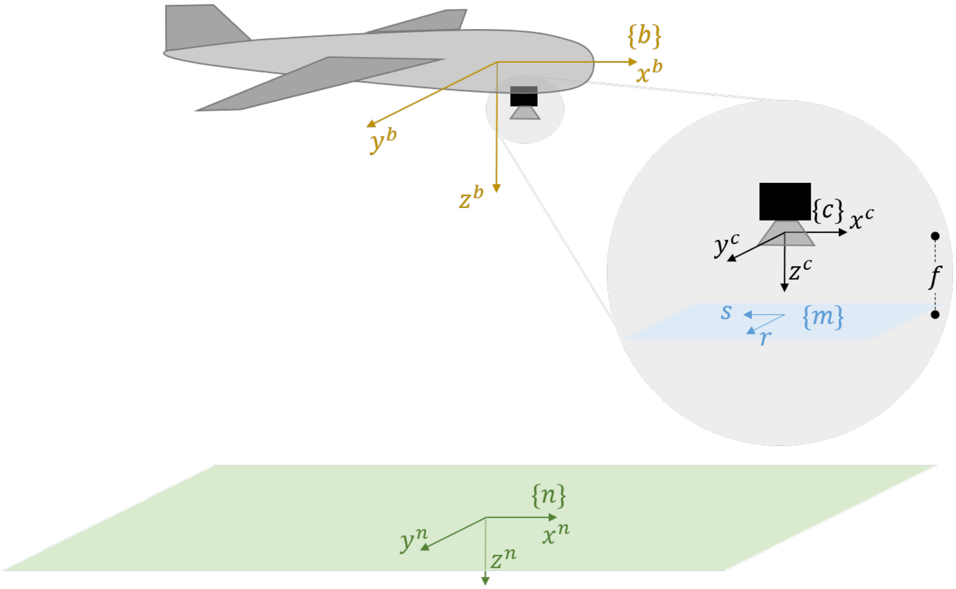


Figure 2.1: Different reference frames are used in this thesis. The figure shows the relationship between body frame {b} , Camera frame {c} , image plane {m} and NED frame {n} . f is the focal length of the camera.

measurement from the IMU of the angular velocity of the UAV body {b} w.r.t {n} given in {b} -frame.

General

$I_{n\times n} \in \mathcal{R}^{n\times n}$: The n-by-n identity matrix.
$0_{n\times m} \in \mathcal{R}^{n\times m}$: The n-by-m zero matrix.
diag($\boldsymbol{x}$) $\in \mathrm{R}^{n\times n}$: The diagonal matrix with the elements of $\boldsymbol{x}$ on the diagonal. $\boldsymbol{x} \in \mathrm{R}^n$.
$[\boldsymbol{x}]_\times \in \mathcal{R}^{3\times 3}$: The skew symmetric matrix representing the crossproduct in $\mathcal{R}^3$. $[\boldsymbol{x}]_\times \boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$, if $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{R}^3$.
A^+: The pseudoinverse of a matrix A.
$\boldsymbol{i} = [1, 0, 0]^T$: Unit vector in x-direction (in $\mathcal{R}^3$).
$\boldsymbol{j} = [0, 1, 0]^T$: Unit vector in y-direction (in $\mathcal{R}^3$).
$\boldsymbol{k} = [0, 0, 1]^T$: Unit vector in z-direction (in $\mathcal{R}^3$).
t: Time.
Δt: Time between two consecutive images.
δt: Time between two iterations in the observer.
$\boldsymbol{g}^n \in \mathcal{R}^3$: Gravitational vector in {n} .

$\hat{x}$: Estimate of x.
$\tilde{x} = x - \hat{x}$: Error in estimate of x.
$\underline{\boldsymbol{x}} = [\boldsymbol{x}^T, 1]^T$: The corresponding homogeneous coordinate of $\boldsymbol{x}$.
$\boldsymbol{f}^n_{b/n}$: Specific force acting on {b} w.r.t. {n} expressed in {n} .
SO(3): The special ortogonal group.

Dynamics

$R^n_b \in \mathrm{SO}(3)$: Rotation matrix relating {b} and {n} .
T^n_b: Homogeneous transformation matrix transforming points from {b} to {n} .
$\boldsymbol{v}^b_{b/n}$: Linear velocity of {b} w.r.t. {n} expressed in {b} .
$\boldsymbol{v}^n_{b/n}$: Linear velocity of {b} w.r.t. {n} expressed in {n} .
$\boldsymbol{\omega}^b_{b/n}$: Angular velocity of {b} w.r.t. {n} expressed in {b} .
$\boldsymbol{\Theta}_{b/n} = [\phi, \theta, \psi]^T$: Euler angles expressing the orientation of {b} w.r.t {n} .

Computer Vision

f: Focal length of the camera.
K: Projection matrix transforming coordinates from camera frame, {c} , to image coordinates {m} .
$\boldsymbol{x}^m = [r, s]^T$: Image coordinate.
$\underline{\boldsymbol{x}}^m = [r, s, 1]^T$: Homogeneous image coordinate.
$\dot{\boldsymbol{x}}^m = [\dot{r}, \dot{s}]^T$: Continuous optical flow.
$\frac{\Delta \boldsymbol{x}^m}{\Delta t} = [\frac{\Delta r}{\Delta t}, \frac{\Delta s}{\Delta t}]^T$: Discrete optical flow.
M: Matrix relating optical flow to ego-motion of the camera.
$\boldsymbol{u}^c = K^{-1}\underline{\boldsymbol{x}}^m$: Back projected point with unit z component.
$\boldsymbol{c}^n = [c^n_x, c^n_y, c^n_z]^T$: Camera position in {n} .
$R^n_c \in \mathrm{SO}(3)$: Rotation matrix relating {c} and {n} .
t_k: The time when image k is captured.
t_{k+1}: The time when image $k+1$ is captured.
$\{c_k\}$: Camera frame at time t_k.
$\{c_{k+1}\}$: Camera frame at time t_{k+1}.
$\delta R = R^{c_{k+1}}_{c_k} \in \mathrm{SO}(3)$: Rotation matrix relating camera pose at time t_k and t_{k+1}.
$\boldsymbol{t} = \boldsymbol{t}^{c_{k+1}}_{c_k} = [t_x, t_y, t_z]^T \in \mathcal{R}^3$: Translation of the camera between time t_k and t_{k+1}.
$\boldsymbol{v}_e \in \mathcal{R}^3$: The scaled body-fixed linear velocity with forward unity component.
$\boldsymbol{v}^b_{\mathrm{cv}} \in \mathcal{R}^3$: Normalized body-fixed linear velocity measured by computer vision.

Sensor Measurements

$\boldsymbol{\omega}^b_{\text{imu}}$: Measured angular velocity by the gyroscope.
$\boldsymbol{f}^b_{\text{imu}}$: Measured spesific force by the accelerometer.
$\boldsymbol{b}^b_{\text{gyro}}$: Gyro bias.
$\boldsymbol{b}^b_{\text{acc}}$: Accelerometer bias.
$\boldsymbol{p}^n_{\text{GPS}}$: Measured position by GPS.
$\boldsymbol{v}^n_{\text{GPS}}$: Measured velocity by GPS.

The Multiplicative Extended Kalman Filter

$q = [\eta, \boldsymbol{\epsilon}^T]^T$: Quaternion expressing the attitude of {b} w.r.t. {n} .
$\hat{q}$: Estimated quaternion expressing the attitude of {b} w.r.t. {n} .
$\delta q = \hat{q}^{-1} \otimes q$: Error quaternion.
$\delta\eta$: The real part of the error quaternion.
$\delta\boldsymbol{\epsilon}$: The imaginary part of the error quaternion.
$R^n_b(q) \in \text{SO}(3)$: Rotation matrix defined by the quaternion q.
P: Error covariance matrix.

The Nonlinear Observer

$\hat{R}^n_b \in \text{SO}(3)$: Estimated rotation matrix.
$\hat{\boldsymbol{b}}^b_{\text{gyro}} \in \mathcal{R}^3$: Estimated gyro bias.
$\boldsymbol{f}^n_{b/n} \in \mathcal{R}^3$: Specific force acting on {b} w.r.t {n} expressed in {n} .
$\hat{J} \in \mathcal{R}^{3\times3}$: Output injection term used in the observer.
$\text{vex}(\boldsymbol{x})$: The inverse operator of $[\boldsymbol{x}]_\times$ ($\boldsymbol{x} = \text{vex}([\boldsymbol{x}]_\times)$).
$\mathbb{P}_a(A) = \frac{1}{2}(A - A^T)$: The skew symmetric part of a matrix A.
$\text{Sat}(A)$: Element wise saturation of the matrix A in the interval $[-1, 1]$.

Part II

Theory

Chapter 3

Computer Vision

Computer vision (CV) has gained significant attention for navigation purposes in recent years. This is partly because of the cheap price of cameras, but also because of the extensive information that a camera can output. The availability of UAVs have increased, the prices for cameras are constantly decreasing and the available computational power is becoming less restrictive. This means that vision-based navigation systems are easier to realize now than ever.

The objective of this chapter is to describe how CV can be used to calculate the egomotion of a camera based on optical flow(OF). In Chapter 4 and Chapter 5 navigation systems are designed to take use of measurements from CV. A camera is placed on a UAV and is used to find the body-fixed linear velocity which is used as a measurement in the navigation system. It is common to use magnetometers when navigating UAVs, but in this thesis the magnetometer measurements will be replaced by measurements from CV. Magnetometer measurements are often prone to errors in small UAVs due to the magnetic field resulting from electronic propulsion systems and other disturbances (Euston et al., 2008). Using CV in UAV navigation might be a good alternative or complimentary to the magnetometer.

Three methods to determine the velocity of a camera will be described in this chapter. The first method described in this chapter utilizes OF vectors, roll, pitch and altitude of the camera in order to calculate linear and angular body-fixed velocity. The method is based on ground truth optical flow (GTOF), which means that the mathematical relationship between motion relative to a structure and theoretical OF is used to recover the egomotion of the camera. The relationship between OF and egomotion was described by Mammarella et al. (2012*b*); Fusini et al. (2014, 2015). These researchers presented this relationship by expressing the attitude by the Euler angles. It is convenient to express this relationship by Euler angles when using an inclinometer to measure the roll and pitch angle directly. However, it is advantageous to express this relationship by a singularity-free attitude representation. Therefore the relationship will be derived by using the attitude quaternion. The main drawback with GTOF is that assumes known distance to every point in the image. The distance is often recovered by assuming flat horizontal terrain and known altitude and attitude.

The two other methods are based on epipolar geometry, OF and relative attitude change between two consecutive images. The epipolar constraint is a well known relationship between motion and image projection (Hartley and Zisserman, 2003), and is often used in navigation (Shakernia et al., 2002; Grabe, Bulthoff and Robuffo Giordano, 2012; Grabe, Bulthoff and Giordano, 2012). By considering homogeneous transformations between images, it is possible to recover the motion of the camera. The first epipolar method is based on the epipolar constraint (Hartley and Zisserman, 2003), and is referred to as egomotion from epipolar optical flow (EOF). By EOF it is possible to recover a measure of the translation up to scale of the camera between two images. As epipolar geometry eliminates the dependence on the depth in the images, it is not possible to recover the scale of the translation vector. The second epipolar method is based on the continuous epipolar constraint (Ma et al., 2001), and is addressed as egomotion from continuous epipolar optical flow (CEOF). From the continuous epipolar constraint it is possible to recover the body-fixed linear velocity of the camera based on OF and the angular velocity. The epipolar methods EOF and CEOF hold some advantages over the GTOF method. The most important advantage is that EOF and CEOF are independent of the distance to the structure being filmed.

This chapter is divided into three main sections. The first section seeks to describe how a camera works, and how a image features relates to real world points through a projection called the pinhole camera model. The second part will give a definition of OF, as well as describe different algorithms that can be used to find the OF of a series of images. The last section describes how the GTOF, EOF and CEOF methods are used to determine the body-fixed linear velocity of a camera.

3.1 Camera

The camera is an imaging sensor, which provides a two dimensional projection of the world onto the image plane. The images hold enormous amounts of data, and a challenge is to extract as much information as possible from the images.

The main part of a camera is the image chip, also called the imaging sensor. The imaging sensor can be implemented in different ways, e.g. with semiconductor charged-coupled devices (CCD) or complementary metal-oxide-semiconductor (CMOS), and is made up by many tiny pixels, which are sensitive to light. When being exposed to light they gets a electronic charge, which can be read out and interpreted as the intensity of the light. By reading out from multiple pixels one are able to create an image of the world seen by the camera. It is common to have some color separation filter that separates red, green and blue light, and passes light with different color to different pixel sensors.

In this section the basics behind camera will be highlighted and the mathematical relationship between points in the world and points on the image plane will be presented.

3.1.1 The Pinhole Camera Model

A camera model is a mathematical model describing the relationship between a photographed object and the perceived image. There are several camera models, but one of the simplest with high validity is the pinhole camera model (Hutchinson et al., 1996). As the camera should give some information about the movement relative to the structure being filmed, it is necessary to map a point from the camera-fixed frame {c} to the two dimensional image plane {m} . A relationship between $\boldsymbol{p}^c = [x^c, y^c, z^c]$ and $\boldsymbol{x}^m = [r, s]$ must be presented.

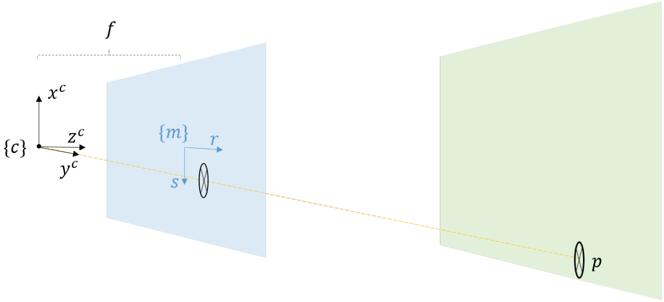


Figure 3.1: The pinhole camera model maps a perceived point in the camera-fixed frame to the image plane.

The pinhole camera model is illustrated in Figure 3.1. By geometric considerations (similarity of form) one may see that the relationship between r and y^c and the relationship between s and x^c can be written as

$$\frac{r}{f} = \frac{y^c}{z^c} \tag{3.1a}$$

$$\frac{s}{f} = -\frac{x^c}{z^c} \tag{3.1b}$$

(3.1a) and (3.1b) might then be combined and stated as

$$\begin{bmatrix} r \\ s \end{bmatrix} = \frac{f}{z^c} \begin{bmatrix} y^c \\ -x^c \end{bmatrix}, z^c \neq 0 \tag{3.2}$$

Moreover this can be written in matrix form as

$$\underline{\boldsymbol{x}}^m = \frac{1}{z^c} K \boldsymbol{p}^c \tag{3.3}$$

with $\underline{\boldsymbol{x}}^m = [r, s, 1]^T$ being the homogeneous image coordinate. K is the projection matrix, mapping points in the camera frame to the image plane, and is defined as

$$K = \begin{bmatrix} 0 & f & 0 \\ -f & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

where f is the focal length of the camera.

3.1.2 Camera Calibration

The pinhole camera model is a good approximation, but it assumes that the images are not affected by lens distortion and that the image projection is continuous. In general all cameras are prone to lens distortion to some extent. Lens distortion could e.g. be identified as straight lines appearing as bended lines. This is because a lens has defects that lead to blur, color changes or geometric distortion from the ideal ray (Sonka et al., 2014). It is possible estimate both the extrinsic (position and pose) and intrinsic (projection matrix, K) camera parameters, as well as the distortion coefficients by camera calibration. The distortion coefficients are used to rectify distorted images. A distorted image coordinate $[r_d, s_d]$ is related to the corresponding undistorted coordinate $[r_u, s_u]$ by the following formula (Sonka et al., 2014)

$$\begin{aligned} r_d &= r_u(1 + K_1\rho^2 + K_2\rho^4 + \ldots) + (P_2(\rho^2 + 2r_u^2) + 2P_1 r_u s_u)(1 + P_3\rho^2 + P_4\rho^4 \ldots) \\ s_d &= s_u(1 + K_1\rho^2 + K_2\rho^4 + \ldots) + (P_2(\rho^2 + 2s_u^2) + 2P_1 r_u s_u)(1 + P_3\rho^2 + P_4\rho^4 \ldots) \\ \rho &= \sqrt{(r_u - r_0)^2 + (s_u - s_0)^2} \end{aligned} \tag{3.5}$$

$[r_0, s_0]$ is the principal point of the image, which is estimated by camera calibration (for the ideal case it is located at the center of the image). P_n is the n^{th} tangential distortion coefficient and K_n is the n^{th} radial distortion coefficient, all estimated by camera calibration.

3.1.3 Field of View

The field of view (FOV) is the observable world seen by the camera at any given moment. The FOV is determined by the imaging chip and the focal length. Given a focal length f, and a imaging chip with dimensions $h \times v$ (horizontal and vertical dimension) the FOV can be calculated from the following, well known, formulas:

$$\alpha_h = 2\arctan(\frac{h}{2f}) \tag{3.6}$$

$$\alpha_v = 2\arctan(\frac{v}{2f}) \tag{3.7}$$

Definition 1. *α_h is the horizontal field of view.*
α_v is the vertical field of view.
h is the size of the imaging sensor in horizontal direction
v is the size of the imaging sensor in vertical direction

How the horizontal and vertical FOV relates to the UAV and the terrain are shown in Figure 3.2.

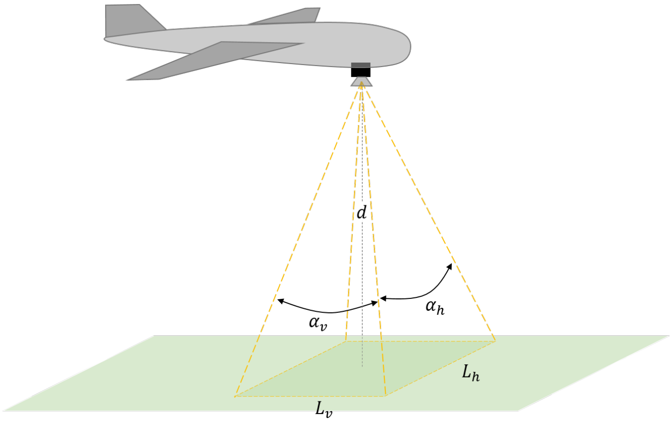


Figure 3.2: Field of view depends on the focal length of the lens and the sensor size. d is the distance to the structure being photographed. L_v and L_h is the vertical and horizontal lengths, depending on the vertical and horizontal field of view and the distance to the structure.

An important factor is the area on the ground the camera covers. The size of this area depends on the distance to the object being photographed. Since the field of view is constant, the width and length of the area covered will be proportional to the distance. The distance is denoted d. The horizontal length L_h and the vertical length L_v of the area displayed by the camera are:

$$L_h = 2d\tan(\frac{\alpha_h}{2}) \tag{3.8}$$

$$L_v = 2d\tan(\frac{\alpha_v}{2}) \tag{3.9}$$

It is necessary to have some overlap in two consecutive images in order to determine the motion of the camera. That means that the horizontal and vertical length, together with the frame rate of the camera, will determine an upper bound for the velocity of the camera.

3.1.4 Shutter

The shutter of a camera is the way the camera allows light to hit the imaging sensors. In digital cameras the shutter can be implemented by transferring multiple pixel cell charges at the same time, or one cell at the time. When the cell charges of the entire image is transferred at the same time, it is called a global shutter. In this thesis a camera with global shutter is used. That means that all pixels are read out the same time, and hence exposed to light the same amount of time. This reduces unwanted phenomenons like stretched images due to movement.

3.2 Optical Flow

OF is how features in an image plane moves between two consecutive images, caused by relative motion between the camera and the object being photographed. In this section OF will be explained, and some algorithms for determining OF will be presented.

OF has been studied for nearly 50 years and belongs to the research field of computer vision. Different algorithms can be ideal for different applications, depending on demands to accuracy and computational efficiency. It is important to find the most appropriate method for velocity calculation of a UAV. Mammarella et al. (2012*a*); Chao et al. (2014); Wen et al. (2011); Zingg et al. (2010) have evaluated different algorithms when estimating the velocity of a UAV based on images.

Some definitions are necessary:

Definition 2. *Continuous optical flow,* $\dot{\boldsymbol{x}}^m = [\dot{r}, \dot{s}]^T$*, is the time-derivative of the image plane position of a feature. Often referred to as optical flow.*

Definition 3. *Discrete optical flow,* $\frac{\Delta \boldsymbol{x}}{\Delta t} = [\frac{\Delta r}{\Delta t}, \frac{\Delta s}{\Delta t}]^T$ *is the displacement of a feature in the image plane from time* t *to time* $t + \Delta t$*.*

Definition 4. *A feature is a point of interest in the image plane.*

The difference between discrete and continuous OF is shown in Figure 3.3. In the simplest case OF could be understood as the pixel displacement of a single

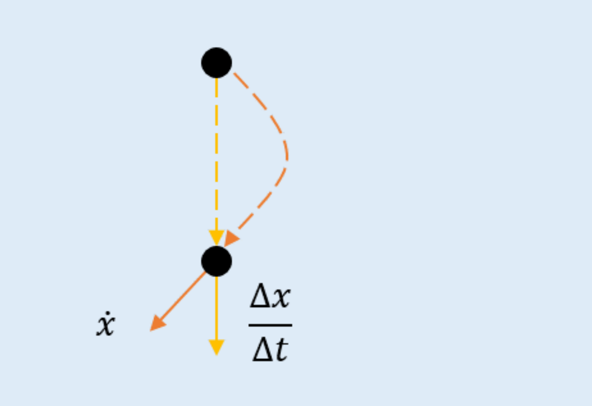


Figure 3.3: The black circle indicates a feature in the image plane. The feature is projected onto the image plane at time t and $t + \Delta t$. The yellow arrow indicates the discrete optical flow, while the orange indicates the continuous optical flow.

feature between two successive images. The OF can be represented as multiple vectors describing the change in the image plane in time. Several methods exists for determining the OF of a series of images (Lucas and Kanade, 1981), (B.K.P.Horn and B.G.Schunk, 1981), (Lowe, 1999), (Bay, Ess, Tuytelaars and Van Gool, 2008). In this thesis so called template matching and feature-based OF algorithms are implemented, and these will now be described.

3.2.1 Template Matching

Template matching (TM) (Sonka et al., 2014) calculates the OF by matching a region of two consecutive frames. A region around a pixel of interest, called a template, from one image is matched with a second image. The goal is to locate the same region in the second image. The OF is then calculated as the displacement of the template from the first to the second image. The TM technique can be based on correlation, difference techniques or other matching techniques. An outline of the TM OF algorithm is found in Table 3.1.

There are multiple ways to determine points of interest. One approach is to use a feature detector to find parts of the image with high contrast. This approach would often yield templates with high contrast, as a feature detector will detect distinct features. This can lead to tracking of areas that are easier to recognize. However there will always be a risk of whether the feature detector detects any features in the image. In worst-case the detector cannot find any features, and thus not a single OF vector is calculated. An other approach is to choose points of interest without any consideration of high/low contrast. This can be done by dividing the images into symmetrical regions, and choosing the points of interest as the centre

Table 3.1: Template Matching Optical Flow Algorithm

1. Find a point of interest.
2. Create a region of optional size around the point of interest in image one. This is the template.
3. Search for the same region in image two. This is done by matching the template with all possible positions in image two, maximizing the correlation of the images.
4. The matching point is the point in image two where the correlation is greatest. The optical flow is the displacement from the point of interest to the matching point.

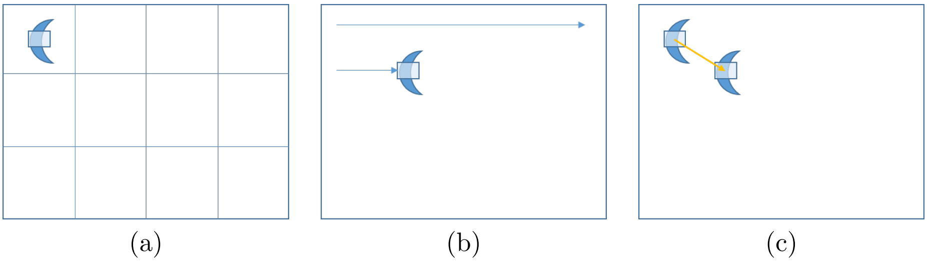


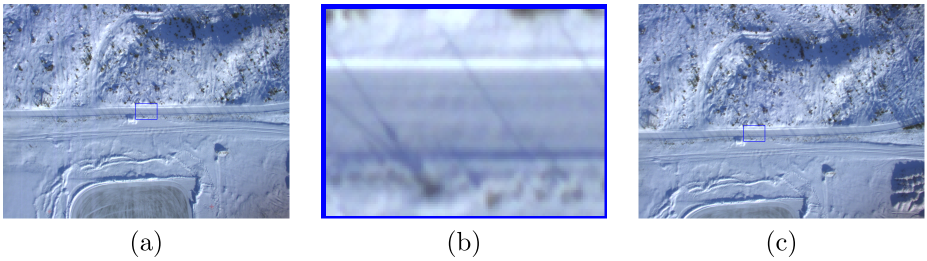
Figure 3.4: The template matching algorithm illustrated. (a) The first image is divided into a number of equally spaced regions, and a template is extracted from the centre of these regions. (b) The template matching searches for the template in the second image. The algorithm calculates the correlation with the template at all positions in the second image. (c) The match is found as where the correlation has its maximum value. The discrete optical flow is calculated as the displacement of the template from the first to the second image, and is illustrated by the orange arrow.

point of each of these regions. This is illustrated in Figure 3.4

Template matching is illustrated in Figure 3.5 by a successful match. It is seen that the camera has moved relative to the ground in the time between the two images were taken. An example of an OF field created with template matching, is displayed in Figure 3.6.

Advantages and disadvantages

An advantage with the TM method is that it performes well with the presence of fast motion. A disadvantage is that the matching techniques are computationally heavy. Another problem is that matching techniques based on template matching require the template to look equal in both images. Therefore the objects in the region of interest should have the same angle with respect to the camera, and the same number of pixels covering the region, as well as equal light conditions.



(a) (b) (c)

Figure 3.5: (a) Template is extracted from the marked region (blue rectangle) in the first image. (b) The template by itself. (c) The template is matched at the marked region (blue rectangle) in the second image.

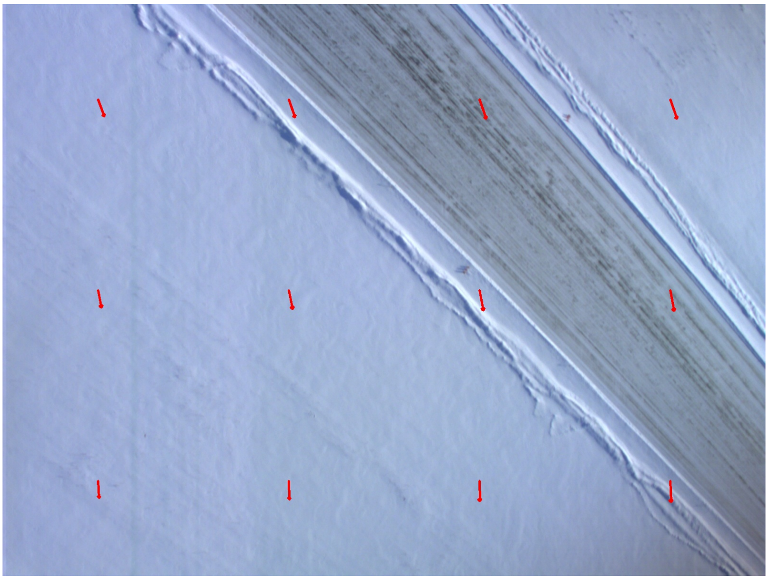


Figure 3.6: Optical flow calculated with template matching (a region-based method)

However the problem is reduced by making sure that the frequency between two images is sufficiently high.

Mismatching can happen when several areas are equal, and can be a problem for snowy or marine environments where the terrain looks very much alike. Mismatching is when a algorithm matches a template with the wrong region in the second image. If the video is recorded with a small frame rate, the projected terrain can change dramatically between two consecutive images which could lead to a mismatch due to big changes in the template area. Thus a large frame rate is an advantage for these methods.

Table 3.2: Feature-Matching Based Optical Flow Algorithm

1. Detect interest points, so called features, at several distinctive locations with a feature detector.
2. Give all features an unique descriptor that describes the neighbourhood of the feature.
3. Match descriptors for consecutive images to find common points in the images.
4. Calculate OF as the displacement of each feature.

3.2.2 Feature-Matching-Based Methods

Feature-Matching-Based optical flow algorithms calculates optical flow of points in the image, rather than regions in an image. Examples of feature-matching based methods are SIFT (Lowe, 1999) and SURF (Bay, Ess, Tuytelaars and Van Gool, 2008). A feature based method is summarized in Table 3.2.

The feature-matching based methods looks for distinct features in images, and assign these features a unique descriptor. A definition of descriptor follows:

Definition 5. *A descriptor is a vector describing a feature and its neighbourhood.*

Typical features extracted from the feature-detector are edges, lines and corners. The descriptors most important property are to uniquely describe each feature. The matching technique is in most cases based on least squares or a similar error minimization method. In this thesis the SIFT algorithm is implemented. Lets first consider the advantages and disadvantages with the feature based algorithms.

Advantages and disadvantages

The main advantage is that the features are handled independently, hence the relative position between the features in the image does not affect the matching. Some methods also allows the feature to change size and rotation with respect to the camera. These methods are called scale and rotation invariant. These methods have the advantage, as the region-based methods, to handle fast motion well. A disadvantage these methods is that the performance depends heavily on the matching technique chosen.

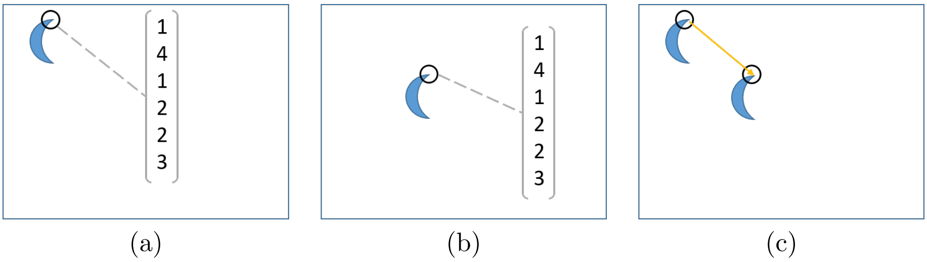


Figure 3.7: The SIFT optical flow algorithm illustrated. (a) The SIFT algorithm searches for features in the first image. The black circle indicates that a feature is found. The feature is given its own descriptor as indicated by the vector of numbers. (b) The SIFT algorithm then does the same for the second image. (c) The Flann based matcher is used to match the descriptors. The discrete optical flow is the displacement of a feature from the first to the second image, and is illustrated by the orange arrow.

Scale Invariant Feature Transform (SIFT)

The SIFT algorithm (Lowe, 1999) allows features to change in rotation and size between consecutive images. It is invariant to translation and partially invariant to brightness changes. The SIFT algorithm searches for distinct feature, and give all of these features their unique descriptor. This is done for two consecutive images. The descriptors could then ne matched to find corresponding features in the images by e.g a Fast Approximate Nearest Neighbor Search Library (Flann) based matcher (Muja and Lowe, 2009). As the SIFT algorithm is a feature based method, there is no lower bound on the number of OF vectors found. Moreover there is no guarantee that the OF vectors will be distributed around in the whole image. The SIFT OF algorithm is illustrated in Figure 3.7. An example of OF vectors from SIFT is seen in Figure 3.8.

3.3 Egomotion from Optical Flow

In this section different methods for estimating egomotion from OF will be described. Egomotion is defined by Definition 6. The main objective of this thesis is to use CV to estimate the navigation states (attitude, position, velocity and gyro bias) of a UAV. As normalized body-fixed linear velocity is needed to estimate these states, one is only interested in recovering the unit vector giving the direction of the body-fixed linear velocity.

Definition 6. *Egomotion is defined as the motion of a camera within an environment.*

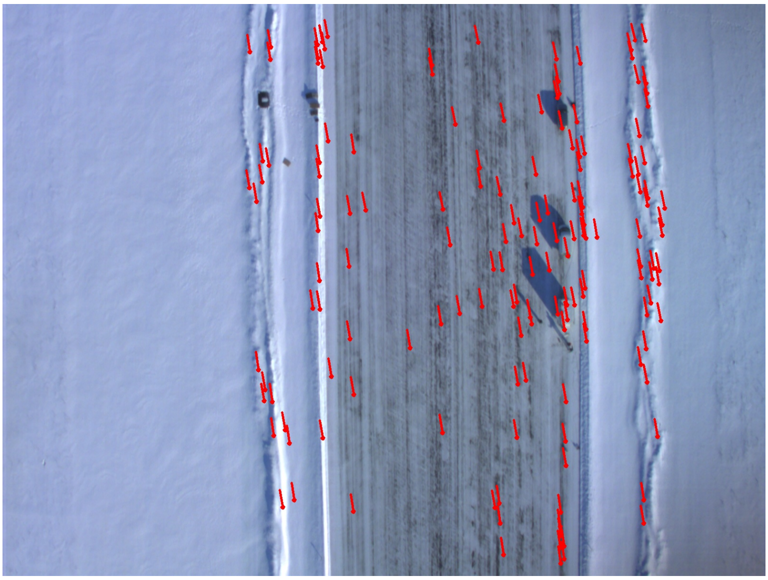


Figure 3.8: Example of optical flow vectors calculated with the SIFT algorithm.

The first method for recovering egomotion is based on ground truth optical flow (GTOF). The GTOF method uses the distance to every feature to recover the six degrees of freedom (6DOF) velocity of the camera with the scale preserved. The GTOF method recovers the distance to every feature by assuming that the attitude and altitude of the camera are known, and that the terrain being filmed is flat. In this thesis the GTOF relationship is derived using the attitude quaternion to avoid singularity issues. The author is not familiar with anyone else having expressed GTOF with the attitude quaternion.

The second method utilizes the well known epipolar constraint (Hartley and Zisserman, 2003) and incremental angles from a gyroscope. By employing the epipolar constraint one are able to generate the linear translation of the camera up to scale, namely a vector of unit length giving the direction of translation. This method is named an Epipolar Optical Flow (EOF) method.

At last a third method based on the continuous epipolar constraint (Ma et al., 2001) and angular velocity from gyroscope is presented. This method is referred to as an continuous epipolar optical flow (CEOF) method.

These three methods will now be described and justified.

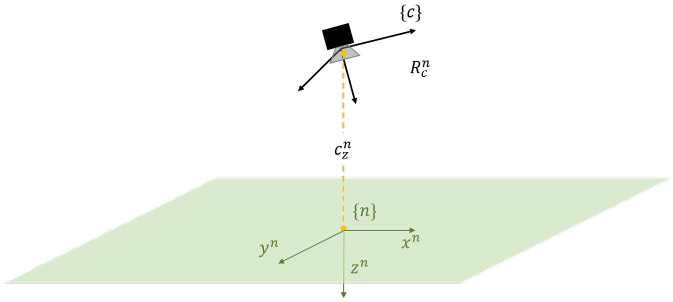


Figure 3.9: The ground truth optical flow (GTOF) method for determining egomotion needs to know the distance to any point in the image. When filming a flat terrain, the distance to a point can be recovered from the relative orientation w.r.t. the terrain, R_b^n, and the height above ground, c_z^n. Here the terrain and {n} is assumed to coincide.

3.3.1 Egomotion from Ground Truth Optical Flow

The problem of recovering egomotion from OF has been studied by Mammarella et al. (2012*b*), Wen et al. (2011) and Fusini et al. (2014) among others. In difference from that literature, this thesis seeks to derive the ground truth optical flow (GTOF) relationship in a singularity-free representation. When dealing with Euler angles, the formulas from the aforementioned researchers would yield a singularity when pointing the camera downwards (pitch at 90 degrees). In order to solve the singularity problem, the GTOF relationship is derived with assuming that the camera z-axis is aligned with body z-axis. Then a camera pointing down would have zero pitch and no singularity would occur. Moreover the attitude used in this derivation is represented by the quaternion. This is particularly interesting when using estimated attitude quaternion as input to the GTOF method, rather than Euler angles measured by an inclinometer.

The egomotion of a camera can be determined given OF vectors and the depth in the image, or equivalently the distance to the world points being captured. In this section the GTOF method for determining the egomotion of a moving camera will be derived. The GTOF-method is a method that is able to output 6DOF motion of a camera given the OF vectors between consecutive images and the depth in the images. When the camera is placed on a UAV, this can give a measure of the body-fixed velocity of the UAV. In the following section the camera motion is assumed to be the same as the UAV motion, meaning {c} and {b} is fixed to each other and aligned. The GTOF-method is derived with motivation in ground truth optical flow, which is assuming known distance to all points being considered w.r.t. the camera. This can be imposed by requiring that the terrain being filmed is flat, and that the attitude and altitude is known. This is illustrated in Figure 3.9. The assumptions and motivation for the GTOF-method will first be presented, before the mathematical relationship is presented.

Assumptions:

The following assumption must be made in order to calculate 6DOF velocity from GTOF:

> **Assumption 3.1.** *The distances from the camera to all features in the image are known.*

The distance in Assumption 3.1 can be recovered if the following three points are all satisfied:

- The attitude of the camera relative to the terrain is known.
- The altitude of the camera above ground is known.
- The terrain being filmed is flat.

Motivation

The motivation behind the GTOF method is that one are able to recover 6DOF velocity of the camera, with the scale of the velocity preserved. As the scale is preserved the measurement from the CV could for instance be used in dead reckoning. The angular velocity is also recovered, and can be used in navigation. However the measure of angular velocity is in general less accurate than the gyroscopic measurement, and should therefore not be used as a substitute for the gyroscope.

The M-matrix

As a relationship between points in the 2D image plane and the 3D camera-fixed frame is established by the pinhole camera model, one can differentiate (3.2) with respect to time in order to derive the motion relationship. A similar relationship was used in Mammarella et al. (2012*b*); Fusini et al. (2014) with camera pitched 45° down from the nose of the UAV. In this thesis the camera is pointing straight down, and this would yield a singularity in the relationship from Mammarella et al. (2012*b*); Fusini et al. (2014). The derivation of the new matrix expressing this relationship matrix is summarized in Appendix C. The result is presented:

$$\begin{bmatrix} \boldsymbol{v}^b_{b/n} \\ \omega^b_{b/n} \end{bmatrix} = - \underbrace{\begin{bmatrix} M_1(f, \boldsymbol{p}^c_1) \\ \vdots \\ M_N(f, \boldsymbol{p}^c_N) \end{bmatrix}^+}_{M^+} \begin{bmatrix} \dot{r_1} \\ \dot{s_1} \\ \vdots \\ \dot{r_N} \\ \dot{s_N} \end{bmatrix} \tag{3.10}$$

$$M_k(f, \boldsymbol{p}^c_k) = \frac{f}{z^c_k} \begin{bmatrix} 0 & 1 & -\frac{y^c_k}{z^c_k} & -\frac{y^c_k(y^c_k)}{z^c_k} - z^c_k & \frac{y^c_k x^c_k}{z^c_k} & x^c_k \\ -1 & 0 & \frac{x^c_k}{z^c_k} & \frac{x^c_k y^c_k}{z^c_k} & -\frac{x^c_k(x^c_k)}{z^c_k} - z^c_k & y^c_k \end{bmatrix} \tag{3.11}$$

Transformation from image plane to terrain coordinates

As can be seen in equation (3.11), the matrix $\boldsymbol{M}$ depends on $\boldsymbol{p}^c$. $\boldsymbol{p}^c$ is a position given in the camera-fixed frame. However it is convenient to express $\boldsymbol{M}$ as a function of the pose and height of the camera. In Fusini et al. (2014) this relationship was found as a function of Euler angles and height. In the following a relationship between $\boldsymbol{p}^c$ and $[r, s]$, depending on camera attitude quaternion q and height, will be derived.

The matrix $\boldsymbol{R}^n_c(q)$ and vector $\boldsymbol{c}^n$ represent a rotation and a translation between NED and the camera-fixed frame. They can be merged to form a 4 × 4-transform $\boldsymbol{T}^n_c$

$$\boldsymbol{T}^n_c = \begin{bmatrix} \boldsymbol{R}^n_c(q) & \boldsymbol{c}^n \\ 0_{1\times 3} & 1 \end{bmatrix}$$

Now let $\boldsymbol{t}^n = [x^n, y^n, z^n, 1]^T$ be a point expressed in NED and $\boldsymbol{t}^c = [x^c, y^c, z^c, 1]$ the same point expressed in the camera-fixed frame. The relationship between the vectors is given as

$$\boldsymbol{t}^c = (\boldsymbol{T}^n_c)^{-1} \boldsymbol{t}^n$$

Recall the pinhole model defined in (3.2). Now $\boldsymbol{t}^c = [x^c, y^c, z^c, 1]^T$ is a function of $x^n, y^n, z^n, \boldsymbol{c}^n$ and q. By inserting $\boldsymbol{t}^c$ into (3.2) one can solve the equation with respect to x^n and y^n. The solution is defined as x^n_T, y^n_T. The equation can be solved in Matlab using the symbolic toolbox. By constructing a new vector with the solution $\boldsymbol{t}^n_T = [x^n_T, y^n_T, z^n, 1]^T$, one can calculate the corresponding solution in the camera-fixed frame. $\boldsymbol{t}^c_T$ can be calculated by employing the 4×4-transformation again.

$$\boldsymbol{t}^c_T = (\boldsymbol{T}^n_c)^{-1} \boldsymbol{t}^n_T$$

Now by defining $\boldsymbol{p}^c$ as the three first rows of $\boldsymbol{t}^c_T$ one ends up with

$$\boldsymbol{p}^c = \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} = \frac{c_z^n - z^n}{f - 2\epsilon_1^2 f - 2\epsilon_2^2 f + 2\epsilon_2\epsilon_3 r + 2\epsilon_1\eta r - 2\epsilon_1\epsilon_3 s + 2\epsilon_2\eta s} \begin{bmatrix} s \\ -r \\ -f \end{bmatrix} \tag{3.12}$$

As the camera frame coincides with the body frame, q used in the derivation above represents the UAV attitude. From the same argument c_z^n is $-h$, with h being the altitude of the UAV.

Alternatively equation (3.12) can be expressed by Euler angles. This might be desirable when using inclinometers to measure the attitude of {c} and {b} w.r.t {n} (Fusini et al., 2015)

$$\boldsymbol{p}^c = \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} = \frac{c_z^n - z^n}{s\sin(\theta) + \cos(\theta)(f\cos(\phi) + r\sin(\phi))} \begin{bmatrix} s \\ -r \\ -f \end{bmatrix} \tag{3.13}$$

Assuming that the terrain is flat would imply z^n in equation (3.12) and equation (3.13) to be zero.

Drawbacks of the ground truth optical flow

First, it should be said that the CV does not actually calculate the OF, but rather the discrete OF. This means that the frame rate of the camera must be sufficiently high in order for the relationship to be valid. Note that CV measures Δr and Δs, and the following relationship holds

$$\dot{r} = \lim_{\Delta t \to 0} \frac{\Delta r}{\Delta t}$$

$$\dot{s} = \lim_{\Delta t \to 0} \frac{\Delta s}{\Delta t}$$

with Δt being the time between two consecutive images are taken. Moreover violating the assumption of flat terrain will cause the GTOF to produce erroneous measurements of the velocity. It is a huge drawback of the GTOF method that one has to assume or know something about the structure being filmed. The GTOF is said to be structure dependent, and will not work properly when flying over terrain with big variations in elevation.

Using normalized body-fixed linear velocity

Later in this thesis the measurements from CV will be normalized before they are used in the observers. As the measurements are normalized, one does not actually need to preserve the scale of the linear velocity. From equation (3.12)-(3.13) it is seen that x^c,y^c and z^c are all proportional to the height of the camera c_z^n. The relationship between the scale of the linear velocity and the depth in the images (distance to the ground) is seen from the three first columns of the matrix M in equation (3.11). It can be seen that if inserting for $\boldsymbol{p}^c = [x^c, y^c, z^c]^T$ the linear velocity is proportional to the height of the camera, c_z^n. This implies that one does not actually have to have a measure of the distance to the ground as long as all features lie on the same elevation (they all lie in a horizontal-plane). Recall that a requirement is that the distance to any feature corresponding to an OF vector needs to be known. When using normalized body-velocity, this requirement however reduces to knowing a scaled distance to all features, with a common arbitrary scale factor for all features. This means that one does not actually need a sensor measuring the distance to the ground when using only normalized body-fixed linear velocity.

3.3.2 Egomotion from the Epipolar Constraint

This section presents a way to calculate the egomotion of a camera based on epipolar geometry (Hartley and Zisserman, 2003), utilizing OF, independent of information about distance, the shape and orientation of the structure being pictured. Epipolar geometry is a well known research field, which relates the motion of the camera frame with the motion in the image plane independent of the structure being filmed. The main problem epipolar geometry concern, is to find $\boldsymbol{t}$ and δR as shown in Figure 3.10. This will now be illustrated.

Denote the camera frame at time t_k $\{c_{k+1}\}$, and at time t_{k+1} by $\{c_{k+1}\}$. $\boldsymbol{t}$ is the translation between the camera frame from $\{c_k\}$ to $\{c_{k+1}\}$, while δR is the rotation matrix relating the attitude of camera from $\{c_k\}$ to $\{c_{k+1}\}$:

$$\boldsymbol{t} \equiv \boldsymbol{t}_{c_k}^{c_{k+1}} = [t_x, t_y, t_z]^T \in \mathbb{R}^3 \tag{3.14}$$

$$\delta R \equiv R_{c_k}^{c_{k+1}} \in \mathrm{SO}(3) \tag{3.15}$$

With epipolar geometry the reconstruction of motion and structure is possible at best up to scale, meaning that one can only determine a dimensionless measure of translation as well as the rotation matrix δR. The epipolar constraint will be derived in equation (3.16)-(3.21). This have been done by multiple researchers, but is included here as reference to the reader.

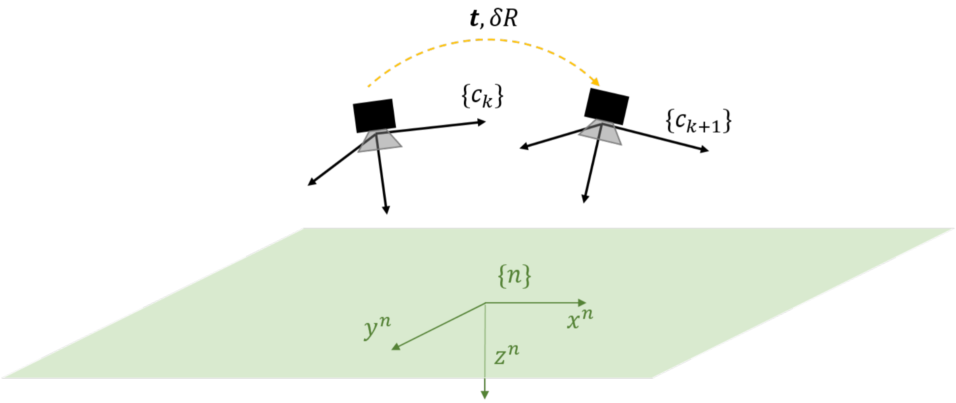


Figure 3.10: With epipolar geometry it is possible to find a dimensionless translation $\boldsymbol{t}$ and the relative rotation, δR, from time t_k to t_{k+1} without any prior information about the structure.

The epipolar constraint

Consider Figure 3.11. The figure explains the different reference frames used in this section. A point $\boldsymbol{y}^c$ is a world point expressed in camera frame coordinates, $\boldsymbol{x}^m$ is the corresponding projected point in the image plane. $\boldsymbol{u}^c$ is the back-projected point lying on the projection ray between {c} origin and $\boldsymbol{y}^c$ with unity z-component.

With Figure 3.11 and the pinhole camera model from equation (3.3) in mind, the following relationship holds:

$$\lambda \underline{\boldsymbol{x}}^m = K\boldsymbol{y}^c \tag{3.16}$$

$$\boldsymbol{u}^c = K^{-1}\underline{\boldsymbol{x}}^m \tag{3.17}$$

K being the projection matrix as defined in equation (3.4) and λ being the distance to the ground.

For a point in the camera frame with a time varying position $\boldsymbol{y}^c(t)$, the following relationship holds

$$\begin{aligned} \lambda(t_k)\underline{\boldsymbol{x}}^m(t_k) &= K\boldsymbol{y}^c(t_k) \\ \lambda(t_{k+1})\underline{\boldsymbol{x}}^m(t_{k+1}) &= K\boldsymbol{y}^c(t_{k+1}) \end{aligned} \tag{3.18}$$

It is assumed that the point $\boldsymbol{y}^c$ is fixed to a rigid body with rotation δR w.r.t. the camera frame, and that the point is subject to a translation $\boldsymbol{t} = [t_x, t_y, t_z]$. Then $\boldsymbol{y}^c(t_k)$ is related to $\boldsymbol{y}^c(t_{k+1})$ by the following:

$$\boldsymbol{y}^c(t_{k+1}) = \delta R\boldsymbol{y}^c(t_k) + \boldsymbol{t} \tag{3.19}$$

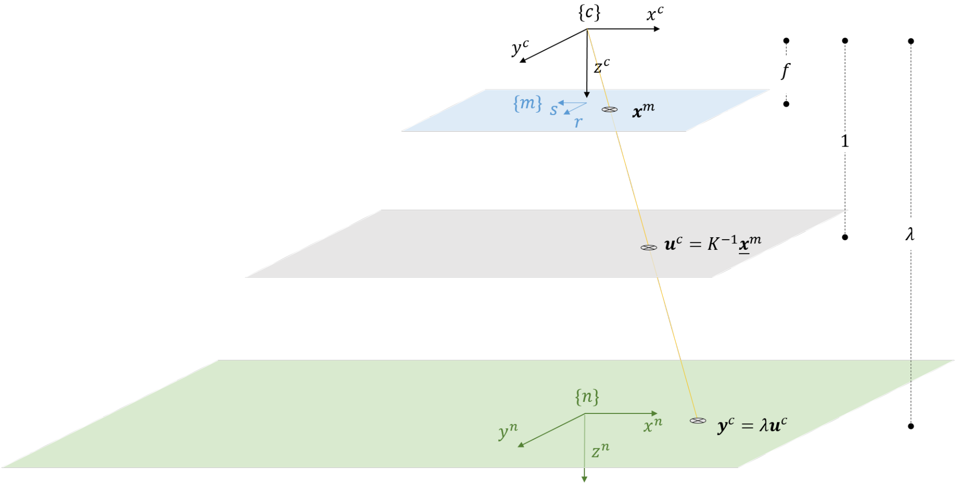

{c}
x^c
y^c
z^c
f
{m}
s
r
$\boldsymbol{x}^m$
1
$\boldsymbol{u}^c = K^{-1}\underline{\boldsymbol{x}}^m$
λ
{n}
y^n
x^n
z^n
$\boldsymbol{y}^c = \lambda \boldsymbol{u}^c$

Figure 3.11: The relationship between Camera frame {c} , image plane {m} illustrated in blue, a back projected plane as illustrated in gray, and the North-East-Down frame {n} in green. The back projected plane is an imaginable plane with unity distance from the camera frame {c} . Image points can be back projected from {m} to this plane by the projection matrix, K as defined in equation (3.4) and the homogeneous image coordinate $\underline{\boldsymbol{x}}^m$. λ is the distance from {c} to {n} and can be seen as the depth in the image. f is the focal length, and the distance between {c} and {m} .

Using the relationship from equation (3.18) it is possible to express equation (3.19) in image plane coordinates

$$\lambda(t_{k+1})\underbrace{K^{-1}\underline{\boldsymbol{x}}^m(t_{k+1})}_{\boldsymbol{u}^c(t_{k+1})} = \delta R(\lambda(t_k)\underbrace{K^{-1}\underline{\boldsymbol{x}}^m(t_k)}_{\boldsymbol{u}^c(t_k)}) + \boldsymbol{t}$$

$$\lambda(t_{k+1})\boldsymbol{u}^c(t_{k+1}) = \lambda(t_k)\delta R\boldsymbol{u}^c(t_k) + \boldsymbol{t} \tag{3.20}$$

This can be simplified by multiplying both sides by $[\boldsymbol{t}]_\times$, and then multiplying with $\boldsymbol{u}^c(t_{k+1})^T$:

$$\lambda(t_{k+1})[\boldsymbol{t}]_\times\boldsymbol{u}^c(t_{k+1}) = \lambda(t_k)[\boldsymbol{t}]_\times\delta R\boldsymbol{u}^c(t_k) + \underbrace{[\boldsymbol{t}]_\times\boldsymbol{t}}_{0}$$

$$\lambda(t_{k+1})[\boldsymbol{t}]_\times\boldsymbol{u}^c(t_{k+1}) = \lambda(t_k)[\boldsymbol{t}]_\times\delta R\boldsymbol{u}^c(t_k)$$

$$\lambda(t_{k+1})\underbrace{\boldsymbol{u}^c(t_{k+1})^T[\boldsymbol{t}]_\times\boldsymbol{u}^c(t_{k+1})}_{0} = \lambda(t_k)\boldsymbol{u}^c(t_{k+1})^T[\boldsymbol{t}]_\times\delta R\boldsymbol{u}^c(t_k)$$

$$\lambda(t_k)\boldsymbol{u}^c(t_{k+1})^T[\boldsymbol{t}]_\times\delta R\boldsymbol{u}^c(t_k) = 0$$

$$\boldsymbol{u}^c(t_{k+1})^T\underbrace{[\boldsymbol{t}]_\times\delta R}_{E}\boldsymbol{u}^c(t_k) = 0$$

$$\boldsymbol{u}^c(t_{k+1})^T E\boldsymbol{u}^c(t_k) = 0 \tag{3.21}$$

Equation (3.21) is the well-known epipolar constraint (Hartley and Zisserman, 2003), with E being the essential matrix. Several methods, such as the eight point algorithm (Maybank and Faugeras, 1992), exist for estimating E, and in turn recovering $\boldsymbol{t}$ and δR (Huang and Netravali, 1994; Zhuang et al., 1986; Hartley, 1997). These methods will not be explained in detail, but they all rely on using multiple discrete OF vectors to estimate E.

Unique solution of the epipolar constraint

The eight point methods (Hartley, 1997; Longuet-Higgins, 1981, 1984) methods fail to produce unique estimates of E in the presence of planar surfaces. Longuet-Higgins (1981, 1984); Maybank and Faugeras (1992); Zhang (1998) have studied whether it is possible to determine an unique solution of the essential matrix E. Their results show that even when a large number of OF vectors, there may exist several solutions for the essential matrix. This means that there are several matrices that explains the data equally well. Such cases are called degenerate cases.

Planar surfaces impose degenerate cases, and even though the earth is not a planar surface the terrain appears planar from large altitudes. The five point methods (Nister, 2003) handles the problem with planar surfaces, but does not provide an unique estimate of $\boldsymbol{t}$ and δR. A specialized version of the eight-point algorithm could be used to estimate E in the presence of planar surfaces (Shakernia et al., 2002). This algorithm is based on a constraint that all points are coplanar, meaning they all lie on a planar surface. This makes it possible to estimate the essential matrix E even though the terrain appear planar from large altitudes, but this implies that it is impossible to determine the motion in presence of non-planar surfaces. Hence the specialized version of the eight-point algorithm is useless for a general case.

Although $\boldsymbol{t}$ and δR can be found from the essential matrix E without any additional information other than the OF, this is not a good way to find the motion of a UAV. As the non-planar terrain may appear planar from large altitudes, any algorithm for estimating E will at some point fail in estimating a unique solution, and this method is hence unsuitable in vision-aided UAV navigation. This motivates to use the gyroscope to estimate the relative rotation matrix δR.

Egomotion from Gyro-Aided Epipolar Geometry

In order to avoid the problem with degenerate configurations, due to the flat terrain, gyroscopic measurements can be used to determine δR. Okatani and Deguchi (2002); Labrie and Hebert (2007) did this by integrating angular velocity measurement from the gyroscope. A IMU can be configured to output incremental angles that can be used to determine δR. When δR is a known quantity, then the problem

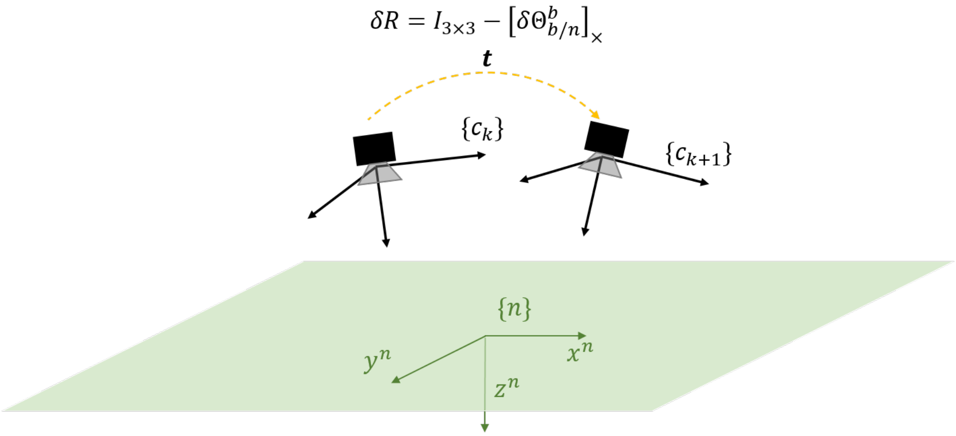

$\delta R = I_{3\times3} - \left[\delta\Theta^b_{b/n}\right]_\times$
$\boldsymbol{t}$
$\{c_k\}$
$\{c_{k+1}\}$
$\{n\}$
y^n
x^n
z^n

Figure 3.12: Gyro aided epipolar geometry. The problem reduces to finding $\boldsymbol{t}$ as δR can be calculated from incremental angles, $\delta\Theta^b_{b/n}$, from an IMU.

of the epipolar constraint reduces to finding only the translation, as illustrated in Figure 3.12. The following will derive a method for recovering $\boldsymbol{t}$ given OF and incremental angles from the IMU. This method is referred to as a epipolar optical flow (EOF) method.

δR can be found by configuring the IMU to output incremental angles, or by integrating locally biased compensated gyro measurements. If incremental angles are known, then this relates to δR by the following:

$$R^n_{c_{k+1}} = R^n_{c_k} R^{c_k}_{c_{k+1}} = R^n_{c_k} \exp\left[\delta\Theta^b_{b/n}\right]_\times \tag{3.22}$$

$$\exp\left[\delta\Theta^b_{b/n}\right]_\times = R^{c_k}_{c_{k+1}} = \delta R^T \approx I_{3\times3} + \left[\delta\Theta^b_{b/n}\right]_\times \tag{3.23}$$

$$\delta R = \exp\left[\delta\Theta^b_{b/n}\right]_\times^T \approx I_{3\times3} - \left[\delta\Theta^b_{b/n}\right]_\times \tag{3.24}$$

$\delta\Theta^b_{b/n}$ being the incremental angle from time t_k to t_{k+1}, defined by the integral of the angular velocity from time t_k to t_{k+1}, that is $\delta\Theta^b_{b/n} = \int_{t_k}^{t_{k+1}} \omega^b_{b/n} dt \approx \Delta t \omega^b_{b/n}$, with $\Delta t = t_{k+1} - t_k$.

The epipolar constraint from equation (3.21) can be expressed as the triple product:

$$\begin{aligned}&\boldsymbol{u}^c(t_{k+1})^T [\boldsymbol{t}]_\times \delta R \boldsymbol{u}^c(t_k)\\ &= \boldsymbol{u}^c(t_{k+1})^T (\boldsymbol{t} \times (\delta R \boldsymbol{u}^c(t_k))) = 0 &(3.25)\\ &\boldsymbol{t}^T ((\delta R \boldsymbol{u}^c(t_k)) \times \boldsymbol{u}^c(t_{k+1})) = 0 &(3.26)\end{aligned}$$

Defining now the crossproduct term as $\boldsymbol{d}$:

$$\boldsymbol{d} = (\delta R\boldsymbol{u}^c(t_k) \times \boldsymbol{u}^c(t_{k+1})) = [d_x, d_y, d_z]^T \tag{3.27}$$

Then equation (3.25) can be written as an equation linear in $\boldsymbol{t}$

$$\boldsymbol{t}^T\boldsymbol{d} = 0 \tag{3.28}$$

It desirable to solve equation (3.28) w.r.t $\boldsymbol{t}$, as $\boldsymbol{t}$ is the direction of translation. Recall that the motion of the camera is desired. The camera is fixed to a fixed-wing UAV, with {c} aligned with {b} . Then, a viable assumption is that the camera has some movement in the x^c direction, as a fixed-wing UAV must have some forward velocity to remain airborne. Given that the frame rate of the camera is sufficiently large, such that the UAV will have some movement in x^b-direction in the time between two images, the following is ensured:

$$t_x \neq 0$$

Since $t_x \neq 0$, it is possible to divide equation (3.28) by t_x, yielding a non-homogeneous linear equation. A non-homogeneous linear system $A\boldsymbol{x} = \boldsymbol{b}, A \in \mathcal{R}^{n\times n}$ has, if A is invertible, exactly one solution. This is advantageous as it would yield an unique solution of $\boldsymbol{x}$. Dividing equation (3.28) by t_x yields

$$\frac{t_y}{t_x}d_y + \frac{t_z}{t_x}d_z = -d_x \tag{3.29}$$

If now extending this to concern any feature j (yielding a corresponding $\boldsymbol{d}_j$), this equation might be expressed as

$$\frac{1}{t_x}\underbrace{\begin{bmatrix} d_{y,j} & d_{z,j} \end{bmatrix}}_{A_{d,j}} \begin{bmatrix} t_y \\ t_z \end{bmatrix} = \underbrace{-d_{x,j}}_{\boldsymbol{b}_{d,j}}, t_x \neq 0 \tag{3.30}$$

Given N image features, yielding N $A_{d,j}$'s and $\boldsymbol{b}_{d,j}$'s, A_d and $\boldsymbol{b}_d$ are defined as

$$A_d = \begin{bmatrix} d_{y,1} & d_{z,1} \\ \vdots & \\ d_{y,N} & d_{z,N} \end{bmatrix}, \quad \boldsymbol{b}_d = -\begin{bmatrix} d_{x,1} \\ \vdots \\ d_{x,N} \end{bmatrix} \tag{3.31}$$

Assume now that A_d has full rank, such that A_d^+ exists, and $t_x \neq 0$. The assumption of A_d having full rank is complied if the OF vectors satisfy Definition 7. Sufficient and necessary conditions for Definition 7 is discussed later in this section. Using multiple image features, solving equation (3.30) can be done with respect to $[t_y, t_z]^T$.

$$\begin{bmatrix} t_y \\ t_z \end{bmatrix} = t_x A_d^+ \boldsymbol{b}_d \tag{3.32}$$

Note that there is a sign ambiguity, as both the vector pointing in the direction of translation and the vector pointing in the opposite direction will fulfil equation (3.32). If the sign of the x-component of $\boldsymbol{t}$, t_x, is known, then $\boldsymbol{t}_e$ can be used to find the direction of translation of the camera. Now defining $\boldsymbol{t}_e$ as a dimensionless measure of the direction of translation with first component of unity length.

$$\boldsymbol{t}_e = \frac{\boldsymbol{t}}{t_x} = \left[1, \frac{t_y}{t_x}, \frac{t_z}{t_x}\right]^T = \left[1, (A_d^{+}\boldsymbol{b}_d)^T\right]^T, \quad t_x \neq 0 \tag{3.33}$$

If the sign of t_x is known, then the direction of translation is found as

$$\frac{\boldsymbol{t}}{\|\boldsymbol{t}\|} = \frac{\boldsymbol{t}_e}{\|\boldsymbol{t}_e\|} \tag{3.34}$$

Definition 7. ***Linearly Independent Image Features***

A pair of image features at time t_k and t_{k+1},$\boldsymbol{x}_1^m(t_k)$, $\boldsymbol{x}_1^m(t_{k+1})$ and $\boldsymbol{x}_2^m(t_k)$, $\boldsymbol{x}_2^m(t_{k+1})$, are said to be linearly independent if and only if the rank of A in equation (3.31) *is full, yielding $[t_y, t_z]^T = t_x A_d^{+}\boldsymbol{b}$ to be uniquely defined. The rank of A_d is full if and only if a 2×2 submatrix of A_d, $A_{d,2\times 2}$ has determinant not equal to zero. That is $det(A_{d,2\times 2}) \neq 0$.*

Approximating body-fixed velocity from the direction of translation

As the time between two images $\Delta t \to 0$, $\boldsymbol{t}_e$ can be used to find the instantaneous normalized camera velocity:

$$\frac{\boldsymbol{v}_{c/n}^c}{\|\boldsymbol{v}_{c/n}^c\|} = \lim_{\Delta t\to 0} \text{sign}(t_x)\frac{\boldsymbol{t}_e/\Delta t}{\|\boldsymbol{t}_e/\Delta t\|} = \lim_{\Delta t\to 0} \text{sign}(t_x)\frac{\boldsymbol{t}_e}{\|\boldsymbol{t}_e\|} \tag{3.35}$$

Moreover as {c} coincides with {b} , $\boldsymbol{v}_{c/n}^c = \boldsymbol{v}_{b/n}^b$. As can be seen from equation (3.35) $\boldsymbol{t}_e$ can be used as a measure of the normalized body velocity when the time between two consecutive images $\Delta t \to 0$.

Equation (3.32) is only solvable if there are some linearly independent image features at time t_k and t_{k+1}, as defined in Definition 7.

Discussion: Sufficient and necessary conditions for linearly independent image features

As Definition 7 might be hard to relate to the OF vectors some sufficient and necessary conditions are discussed. Given two discrete OF vectors $[dr_1, ds_1]^T$, $[dr_2, ds_2]^T$

corresponding to the time from t_k to t_{k+1}, yielding two corresponding back-projected point pairs $\boldsymbol{u}_1^c(t_k), \boldsymbol{u}_1^c(t_{k+1})$ and $\boldsymbol{u}_2^c(t_k), \boldsymbol{u}_2^c(t_{k+1})$.

$$\boldsymbol{u}_1^c(t_k) = K^{-1}[r_1, s_1, 1]^T \qquad \boldsymbol{u}_1^c(t_{k+1}) = K^{-1}[r_1 + dr_1, s_1 + ds_1, 1]^T \tag{3.36}$$

$$\boldsymbol{u}_2^c(t_k) = K^{-1}[r_2, s_2, 1]^T \qquad \boldsymbol{u}_2^c(t_{k+1}) = K^{-1}[r_2 + dr_2, s_2 + ds_2, 1]^T \tag{3.37}$$

K being the projective mapping as defined in equation (3.4). Moreover assume that δR is known. The corresponding crossproduct terms are then $\boldsymbol{d}_1 = \delta R \boldsymbol{u}_1^c(t_k) \times \boldsymbol{u}_1^c(t_{k+1})$ and $\boldsymbol{d}_2 = \delta R \boldsymbol{u}_2^c(t_k) \times \boldsymbol{u}_2^c(t_{k+1})$. Letting the unit vectors in x^c, y^c and z^c direction be denoted $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ respectively. Then $det(A_{d,2\times2})$ can be expressed as:

$$det(A_{d,2\times2}) = det\left(\begin{bmatrix} \boldsymbol{d}_1^T\boldsymbol{j} & \boldsymbol{d}_1^T\boldsymbol{k} \\ \boldsymbol{d}_2^T\boldsymbol{j} & \boldsymbol{d}_2^T\boldsymbol{k} \end{bmatrix}\right) \tag{3.38}$$

$$= \boldsymbol{d}_1^T\boldsymbol{j}\boldsymbol{d}_2^T\boldsymbol{k} - \boldsymbol{d}_1^T\boldsymbol{k}\boldsymbol{d}_2^T\boldsymbol{j} \tag{3.39}$$

$$= \boldsymbol{d}_{2,z}\boldsymbol{d}_{1,y} - \boldsymbol{d}_{1,z}\boldsymbol{d}_{2,y} \tag{3.40}$$

$$= (\boldsymbol{d}_1 \times \boldsymbol{d}_2)^T\boldsymbol{i} \tag{3.41}$$

The sufficient and necessary condition for a pair of image features to be linearly independent is therefore $(\boldsymbol{d}_1 \times \boldsymbol{d}_2)^T\boldsymbol{i} \neq 0$. A pair of image features that are not linearly independent are said to be linearly dependent.

Degeneracy will happen if all vectors $\boldsymbol{u}_j^c(t_k)$ and $\boldsymbol{u}_j^c(t_{k+1})$ are linearly dependent, yielding or $(\boldsymbol{d}_1 \times \boldsymbol{d}_2) = 0$, or if there is no flow in x^c direction, causing $(\boldsymbol{d}_1 \times \boldsymbol{d}_2)^T\boldsymbol{i} = 0$. Notice that the fact that no flow in x^c direction will cause a degeneracy, implies that the camera must have some movement. Figure 3.13 illustrates some linearly independent and dependent configurations of pair of image features.

Advantages and disadvantages with the EOF method

A drawback of the EOF method is the fact that it measures the direction of translation, and not direction of body-fixed velocity. As the frame rate gets sufficiently high, the time between two consecutive images Δt decreases, and a reasonable approximation is that the direction of translation is the same as the direction of the body-fixed velocity. Another drawback is the fact that the EOF method does not preserve the scale of the velocity. The main advantage is that it is structure independent, meaning there is no difference in flying over flat or hilly terrain. The velocity direction estimate is not affected by the structure of the terrain, meaning one does not have to have some previous information about the terrain being filmed. Moreover the EOF does not depend on the height above ground, nor the orientation relative to the terrain being filmed. The only necessary information is the OF and the incremental angles or angular velocity from a gyroscope.

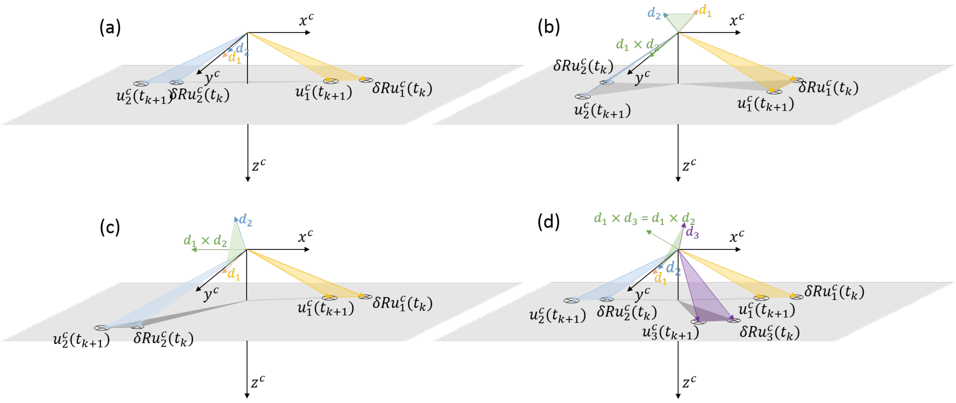


Figure 3.13: (a) Linearly dependent: $\boldsymbol{d}_1 \times \boldsymbol{d}_2$ does not exist. (b) Linearly dependent: $\boldsymbol{d}_1 \times \boldsymbol{d}_2$ exists, but is orthogonal to camera x-axis, $\boldsymbol{x}^c$. (c) Linearly independent: $\boldsymbol{d}_1 \times \boldsymbol{d}_2$ exists and has a component in camera x-axis, $\boldsymbol{x}^c$. (d) Linearly independent: $\boldsymbol{d}_1 \times \boldsymbol{d}_2$ does not exist. $\boldsymbol{d}_1 \times \boldsymbol{d}_3 = \boldsymbol{d}_2 \times \boldsymbol{d}_3$ exists and have a component in camera x-axis, $\boldsymbol{x}^c$.

3.3.3 Egomotion from the Continuous Epipolar Constraint

Another variant of the epipolar constraint was presented by Ma et al. (2001). Ma et al. (2001) presented the epipolar constraint in the continuous case. The continuous epipolar constraint can be used to calculate the body-fixed linear velocity of the camera fixed to the UAV (Grabe, Bulthoff and Robuffo Giordano, 2012; Grabe, Bulthoff and Giordano, 2012). For a reference to the reader, the derivation of the continuous epipolar constraint is included in equation (3.42)-(3.49).

Given a feature j with position $\boldsymbol{y}(t)$. The point may be expressed in {c} as $\boldsymbol{y}^c(t)$. The motion of this feature relative to the camera can be stated as

$$\dot{\boldsymbol{y}}^c(t) = \left[\boldsymbol{\omega}^c_{j/c}(t)\right]_\times \boldsymbol{y}^c(t) + \boldsymbol{v}^c_{j/c}(t) \tag{3.42}$$

with $\boldsymbol{\omega}^c_{j/c}$ and $\boldsymbol{v}^c_{j/c}$ being the angular and linear velocity of the feature relative to the camera respectively. From now omitting the time t in the parentheses, noting that the vectors are time variant. From equation (3.3) it is known that $\boldsymbol{y}^c = z^c K^{-1} \underline{\boldsymbol{x}}^m \equiv z^c \boldsymbol{u}^c$, with $\underline{\boldsymbol{x}}^m = [r, s, 1]^T$, $\boldsymbol{u}^c$ as defined in equation (3.17). Consider now the following triple product:

$$\dot{\boldsymbol{y}}^{cT}\left(\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c\right) = (\left[\boldsymbol{\omega}^c_{j/c}\right]_\times \boldsymbol{y}^c + \boldsymbol{v}^c_{j/c})^T (\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c) = \boldsymbol{y}^{cT}\left[\boldsymbol{\omega}^c_{j/c}\right]_\times^T (\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c) \tag{3.43}$$

This implies

$$\dot{\boldsymbol{y}}^{cT}\left(\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c\right) - \boldsymbol{y}^{cT}\left[\boldsymbol{\omega}^c_{j/c}\right]_\times^T (\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c) = 0 \tag{3.44}$$

Furthermore notice that

$$\dot{\boldsymbol{y}}^c = \dot{z}^c \boldsymbol{u}^c + z^c \dot{\boldsymbol{u}}^c \tag{3.45}$$

With $\dot{\boldsymbol{u}}^c = K^{-1}\underline{\dot{\boldsymbol{x}}}^m = K^{-1}[\dot{r}, \dot{s}, 0]^T$ being the back projected OF. Then the following holds:

$$\left(\dot{z}^c \boldsymbol{u}^c + z^c \dot{\boldsymbol{u}}^c\right)^T \left(\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c\right) - z^c \boldsymbol{u}^{cT} \left[\boldsymbol{\omega}^c_{j/c}\right]_\times^T \left(\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c\right) = 0 \tag{3.46}$$

Using now that $\boldsymbol{u}^T(\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}) = 0$, the following is obtained

$$z^c \dot{\boldsymbol{u}}^{cT}(\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c) - z^c \boldsymbol{u}^{cT} \left[\boldsymbol{\omega}^c_{j/c}\right]_\times^T (\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c) = 0 \tag{3.47}$$

When the depth in the image $z^c \neq 0$, one can divide equation (3.47) by z^c. Then the scale of the motion parameter $\boldsymbol{v}^c_{j/c}$ are lost. As it is only desirable to use the normalized body-fixed linear velocity, this introduces no problems. When dividing equation (3.47) by z^c one ends up at

$$\dot{\boldsymbol{u}}^{cT}(\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c) - \boldsymbol{u}^{cT} \left[\boldsymbol{\omega}^c_{j/c}\right]_\times^T (\boldsymbol{v}^c_{j/c} \times \boldsymbol{u}^c) = 0 \tag{3.48}$$

The objective is to recover the body-fixed velocity of the camera, that is $\boldsymbol{v}^c_{c/n}$. Assuming now that every feature is fixed to {n} , the following holds:

$$\boldsymbol{v}^c_{c/n} = -\boldsymbol{v}^c_{j/c} = [v_x, v_y, v_z]^T$$
$$\boldsymbol{\omega}^c_{c/n} = -\boldsymbol{\omega}^c_{j/c}$$

Then equation (3.48) might be rewritten as

$$\begin{aligned}
\dot{\boldsymbol{u}}^{cT}(-\boldsymbol{v}^c_{c/n} \times \boldsymbol{u}^c) - \boldsymbol{u}^{cT} \left[-\boldsymbol{\omega}^c_{c/n}\right]_\times^T (-\boldsymbol{v}^c_{c/n} \times \boldsymbol{u}^c) &= 0 \\
-\dot{\boldsymbol{u}}^{cT}(\boldsymbol{v}^c_{c/n} \times \boldsymbol{u}^c) - \boldsymbol{u}^{cT} \left[\boldsymbol{\omega}^c_{c/n}\right]_\times^T (\boldsymbol{v}^c_{c/n} \times \boldsymbol{u}^c) &= 0 \\
-\left(\dot{\boldsymbol{u}}^{cT} + \boldsymbol{u}^{cT} \left[\boldsymbol{\omega}^c_{c/n}\right]_\times^T\right) (\boldsymbol{v}^c_{c/n} \times \boldsymbol{u}^c) &= 0
\end{aligned} \tag{3.49}$$

This is the continuous epipolar constraint (Ma et al., 2001). Using now the properties of a triple product (Kreyszig, 2006), and ignoring the minus sign, equation (3.49) can be rewritten as:

$$\begin{aligned}
&{\boldsymbol{v}^c_{c/n}}^T \left(\boldsymbol{u}^c \times \left(\dot{\boldsymbol{u}}^{cT} + \boldsymbol{u}^{cT} \left[\boldsymbol{\omega}^c_{c/n}\right]_\times^T\right)^T\right) \\
&= {\boldsymbol{v}^c_{c/n}}^T \left(\boldsymbol{u}^c \times \left(\dot{\boldsymbol{u}}^{cT} + \left[\boldsymbol{\omega}^c_{c/n}\right]_\times \boldsymbol{u}^c\right)\right) = 0
\end{aligned} \tag{3.50}$$

Following now the same procedure as in Section 3.3.2, equation (3.50) might be rewritten as linear equation in $\boldsymbol{v}^c_{c/n}$. Defining the crossproduct term as follows:

$$\boldsymbol{c} = \boldsymbol{u}^c \times \left(\dot{\boldsymbol{u}}^{cT} + \left[\boldsymbol{\omega}^c_{c/n} \right]_\times \boldsymbol{u}^c \right) = [c_x, c_y, c_z]^T$$

Using this equation (3.50) is rewritten as

$$\boldsymbol{v}^c_{c/n}{}^T \boldsymbol{c} = \boldsymbol{c}^T \boldsymbol{v}^c_{c/n} = 0 \tag{3.51}$$

Assuming that a fixed-wing UAV will never have zero forward velocity and that {c} and {b} are aligned, one can divide equation (3.51) by the forward velocity component v_x in order to obtain a non-homogeneous linear equation:

$$\frac{1}{v_x} \boldsymbol{c}^T \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \boldsymbol{c}^T \begin{bmatrix} 1 \\ \frac{v_y}{v_x} \\ \frac{v_z}{v_x} \end{bmatrix} = 0 \tag{3.52}$$

$$c_x + [c_y, c_z] \begin{bmatrix} \frac{v_y}{v_x} \\ \frac{v_z}{v_x} \end{bmatrix} = 0 \tag{3.53}$$

$$[c_y, c_z] \begin{bmatrix} \frac{v_y}{v_x} \\ \frac{v_z}{v_x} \end{bmatrix} = -c_x \tag{3.54}$$

As can be seen from equation (3.54), one ends up with a similar equation of what was obtained for the translational direction in equation (3.30). Assuming N features with corresponding OF vectors, whereas some of the OF vectors are linearly independent as defined in Definition 8. The body-fixed velocity with can be found as:

$$\begin{aligned} \boldsymbol{v}^c_{c/n} &= v_x A_c^+ \boldsymbol{b}_c \quad v_x \neq 0 \\ A_c &= \begin{bmatrix} c_{y,1} & c_{z,1} \\ \vdots & \\ c_{y,N} & c_{z,N} \end{bmatrix} \\ \boldsymbol{b}_c &= - \begin{bmatrix} c_{x,1} \\ \vdots \\ c_{x,N} \end{bmatrix} \\ \boldsymbol{c}_j &= \boldsymbol{u}^c_j \times \left(\dot{\boldsymbol{u}}^{cT}_j + \left[\boldsymbol{\omega}^c_{c/n} \right]_\times \boldsymbol{u}^c_j \right) = [c_{x,j}, c_{y,j}, c_{z,j}]^T \end{aligned} \tag{3.55}$$

$\boldsymbol{u}^c_j = K^{-1} \underline{\boldsymbol{x}}^m_j$ and $\dot{\boldsymbol{u}}^c_j = K^{-1} [\dot{r}_j, \dot{s}_j, 0]^T$ being the back projected coordinate and OF of feature j respectively.

Definition 8. ***Linearly Independent Optical Flow Vectors***

A pair of image features and their corresponding OF vectors $\boldsymbol{x}^m_1$, $\dot{\boldsymbol{x}}^m_1$ and $\boldsymbol{x}^m_2$, $\dot{\boldsymbol{x}}^m_2$, are said to be linearly independent if and only if the rank of A_c in equation (3.55) *is full, yielding $[v_y, v_z]^T = v_x A_c^+ \boldsymbol{b}_c$ to be uniquely* *defined. The rank of A_c is full if and only if a 2×2 submatrix of A_c, $A_{c,2\times 2}$ has determinant not equal to zero. That is det$(A_{c,2\times 2}) \neq 0$.*

An expression for the body-fixed linear velocity is obtained in equation (3.55). This relation can be used to extract information from a camera that can be used in a navigation system.

Chapter 4

Multiplicative Extended Kalman Filter for UAV Navigation

In this chapter a Multiplicative Extended Kalman Filter (MEKF) (Lefferts et al., 1982; Markley, 2004) for navigation of a UAV is designed. The objective of the MEKF is to estimate the navigation states (attitude, position, velocity, gyro- and accelerometer bias) of a fixed-wing UAV. The MEKF is an extension of the famous Extended Kalman Filter (EKF) (Schmidt, 1981; Crassidis et al., 2007). The MEKF in this chapter is designed to make use of measurements of angular velocity and specific force from a gyroscope and an accelerometer respectively, body-fixed linear velocity from a computer vision (CV) system, and velocity and position measurements from a Global Navigation Satellite System (GNSS). Further information about the sensors, and their measurement equations are given in Appendix H. Some brief introduction to the MEKF will be given before the design of the MEKF is presented. In this thesis {n} is assumed inertial, as motivated Appendix F.

The motivation behind the MEKF is to bypass the problem of quaternions introducing a singularity in the covariance matrix, P, due to the unity constraint of the unit quaternion (Lefferts et al., 1982). This singularity appears due to the lack of independence between the four elements of a unit quaternion. By estimating only the imaginary part of the error quaternion one are able to avoid the aforementioned singularity.

This chapter is organized in four sections. Section 4.1 give some basic introduction to the objective of the MEKF. A MEKF for estimating gyro bias and attitude of a UAV, utilizing GNSS, IMU and CV measurements, is designed in Section 4.2. The MEKF is expanded in Section 4.3 to include translational motion such as position, velocity and accelerometer bias. The chapter ends with an outline of the MEKF algorithm in Section 4.4

4.1 Objective of the MEKF

In the MEKF the following relationship is used

$$q = \hat{q} \otimes \delta q(\delta \boldsymbol{\epsilon}) \tag{4.1}$$

$$\delta q(\delta \boldsymbol{\epsilon}) = \begin{bmatrix} \delta \eta \\ \delta \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \sqrt{1 - \delta \boldsymbol{\epsilon}^T \delta \boldsymbol{\epsilon}} \\ \delta \boldsymbol{\epsilon} \end{bmatrix} \tag{4.2}$$

with $\hat{q}$ being a nonsingular estimated unit quaternion, q being the real quaternion and $\delta q(\delta \boldsymbol{\epsilon})$ being the error in our estimate. $\otimes$ is denoting the quaternion product as described in Appendix G. As seen the estimation of $\delta \boldsymbol{\epsilon}$ can be done without taking the unit constraint of the quaternion into consideration, because the unity constraint is enforced when creating δq from $\delta \boldsymbol{\epsilon}$.

In the MEKF the objective is to estimate $\delta \boldsymbol{\epsilon}$, the imaginary part of the error quaternion, which can be used to generate the error quaternion. The error quaternion tells us how far from the real value the estimated quaternion $\hat{q}$ is. When knowing what the error is, one can update the estimate $\hat{q}$ by the following formula:

$$\hat{q}_{k-1} \otimes \delta q(\delta \hat{\boldsymbol{\epsilon}_k}) = \hat{q}_k \otimes \delta q(0_{3\times 1}) = \hat{q}_k \tag{4.3}$$

After updating $\hat{q}$, $\delta \boldsymbol{\epsilon}$ is reset to zero.

4.2 Attitude estimation

Lets in the following assume constant velocity, $\dot{\boldsymbol{v}}^n_{b/n} = 0$. By assuming constant velocity, the accelerometer measures the gravity vector in body coordinates, $\boldsymbol{g}^b$. The CV outputs a measure of the body-fixed linear velocity. Neglecting the accelerometer bias, the following measurement equations can be used to determine the attitude of the vehicle.

$$\boldsymbol{f}^b_{\text{imu}} = -\boldsymbol{g}^b + \boldsymbol{w}_{\text{acc}} \tag{4.4}$$

$$\boldsymbol{v}^b_{\text{cv}} = \boldsymbol{v}^b_{b/n} + \boldsymbol{w}_{\text{cv}} \tag{4.5}$$

In contradiction to when dealing with magnetometer measurements, the CV system does not measure an inertial reference vector. A typical MEKF enables measurements of the magnetic field and gravity, which are indeed inertial and known vectors. Lets however by now assume that the body velocity in {n} is known. The attitude observer is seen in Figure 4.1.

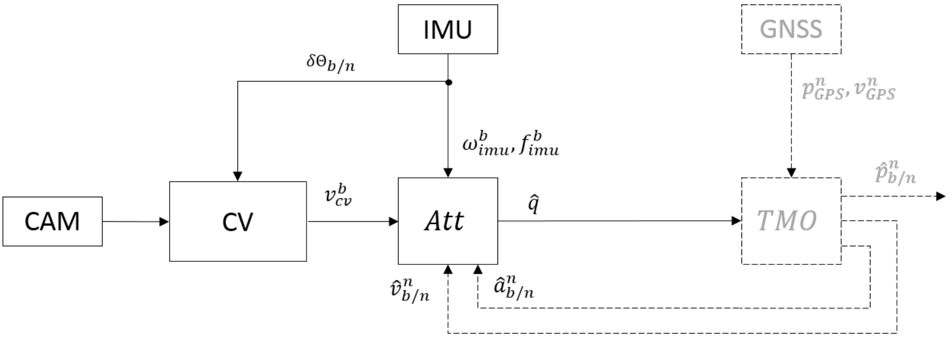


Figure 4.1: Block diagram of the attitude MEKF. Att represents the attitude observer, and TMO translational motion observer.

Moreover it of interest to know the dynamics of δq, as it is desirable to estimate this vector. The time derivative of δq can be found by differentiating equation (4.1).

$$\delta q = \begin{bmatrix} \delta\eta \\ \delta\boldsymbol{\epsilon} \end{bmatrix} = \hat{q}^{-1} \otimes q \tag{4.6}$$

$$\dot{\delta q} = \dot{\hat{q}}^{-1} \otimes q + \hat{q}^{-1} \otimes \dot{q} \tag{4.7}$$

As the estimated attitude $\hat{q}$ does not change between updates, it can been used that $\dot{\hat{q}} = 0$:

$$\dot{\delta q} = \hat{q}^{-1} \otimes \dot{q} \tag{4.8}$$

$$= \frac{1}{2}\hat{q}^{-1} \otimes q \otimes \boldsymbol{\omega}^b_{b/n} \tag{4.9}$$

$$= \frac{1}{2}\delta q \otimes \boldsymbol{\omega}^b_{b/n} \tag{4.10}$$

where it have been used that $\dot{q} = \frac{1}{2} q \otimes \boldsymbol{\omega}^b_{b/n}$. Using the quaternion rules from Appendix G, this quaternion product might be expressed as a matrix multiplication:

$$\dot{\delta q} = \frac{1}{2} \begin{bmatrix} \delta\eta & -\delta\boldsymbol{\epsilon}^T \\ \delta\boldsymbol{\epsilon} & \delta\eta I_{3\times3} + [\delta\boldsymbol{\epsilon}]_\times \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b_{b/n} \end{bmatrix} \tag{4.11}$$

$$= \frac{1}{2} \begin{bmatrix} -\delta\boldsymbol{\epsilon}^T \boldsymbol{\omega}^b_{b/n} \\ \delta\eta I_{3\times3}\boldsymbol{\omega}^b_{b/n} + [\delta\boldsymbol{\epsilon}]_\times \boldsymbol{\omega}^b_{b/n} \end{bmatrix} \tag{4.12}$$

However it is desirable to model the behaviour of $\delta\boldsymbol{\epsilon}$, as this is the quantity we want to estimate. Using Equation (4.2) together with Equation (4.12) the dynamics of $\delta\boldsymbol{\epsilon}$ is obtained

$$\dot{\delta\boldsymbol{\epsilon}} = \frac{1}{2}\left(\sqrt{1 - \delta\boldsymbol{\epsilon}^T\delta\boldsymbol{\epsilon}} I_{3\times3}\boldsymbol{\omega}^b_{b/n} + [\delta\boldsymbol{\epsilon}]_\times \boldsymbol{\omega}^b_{b/n}\right) \tag{4.13}$$

The accelerometer and gyroscope have biases which are modelled as Wiener processes. The gyroscope measurement equation can be written as $\boldsymbol{\omega}^b_{\text{imu}} = \boldsymbol{\omega}^b_{b/n} + \boldsymbol{b}^b_{\text{gyro}} + \boldsymbol{w}^b_{\text{gyro}}$, and in turn rewritten as $\boldsymbol{\omega}^b_{b/n} = \boldsymbol{\omega}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{gyro}} + \boldsymbol{w}^b_{\text{gyro}}$. Using this and the gyroscopic bias model (Wiener process) one can write the attitude and bias dynamics on vector form:

$$\underbrace{\begin{bmatrix} \delta\dot{\boldsymbol{\epsilon}} \\ \dot{\boldsymbol{b}}^b_{\text{gyro}} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \frac{1}{2}\left(\sqrt{1-\delta\boldsymbol{\epsilon}^T\delta\boldsymbol{\epsilon}}I_{3\times3} + [\delta\boldsymbol{\epsilon}]_\times\right)\left(\boldsymbol{\omega}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{gyro}}\right) \\ 0_{3\times1} \end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})} + \underbrace{I_{6\times6}\boldsymbol{w}}_{\Gamma\boldsymbol{w}} \tag{4.14}$$

$$\boldsymbol{x} = \begin{bmatrix} \delta\boldsymbol{\epsilon} \\ \boldsymbol{b}^b_{\text{gyro}} \end{bmatrix} \tag{4.15}$$

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{\omega}^b_{\text{imu}} \end{bmatrix} \tag{4.16}$$

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}^b_{\text{gyro}} \\ \boldsymbol{w}^b_{\text{bgyro}} \end{bmatrix} \tag{4.17}$$

The process noise $\boldsymbol{w}$ is assumed white with zero mean and variance $\Sigma_w \in \mathcal{R}^{6\times6}$

$$\boldsymbol{w} \sim \mathcal{N}(0_{6\times1}, \Sigma_w)$$

The measurements used in the attitude filter are the measured body acceleration, which is a measure of the gravity vector, given a constant velocity, and the body-fixed velocity from CV.

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{f}^b_{\text{imu}} \\ \boldsymbol{v}^b_{\text{cv}} \end{bmatrix} = \underbrace{\begin{bmatrix} -R^{\hat{b}}_b(\delta q)^T R^n_{\hat{b}}(\hat{q})^T \boldsymbol{g}^n \\ R^{\hat{b}}_b(\delta q)^T R^n_{\hat{b}}(\hat{q})^T \boldsymbol{v}^n_{b/n} \end{bmatrix}}_{\boldsymbol{h}(x)} + \underbrace{\begin{bmatrix} \boldsymbol{w}_{\text{acc}} \\ \boldsymbol{w}_{\text{cv}} \end{bmatrix}}_{\boldsymbol{v}} \tag{4.18}$$

The measurement noise is assumed white with variance $\Sigma_v \in \mathcal{R}^{6\times6}$

$$\boldsymbol{v} \sim \mathcal{N}(0_{6\times1}, \Sigma_v)$$

The measured quantity from CV can take on two forms. One form which preserves the scale of the body-fixed linear velocity, and one form that does not preserve the scale of the velocity. When ignoring the scale of the measurement, one are still able to measure the direction of travel, namely normalized body-fixed linear velocity. Then the measurement equation takes on the form as shown below:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{f}^b_{\text{imu}} \\ \boldsymbol{v}^b_{\text{cv}} \end{bmatrix} = \underbrace{\begin{bmatrix} -R^{\hat{b}}_b(\delta q)^T R^n_{\hat{b}}(\hat{q})^T \boldsymbol{g}^n \\ R^{\hat{b}}_b(\delta q)^T R^n_{\hat{b}}(\hat{q})^T \frac{\boldsymbol{v}^n_{b/n}}{\|\boldsymbol{v}^n_{b/n}\|} \end{bmatrix}}_{\boldsymbol{h}(x)} + \underbrace{\begin{bmatrix} \boldsymbol{w}_{\text{acc}} \\ \boldsymbol{w}_{\text{cv}} \end{bmatrix}}_{\boldsymbol{v}} \tag{4.19}$$

4.2.1 Discrete time equations

As the MEKF is supposed to run on a computer, equation (4.14)-(4.19) must be discretized. This can be done in multiple ways (Egeland and Gravdahl, 2002), but the easiest method might be Euler discretizing:

$$\boldsymbol{x}_{k+1} = \underbrace{\boldsymbol{x}_k + h\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)}_{\boldsymbol{f}_k} + \underbrace{h\Gamma}_{\Gamma_k} \boldsymbol{w}_k \tag{4.20}$$

$$\boldsymbol{y}_k = \underbrace{\boldsymbol{h}(\boldsymbol{x}_k)}_{\boldsymbol{h}_k} + \boldsymbol{v}_k \tag{4.21}$$

Equation (4.20)-(4.21) is then linearized, yielding discrete time matrices that can be applied to the MEKF algorithm in Section 4.4.

$$Q_k = \Sigma_{wk} \tag{4.22}$$

$$R_k = \Sigma_{vk} \tag{4.23}$$

$$\boldsymbol{f}_k(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) = \hat{\boldsymbol{x}}_k + h \begin{bmatrix} \frac{1}{2}\left[\boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{gyro}}\right] \\ 0_{3\times 1} \end{bmatrix} \tag{4.24}$$

$$\Phi_k = \left.\frac{\delta \boldsymbol{f}_k}{\delta x_k}\right|_{x_k = \hat{x}_k} = I_{6\times 6} + h \begin{bmatrix} -\frac{1}{2}\left[\boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{gyro}}\right]_\times & -\frac{1}{2} I_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix} \tag{4.25}$$

$$\Gamma_k = hI_{6\times 6} \tag{4.26}$$

$$H_k = \left.\frac{\delta \boldsymbol{h}_k}{\delta x_k}\right|_{x_k = \bar{x}_k} = \begin{bmatrix} -W(\delta\bar{\boldsymbol{\epsilon}}_k, R^n_{\hat{b}}(\hat{q})^T \boldsymbol{g}^n) & 0_{3\times 3} \\ W(\delta\bar{\boldsymbol{\epsilon}}_k, R^n_{\hat{b}}(\hat{q})^T \boldsymbol{v}^n_{b/n}) & 0_{3\times 3} \end{bmatrix} \tag{4.27}$$

$$W(\boldsymbol{\epsilon}, \boldsymbol{v}) = 2\sqrt{1 - \boldsymbol{\epsilon}^T\boldsymbol{\epsilon}}[\boldsymbol{v}]_\times - \frac{2}{\sqrt{1 - \boldsymbol{\epsilon}^T\boldsymbol{\epsilon}}}[\boldsymbol{v}]_\times \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + 2\boldsymbol{v}^T\boldsymbol{\epsilon} I_{3\times 3} + 2\boldsymbol{\epsilon}\boldsymbol{v}^T - 4\boldsymbol{v}\boldsymbol{\epsilon}^T \tag{4.28}$$

When using normalized body velocities equation (4.27) is replaced by equation (4.29):

$$H_k = \left.\frac{\delta \boldsymbol{h}_k}{\delta x_k}\right|_{x_k = \bar{x}_k} = \begin{bmatrix} -W(\delta\bar{\boldsymbol{\epsilon}}_k, R^n_{\hat{b}}(\hat{q})^T \boldsymbol{g}^n) & 0_{3\times 3} \\ W(\delta\bar{\boldsymbol{\epsilon}}_k, R^n_{\hat{b}}(\hat{q})^T \frac{\boldsymbol{v}^n_{b/n}}{\|\boldsymbol{v}^n_{b/n}\|}) & 0_{3\times 3} \end{bmatrix} \tag{4.29}$$

4.3 Position, Velocity and Attitude Estimation

A filter for determining position, velocity and attitude will now be presented. When doing this, the assumption of constant speed introduced in the attitude filter may be relaxed. The state variables are chosen to be $\boldsymbol{x} = [\delta\boldsymbol{\epsilon}, \boldsymbol{b}^b_{\text{gyro}}, \boldsymbol{p}^n_{b/n}, \boldsymbol{v}^n_{b/n}, \boldsymbol{b}^b_{\text{acc}}]$ and

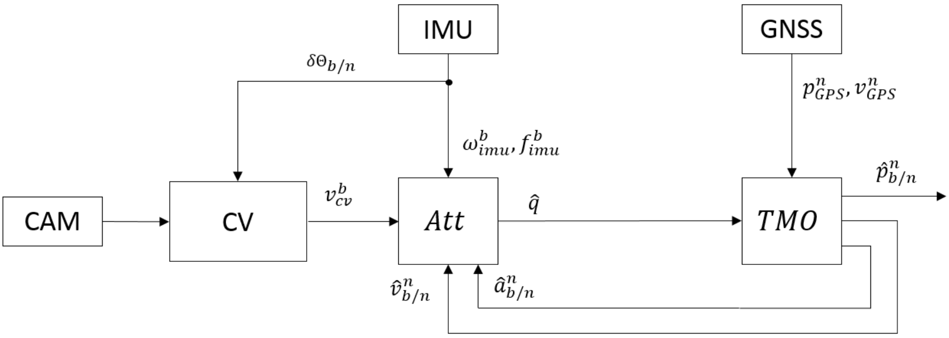


Figure 4.2: Block diagram of the position, velocity and attitude MEKF. Att represents the attitude observer, and TMO translational motion observer.

the input is $u = [\boldsymbol{\omega}^b_{\text{imu}}, \boldsymbol{f}^b_{\text{imu}}]$, yielding the following state-space system:

$$\underbrace{\begin{bmatrix} \delta\dot{\boldsymbol{\epsilon}} \\ \dot{\boldsymbol{b}}^b_{\text{gyro}} \\ \dot{\boldsymbol{p}}^n_{b/n} \\ \dot{\boldsymbol{v}}^n_{b/n} \\ \dot{\boldsymbol{b}}^b_{\text{acc}} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \frac{1}{2}\left(\sqrt{1-\delta\boldsymbol{\epsilon}^T\delta\boldsymbol{\epsilon}}I_{3\times3} + [\delta\boldsymbol{\epsilon}]_\times\right)\left(\boldsymbol{\omega}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{gyro}}\right) \\ 0_{3\times1} \\ \boldsymbol{v}^n_{b/n} \\ R^n_{\hat{b}}(\hat{q})R^{\hat{b}}_b(\delta q)\left[\boldsymbol{f}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{acc}}\right] + \boldsymbol{g}^n \\ 0_{3\times1} \end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})} + I_{15\times15}\underbrace{\begin{bmatrix} \boldsymbol{w}_{\text{gyro}} \\ \boldsymbol{w}_{\text{bgyro}} \\ 0_{3\times1} \\ \boldsymbol{w}_{\text{acc}} \\ \boldsymbol{w}_{\text{bacc}} \end{bmatrix}}_{\boldsymbol{w}} \tag{4.30}$$

The measurements used in the position, velocity and attitude filter are the measured specific force acting on the body, the body-fixed linear velocity from the CV, position from GPS, and velocity from GPS. As the lever arm of the GPS antenna is very small, the lever arm compensation is neglected, and it is assumed that $\boldsymbol{v}^n_{\text{GPS}} = \boldsymbol{v}^n_{b/n} + \boldsymbol{w}_{\text{vel}}$:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{f}^b_{\text{imu}} \\ \boldsymbol{v}^b_{\text{cv}} \\ \boldsymbol{p}^n_{\text{GPS}} \\ \boldsymbol{v}^n_{\text{GPS}} \end{bmatrix} = \underbrace{\begin{bmatrix} R^{\hat{b}}_b(\delta q)^T R^n_{\hat{b}}(\hat{q})^T\left[\dot{\boldsymbol{v}}^n_{b/n} - \boldsymbol{g}^n\right] + \boldsymbol{b}^b_{\text{acc}} \\ R^{\hat{b}}_b(\delta q)^T R^n_{\hat{b}}(\hat{q})^T \boldsymbol{v}^n_{b/n} \\ \boldsymbol{p}^n_{b/n} \\ \boldsymbol{v}^n_{b/n} \end{bmatrix}}_{\boldsymbol{h}(x)} + \underbrace{\begin{bmatrix} \boldsymbol{w}_{\text{acc}} \\ \boldsymbol{w}_{\text{cv}} \\ \boldsymbol{w}_{\text{pos}} \\ \boldsymbol{w}_{\text{vel}} \end{bmatrix}}_{\boldsymbol{v}} \tag{4.31}$$

The measurement noise is assumed white with variance $\Sigma_v \in \mathcal{R}^{12\times12}$

$$\boldsymbol{v} \sim \mathcal{N}(0_{12\times1}, \Sigma_v)$$

Alternatively, when dealing with normalized velocity from CV, the measurement equation takes on the form:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{f}^b_{\text{imu}} \\ \boldsymbol{v}^b_{\text{cv}} \\ \boldsymbol{p}^n_{\text{GPS}} \\ \boldsymbol{v}^n_{\text{GPS}} \end{bmatrix} = \underbrace{\begin{bmatrix} R^{\hat{b}}_b(\delta q)^T R^n_{\hat{b}}(\hat{q})^T \left[\dot{\boldsymbol{v}}^n_{b/n} - \boldsymbol{g}^n\right] + \boldsymbol{b}^b_{\text{acc}} \\ R^{\hat{b}}_b(\delta q)^T R^n_{\hat{b}}(\hat{q})^T \frac{\boldsymbol{v}^n_{b/n}}{\|\boldsymbol{v}^n_{b/n}\|} \\ \boldsymbol{p}^n_{b/n} \\ \boldsymbol{v}^n_{b/n} \end{bmatrix}}_{\boldsymbol{h}(x)} + \underbrace{\begin{bmatrix} \boldsymbol{w}_{\text{acc}} \\ \boldsymbol{w}_{\text{cv}} \\ \boldsymbol{w}_{\text{pos}} \\ \boldsymbol{w}_{\text{vel}} \end{bmatrix}}_{\boldsymbol{v}} \tag{4.32}$$

4.3.1 Discrete time equations

Discretizing equation (4.30) -(4.32) with the Euler method yields:

$$\boldsymbol{x}_{k+1} = \underbrace{\boldsymbol{x}_k + h\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)}_{\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)} + \underbrace{h\Gamma}_{\Gamma_k} \boldsymbol{w}_k \tag{4.33}$$

$$\boldsymbol{y}_k = \underbrace{\boldsymbol{h}(\boldsymbol{x}_k)}_{\boldsymbol{h}_k} + \boldsymbol{v}_k \tag{4.34}$$

Now that a discrete model is present, it is desirable to linearize this model, so that it will fit in the MEKF algorithm in Section 4.4.

$$Q_k = \Sigma_{wk} \tag{4.35}$$

$$R_k = \Sigma_{vk} \tag{4.36}$$

$$\boldsymbol{f}_k(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) = \hat{\boldsymbol{x}}_k + h \begin{bmatrix} \frac{1}{2}\left[\boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{gyro}}\right] \\ 0_{3\times1} \\ \hat{\boldsymbol{v}}^n_{b/n} \\ R^n_b(\hat{q})\left[\boldsymbol{f}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{acc}}\right] + \boldsymbol{g}^n \\ 0_{3\times1} \end{bmatrix} \tag{4.37}$$

$$\Phi_k = \left.\frac{\delta \boldsymbol{f}_k}{\delta x_k}\right|_{x_k=\hat{x}_k} = \tag{4.38}$$

$$I_{15\times15} + h \begin{bmatrix} -\frac{1}{2}\left[\boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{gyro}}\right]_\times & -\frac{1}{2}I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ -2R^n_b(\hat{q})\left[\boldsymbol{f}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{acc}}\right]_\times & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -R^n_{\hat{b}}(\hat{q}) \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{4.39}$$

$$\Gamma_k = hI_{15\times15} \tag{4.40}$$

$$H_k = \left.\frac{\delta \boldsymbol{h}_k}{\delta x_k}\right|_{x_k=\bar{x}_k} = \begin{bmatrix} -W(\delta\bar{\boldsymbol{\epsilon}}_k, R^n_{\hat{b}}(\hat{q})^T\boldsymbol{g}^n) & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ W(\delta\bar{\boldsymbol{\epsilon}}_k, R^n_{\hat{b}}(\hat{q})^T\boldsymbol{v}^n_{b/n}) & 0_{3\times3} & 0_{3\times3} & R^{\hat{b}}_b(\delta\bar{q})^T R^n_{\hat{b}}(\hat{q})^T & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{4.41}$$

With $W(\boldsymbol{\epsilon}, \boldsymbol{v})$ defined as in equation (4.28)

Alternatively when dealing with normalized body-fixed velocities from CV, equation (4.32) applies for the measurement. Then equation (4.41) is replaced by equation (4.42):

$$H_k = \left.\frac{\delta \boldsymbol{h}_k}{\delta x_k}\right|_{x_k=\bar{x}_k} = \begin{bmatrix} -W\left(\delta\bar{\boldsymbol{\epsilon}}_k, R^n_{\hat{b}}(\hat{q})^T\boldsymbol{g}^n\right) & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ W\left(\delta\bar{\boldsymbol{\epsilon}}_k, R^n_{\hat{b}}(\hat{q})^T\frac{\boldsymbol{v}^n_{b/n}}{\|\boldsymbol{v}^n_{b/n}\|}\right) & 0_{3\times3} & 0_{3\times3} & R^{\hat{b}}_b(\delta\bar{q})^T R^n_{\hat{b}}(\hat{q})^T Z(\boldsymbol{v}^n_{b/n}) & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{4.42}$$

$$Z(\boldsymbol{v}) = \frac{1}{\|\boldsymbol{v}\|^3}\begin{bmatrix} v_2^2 + v_3^2 & -v_1v_2 & -v_1v_3 \\ -v_1v_2 & v_1^2 + v_3^2 & -v_2v_3 \\ -v_1v_3 & -v_2v_3 & v_1^2 + v_2^2 \end{bmatrix} = \frac{1}{\|\boldsymbol{v}\|^3}\left(\text{diag}(\boldsymbol{v})^2 - \boldsymbol{v}\boldsymbol{v}^T\right) \tag{4.43}$$

As seen from equation (4.43), $Z(\boldsymbol{v}^n_{b/n})$ will be very small due to the scaling by $\frac{1}{\|\boldsymbol{v}^n_{b/n}\|^3}$, hence it can be argued that CV does not affect the velocity estimate when measuring the normalized body-fixed velocity, and the term $R_b^{\hat{b}}(\delta\bar{q})^T R_{\hat{b}}^n(\hat{q})^T Z(\boldsymbol{v}_{b/n}^n)$ in equation (4.42) might be set to zero.

4.4 The MEKF Algorithm

Now that discrete time equations are present, the iteration loop of the MEKF algorithm is stated. The MEKF is initialized with an initial prior estimate $\bar{\boldsymbol{x}}_0$. The initial error covariance matrix, $\bar{P}_0 = E((\boldsymbol{x}(0) - \bar{\boldsymbol{x}}_0)(\boldsymbol{x}(0) - \bar{\boldsymbol{x}}_0)^T)$, is given values based on the uncertainty in the initial estimates.

$$
\begin{aligned}
&(1.)K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1} \\
&(2.)\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k + K_k(\boldsymbol{y}_k - \boldsymbol{h}(\bar{x}_k)) \\
&(3.)\hat{q}_k = \begin{bmatrix} \sqrt{1 - \delta\hat{\boldsymbol{\epsilon}_k}^T \delta\hat{\boldsymbol{\epsilon}}_k} \\ \delta\hat{\boldsymbol{\epsilon}_k} \end{bmatrix} \\
&(4.)\delta\hat{\boldsymbol{\epsilon}}_{k+1} = 0_{3\times 1} \\
&(5.)\hat{P}_k = (I - K_k H_k)\bar{P}_k (I - K_k H_k)^T + K_k R_k K_k^T \\
&(6.)\hat{P}_k = \frac{1}{2}(\hat{P}_k + \hat{P}_k^T) \\
&(7.)\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{f}_k(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) \\
&(8.)\bar{P}_{k+1} = \Phi_k \hat{P}_k \Phi_k^T + \Gamma_k Q_k \Gamma_k^T
\end{aligned}
$$

With $Q_k, R_k, \boldsymbol{f}_k, \Phi_k, \Gamma_k$ being evaluated at every step k of the MEKF, and defined as in equation (4.35)-(4.40). H_k being defined in equation (4.42). In this thesis we always normalize the velocity from CV, before using it in the MEKF. As seen from equation (4.43), $Z(\boldsymbol{v}_{b/n}^n)$ will tend to zero when the CV measurement is normalized, and is therefore set to zero. $\boldsymbol{h}(x)$ is defined in equation (4.32). $\dot{\boldsymbol{v}}_{b/n}^n$ in equation (4.32) is obtained from the continuous expression of the derivative of the velocity from equation (4.30) and found by $\dot{\boldsymbol{v}}_{b/n}^n = R_b^n(\hat{q}_k)(\boldsymbol{f}_{\text{imu},k}^b - \hat{\boldsymbol{b}}_{\text{acc},k}^b) + \boldsymbol{g}^n$. $\bar{P}_k$ and $\hat{P}_k$ being the prior and estimated error covariance matrix.

Chapter 5

Nonlinear Observer for UAV Navigation

This chapter presents different vision-aided nonlinear observers (NO) (Mahony et al., 2008; Hua, 2010; Grip et al., 2012*a*; Fusini et al., 2014, 2015) for estimating the attitude, position, velocity and gyroscopic bias of a fixed-wing UAV.

The observer from Fusini et al. (2014), utilizing ground truth optical flow (GTOF from Section 3.3.1), is referred to as Ground Truth Optical Flow Nonlinear Observer (GTOF NO) throughout this thesis. A requirement of the GTOF NO is that the terrain being filmed must be flat horizontal. This is a fundamental restriction, delimiting the various environments where the observer is applicable. When flying at high altitudes the assumption of flat terrain might be viable, but when flying at lower altitudes, the presence of rugged terrain will cause the GTOF NO representation to be highly inaccurate, as CV fails to output correct measurements. As a consequence of this, a new CV subsystem is needed. Two new observer representations are therefore proposed. The first representation enables the continuous epipolar constraint as described in Section 3.3.3, and is named Continuous Epipolar Optical Flow Nonlinear Observer (CEOF NO). The second uses the epipolar constraint from Section 3.3.2, and is named Epipolar Optical Flow Nonlinear Observer (EOF NO). With epipolar geometry it is possible to calculate a scaled measure of the body-fixed linear velocity without any prior information about the structure being filmed (e.g. terrain is a planar surface, known attitude of UAV relative to the terrain plane). The CEOF and EOF NO representations are one step further to a more general and practical observer representation. The CEOF and EOF NO representations will be presented together with assumptions and motivation. Moreover theorems and proofs show that the proposed observers have the same stability properties as the GTOF NO, namely a Uniformly Semi-Globally Exponentially Stable (USGES) origin of the error dynamics.

In this chapter the GTOF NO from Fusini et al. (2014) will be stated, together with the assumptions behind. Furthermore the proposed CEOF and EOF NO will be presented. The motivation, assumptions, advantages and drawbacks with the EOF and CEOF nonlinear observers will be listed, together with the equations describing the observer and its stability properties. The chapter ends with a comparison between the GTOF, EOF and the CEOF observers, listing pros and cons with the different representations. The following observer representations are designed and presented

- Previous work 1 (Section 5.4): Ground truth optical flow nonlinear observer with inclinometer and altimeter measurements (Fusini et al., 2014, 2015), abbreviation GTOF NO. Computer vision uses the ground truth optical flow (GTOF) relationship from Section 3.3.1 to calculate body-fixed linear velocity.

- Previous work 2 (Section 5.4): Ground truth optical flow nonlinear observer with attitude and altitude feedback (Fusini et al., 2014, 2015), abbreviation GTOF(f) NO, where "(f)" indicates that computer vision relies on feedback of the estimated attitude. No stability analysis exists for this observer, but an experimental validation follows in Chapter 8.

- Proposed system 1 (Section 5.5): Continuous epipolar optical flow nonlinear observer, abbreviation CEOF NO. Computer vision uses the gyro-aided continuous epipolar constraint from Section 3.3.3 to calculate the normalized body-fixed linear velocity.

- Proposed system 2 (Section 5.6): Epipolar optical flow nonlinear observer, abbreviation EOF NO. Computer vision uses the gyro-aided epipolar constraint from Section 3.3.2 to calculate an approximation of the normalized body-fixed linear velocity.

This chapter starts with presenting the kinematics of the estimated states in Section 5.1. The different observer representations differ in the CV subsystem, therefore the overall observer assumptions and equations are stated in Section 5.2 and Section 5.3 respectively. Then, in Section 5.4-5.6, the different observer representations are presented with their specific CV subsystem. The observers are stated together with the necessary assumptions and stability properties. The chapter ends with a comparison of the different observer representations in Section 5.7.

5.1 Kinematics

The nonlinear observer expresses attitude in means by a rotation matrix. The differential equation describing the dynamics of attitude, position, and velocity is

$$\dot{R}_b^n = R_b^n \left[\boldsymbol{\omega}_{b/n}^b\right]_\times \tag{5.1a}$$

$$\dot{\boldsymbol{p}}_{b/n}^n = \boldsymbol{v}_{b/n}^n \tag{5.1b}$$

$$\dot{\boldsymbol{v}}_{b/n}^n = \boldsymbol{f}_{b/n}^n + \boldsymbol{g}^n \tag{5.1c}$$

With the observer one are able to estimate the attitude R_b^n, the position $\boldsymbol{p}_{b/n}^n$, and the velocity $\boldsymbol{v}_{b/n}^n$ with exponential convergence rate (Fusini et al., 2014). An estimator for the gyro bias $\boldsymbol{b}_{\text{gyro}}^b$ is also provided.

5.2 Assumptions

The observer design by Fusini et al. (2014) is based on the following assumptions:

Assumption 5.1. *The gyro bias $\boldsymbol{b}_{gyro}^b$ is constant, and there exists a known constant $L_b > 0$ such that $\|b_{\text{gyro}}^b\| \leq L_b$.*

Assumption 5.2. *There exists a constant $c_{obs} > 0$ such that, $\forall t \geq 0$, $\|\boldsymbol{v}_{\text{cv}}^b \times \boldsymbol{f}_{\text{imu}}^b\| \geq c_{obs}$.*

Assumption 5.2 states that the UAV cannot have a specific force in the same direction as the velocity of the UAV. This implies that the UAV cannot fly straight up, nor straight down, as this would align the velocity vector with the gravity vector. This means that the nonlinear observer should not be used in a helicopter-like vehicle, as this would cause a problem in hover or when purely gaining altitude. This is however not a problem for fixed wing UAVs as they will always have some forward speed to remain in the air.

5.3 Observer Equations

The full observer was introduced in Fusini et al. (2014), and all estimated states are expressed w.r.t. {n} . The observer uses measurements of the linear velocity and the specific force as body-fixed reference vectors. The body-fixed linear velocity is measured by some CV subsystem, while the body-fixed specific force is measured by an accelerometer. The estimated linear velocity in {n} and the estimated specific force in {n} are used as {n} reference vectors. The equations of the observer are stated as

$$\Sigma_1 \begin{cases} \dot{\hat{R}}_b^n = \hat{R}_b^n S(\omega_{\text{imu}}^b - \hat{\boldsymbol{b}}_{\text{gyro}}^b) + \sigma K_P \hat{J} \\ \dot{\hat{\boldsymbol{b}}}_{\text{gyro}}^b = \text{Proj}(\hat{\boldsymbol{b}}_{\text{gyro}}^b, -k_I \text{vex}(\mathbb{P}_a(\hat{R}_s^T K_P \hat{J}))) \end{cases} \tag{5.2}$$

$$\Sigma_2 \begin{cases} \dot{\hat{\boldsymbol{p}}}_{b/n}^n = \hat{\boldsymbol{v}}_{b/n}^n + K_{pp}(\boldsymbol{p}_{\text{GPS}}^n - \hat{\boldsymbol{p}}_{b/n}^n) + K_{pv}(\boldsymbol{v}_{\text{GPS}}^n - \hat{\boldsymbol{v}}_{b/n}^n) \\ \dot{\hat{\boldsymbol{v}}}_{b/n}^n = \hat{\boldsymbol{f}}_{b/n}^n + \boldsymbol{g}^n + K_{vp}(\boldsymbol{p}_{\text{GPS}}^n - \hat{\boldsymbol{p}}_{b/n}^n) + K_{vv}(\boldsymbol{v}_{\text{GPS}}^n - \hat{\boldsymbol{v}}_{b/n}^n) \\ \dot{\xi} = -\sigma K_P \hat{J} \boldsymbol{f}_{\text{imu}}^b + K_{\xi p}(\boldsymbol{p}_{\text{GPS}}^n - \hat{\boldsymbol{p}}_{b/n}^n) + K_{\xi v}(\boldsymbol{v}_{\text{GPS}}^n - \hat{\boldsymbol{v}}_{b/n}^n) \\ \hat{\boldsymbol{f}}_{b/n}^n = \hat{R}_b^n \boldsymbol{f}_{\text{imu}}^b + \xi \end{cases} \tag{5.3}$$

$$\text{CV} \begin{cases} \text{Computer vision measures } \boldsymbol{v}_{\text{cv}}^b \text{ and will be described} \\ \text{in Section 5.4, Section 5.5 and Section 5.6} \end{cases}$$

The CV subsystem will be defined in Section 5.4, Section 5.5 and Section 5.6 for the GTOF, CEOF and EOF observer representations respectively.

$\hat{J} = \hat{A}_n A_b^T - \hat{R} A_b A_b^T$ is the output injection term. A_b and $\hat{A}_n$ are matrices made up from combinations of reference vectors. A reference vector is a vector that is known (inertial, measured or estimated) in different frames. The reference vectors used in this thesis are linear velocity $\boldsymbol{v}_{b/n}^\tau$ and specific force $\boldsymbol{f}_{b/n}^\tau$, $\tau = \{n, b\}$. In this thesis normalized reference vectors are used when constructing A_b and $\hat{A}_n$. Then $\hat{A}_n$ and A_b are orthogonal matrices, and must be constructed to satisfy $\hat{A}_n A_b = R_b^n$ (Grip et al., 2015). These matrices can be constructed in several ways as long as they are continuous in time and uniformly bounded by some upper bound $\|A_b\| = \|\hat{A}_n\| \leq \|M_A\|$. In this thesis $\hat{J}$ is defined as

$$\begin{aligned} &\hat{J}(\boldsymbol{v}_{\text{cv}}^b,\ \hat{\boldsymbol{v}}_{b/n}^n, \boldsymbol{f}_{\text{imu}}^b, \hat{\boldsymbol{f}}_{b/n}^n, \hat{R}) := \hat{A}_n A_b^T - \hat{R} A_b A_b^T \\ &A_b := \left[\frac{\boldsymbol{f}_{\text{imu}}^b}{\|\boldsymbol{f}_{\text{imu}}^b\|}, \frac{\boldsymbol{f}_{\text{imu}}^b}{\|\boldsymbol{f}_{\text{imu}}^b\|} \times \frac{\boldsymbol{v}_{\text{cv}}^b}{\|\boldsymbol{v}_{\text{cv}}^b\|}, \frac{\boldsymbol{f}_{\text{imu}}^b}{\|\boldsymbol{f}_{\text{imu}}^b\|} \times \left(\frac{\boldsymbol{f}_{\text{imu}}^b}{\|\boldsymbol{f}_{\text{imu}}^b\|} \times \frac{\boldsymbol{v}_{\text{cv}}^b}{\|\boldsymbol{v}_{\text{cv}}^b\|} \right) \right] \\ &\hat{A}_n := \left[\frac{\hat{\boldsymbol{f}}_{b/n}^n}{\|\hat{\boldsymbol{f}}_{b/n}^n\|}, \frac{\hat{\boldsymbol{f}}_{b/n}^n}{\|\hat{\boldsymbol{f}}_{b/n}^n\|} \times \frac{\hat{\boldsymbol{v}}_{b/n}^n}{\|\hat{\boldsymbol{v}}_{b/n}^n\|}, \frac{\hat{\boldsymbol{f}}_{b/n}^n}{\|\hat{\boldsymbol{f}}_{b/n}^n\|} \times \left(\frac{\hat{\boldsymbol{f}}_{b/n}^n}{\|\hat{\boldsymbol{f}}_{b/n}^n\|} \times \frac{\hat{\boldsymbol{v}}_{b/n}^n}{\|\hat{\boldsymbol{v}}_{b/n}^n\|} \right) \right] \end{aligned} \tag{5.4}$$

Note that the injection term $\hat{J}$ is slightly different from what was used in Fusini et al. (2014, 2015). The measurement of the specific force $\boldsymbol{f}_{\text{imu}}^b$ from the accelerometer is assumed more accurate than the body-fixed velocity measurement from CV, and is therefore placed alone in the first column of A^b and $\hat{A}^n$, compared to Fusini et al. (2014, 2015) where $\boldsymbol{v}_{\text{cv}}^b$ was placed in the first column. The specific force reference vector is then weighted the most in the injection term.

The parameter projection $\mathrm{Proj}(\cdot,\cdot)$ is defined as:

$$\mathrm{Proj}(\hat{b}^b,\tau)=\begin{cases}\left(I-\frac{c(\hat{b}^b)}{\|\hat{b}^b\|^2}\hat{b}^b\hat{b}^{bT}\right)\tau, & \|\hat{b}^b\|\geq L_b,\ \hat{b}^{bT}\tau>0\\ \tau, & \text{otherwise}\end{cases}$$

where $c(\hat{b}^b) = \min\{1,(\|\hat{b}^b\|^2 - L_b^2)/(L_{\hat{b}}^2 - L_b^2)\}$. $\mathrm{Proj}(\cdot,\cdot)$ represents a parameter projection Krstic et al. (1995) that ensures that $\|\hat{\boldsymbol{b}}^b_{\text{gyro}}\|$ does not exceed a design constant $L_{\hat{b}} > L_b$. This operator is a special case of that from Appendix E of Krstic et al. (1995).

The subsystem Σ_1 represents the attitude observer, whereas Σ_2 represents the translational motion observer. $\sigma \geq 1$ is a scaling factor tuned to achieve stability (Fusini et al., 2014), k_I is a positive scalar gain, K_P is a symmetric positive definite gain matrix, and $\hat{R}_s = \mathrm{sat}(\hat{R}_b^n)$, ensuring that all elements of R_s are between -1 and 1. $K_{pp}, K_{pv}, K_{vp}, K_{vv}, K_{\xi p}$, and $K_{\xi v}$ are observers gains with bounds given by Lyapunov stability (Fusini et al., 2014).

5.4 Ground Truth Optical Flow Observer Representation

In this section the observer from Fusini et al. (2014) will be presented together with its CV subsystem. It was shown by Fusini et al. (2014) that the observer is USGES in the origin of the error dynamics, given that the CV outputs correct measurements of the body-fixed linear velocity. In this thesis this observer is addressed as a Ground Truth Optical Flow (GTOF) observer representation.

There are two versions of the GTOF observer. The first version, which is shown in Figure 5.1, assumes that the roll and pitch angles of the UAV can be measured by an inclinometer, and that the height above the terrain can be extracted from the GNSS or from an altimeter. By assuming this, the CV subsystem does only rely on known quantities and the body-fixed velocity can be calculated from the OF. For this version the system has USGES stability of the origin of the error dynamics (Fusini et al., 2014), and the performance of the observer was presented as an experimental validation in the ICUAS'15 article Fusini et al. (2015).

The second version, shown in Figure 5.2, uses feedback of the attitude and height from the observer, hence introducing another feedback loop in the observer structure. However, using the estimated position as a measure of the height requires the terrain to have constant or known elevation. The stability of the observer when introducing this feedback has not yet been assessed, but an experimental validation of this structure follows in Chapter 8. In this section the observer structure from Figure 5.1, with measured attitude and height, will be emphasized.

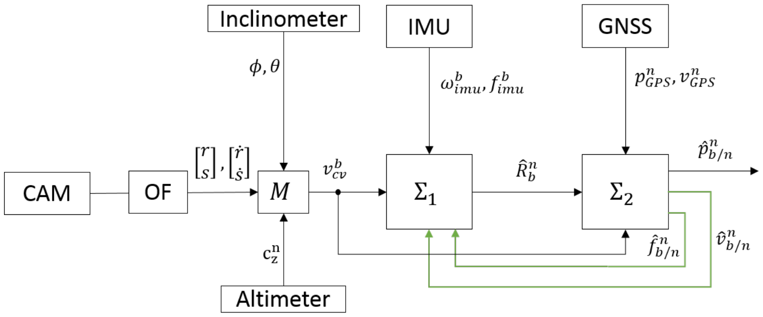


Figure 5.1: Ground truth optical flow representation of the observer with attitude measurements. OF represents the algorithm that calculates optical flow from the camera images. Σ_1 represents the attitude observer, and Σ_2 the translational motion observer. The feedback indicated in green has been proved to have USGES stability properties, assuming measured Euler angles and measured altitude.

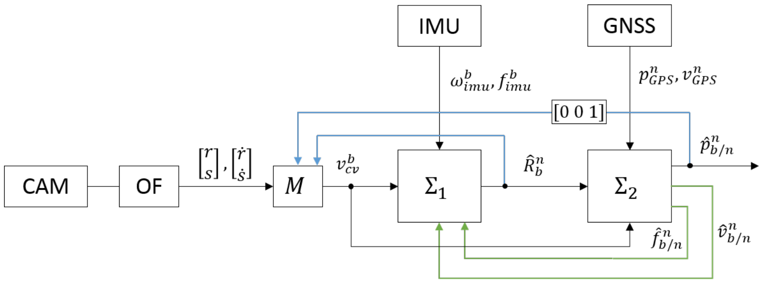


Figure 5.2: Ground truth optical flow representation of the observer with attitude and altitude feedback. OF represents the algorithm that calculates optical flow from the camera images. Σ_1 represents the attitude observer, and Σ_2 the translational motion observer. The feedback indicated in green has been proved to have USGES stability properties, assuming measured Euler angles and measured altitude. For the blue feedback an experimental validation will follow in Chapter 8.

5.4.1 Motivation and Advantages

The advantages with using the GTOF CV subsystem in the observer is

- The scale of the body-fixed linear velocity is preserved
- Both linear and angular velocity are measured

The main advantage with the GTOF observer is that it measures the body-fixed linear velocity with preservation of the scale. The GTOF CV subsystem can therefore give some information about the velocity that in theory can be used in the translational motion observer Σ_2. The velocity from the GTOF CV subsystem could for instance be used in dead reckoning. In the attitude observer, Σ_1, one are only interested in using the normalized reference vector in order to avoid scaling issues. Moreover the GTOF actually measures 6DOF velocities, meaning it also measures the angular rates of the UAV. This could in theory be used in addition to the gyroscope, but the quality of the angular velocity measurement is in general poor compared to the gyroscopic measurements.

5.4.2 Disadvantages

A big disadvantage with the GTOF observer comes from the requirement that the distance to all points in the image needs to be known. If the distance to a point is not known, then it is impossible to say something about the theoretical flow, hence it is impossible to recover the motion of the UAV.

The fact that the GTOF observer needs to know the distance to all points in the image causes problems. In Fusini et al. (2014, 2015) the distance to every point in the image was approximated by assuming that the terrain is a horizontal flat plane. By knowing the altitude of the UAV above ground, and the attitude, it is possible to recover the distance to every point of the horizontal terrain plane. If, however, the terrain is not horizontal, then the assumed distance will be incorrect. This leads to false information being fed into the CV subsystem. This will cause CV to output erroneous measurements, as the relationship between ground truth optical flow and motion is no longer valid.

The GTOF CV subsystem assumes that the OF algorithms measures continuous OF. As the OF algorithms are discrete, they only measure the discrete OF. This implies that the continuous OF, $\dot{\boldsymbol{x}}^m$ must be approximated by the discrete OF, $\dot{\boldsymbol{x}}^m \approx \frac{\Delta \boldsymbol{x}^m}{\Delta t}$. This will cause no problems if the frame rate of the camera is sufficiently high as $\dot{\boldsymbol{x}}^m = \lim_{\Delta t \to 0} \frac{\Delta \boldsymbol{x}^m}{\Delta t}$

5.4.3 Assumptions

The observer design by Fusini et al. (2014) is based on the assumptions 5.1-5.2. In adition to these assumptions, three more assumptions must be added when using the GTOF CV subsystem.

Assumption 5.3. *The OF algorithm provides a sufficient number of image features, such that M has full rank and equation* (3.10) *can be used to find* $\boldsymbol{v}^b_{\text{cv}} = \boldsymbol{v}^b_{b/n}$.

Assumption 5.4. *Euler angles are considered measured by inclinometer and not extracted from the estimated* $\hat{R}^n_b$ *(i.e.* $\hat{R}^n_b$ *is not fed back to* M*) as well as the altitude* c^n_z*, so that matrix* M *and subsequently* $\boldsymbol{v}^b_{\text{cv}}$ *depend only on known values.*

Assumption 5.5. *The UAV flies over flat horizontal terrain.*

5.4.4 Observer Representation

Given the aforementioned assumptions, the full observer is stated as equation (5.2)-(5.3) with the following computer vision subsystem:

$$\text{CV}\left\{\begin{bmatrix} \boldsymbol{v}^b_{\text{cv}} \\ \boldsymbol{\omega}^b_{\text{cv}} \end{bmatrix} = -M^{+}\begin{bmatrix} \dot{r} \\ \dot{s} \end{bmatrix}\right. \tag{5.5}$$

With M being the matrix defined in equation (3.11) in Section 3.3.1.

5.4.5 Stability Proof

The error dynamics of the nonlinear observer can be written in a compact form as

$$\Sigma_1\left\{\begin{aligned} \dot{\hat{R}}^n_b &= R^n_b\left[\omega^b_{b/n}\right]_\times - \hat{R}^n_b\left[\omega^b_{\text{imu}} - \hat{b}^b_{\text{gyro}}\right]_\times - \sigma K_P \hat{J} \\ \dot{\hat{b}}^b_{\text{gyro}} &= -\text{Proj}(\hat{b}^b_{\text{gyro}}, -k_I \text{vex}(\mathbb{P}_a(\hat{R}^T_s K_P \hat{J}))) \end{aligned}\right. \tag{5.6a}$$

$$\Sigma_2\left\{\dot{\tilde{w}} = (A_w - K_w C_w)\tilde{w} + B_w \tilde{d}\right. \tag{5.6b}$$

where $\tilde{w} = [(\tilde{p}^n_{b/n})^T, (\tilde{v}^n_{b/n})^T, (\tilde{f}^n_{b/n})^T]^T$ collects the estimated position, velocity and acceleration vectors, $\tilde{d} = \left(R^n_b\left[\omega^b_{b/n}\right]_\times - \hat{R}^n_b\left[\omega^b_{\text{imu}} - \hat{b}^b_{\text{gyro}}\right]_\times\right) f^b_{b/n} + \left(R^n_b - \hat{R}^n_b\right)\dot{f}^b_{b/n}$, and the four matrices in (5.6b) are defined as

$$A_w = \begin{bmatrix} 0_{6\times 3} & I_6 \\ 0_{3\times 3} & 0_{3\times 6} \end{bmatrix}, \qquad B_w = \begin{bmatrix} 0_{6\times 3} \\ I_3 \end{bmatrix},$$

$$C_w = \begin{bmatrix} I_6 & 0_{6\times 3} \end{bmatrix}, \qquad K_w = \begin{bmatrix} K_{pp} & K_{pv} \\ K_{vp} & K_{vv} \\ K_{\xi p} & K_{\xi v} \end{bmatrix}.$$

Theorem 1 is adapted from Fusini et al. (2014) and describes the stability properties of the observer.

Theorem 1. ***(Stability of the GTOF observer)*** *Let σ be chosen to ensure stability according to Lemma 1 in Grip et al. (2012*a*) and define $H_K(s) = (Is - A_w + K_w C_w)^{-1} B_w$. Provided Assumptions 5.1-5.5, there exists a set (0,c) such that, if K_w is chosen such that $A_w - K_w C_w$ is Hurwitz, and $\|H_K(s)\|_\infty < \gamma$, for $\gamma \in (0, c)$, then the origin of the error dynamics equation* (5.2)-(5.3) *and equation* (5.5) *is USGES when the initial conditions satisfy $\|\hat{b}^b_{\text{gyro}}(0)\| \leq L_{\hat{b}}$.*

Proof. See Fusini et al. (2014). □

5.5 Continuous Epipolar Optical Flow Observer Representation

In this section a new observer representation, named Continuous Epipolar Optical Flow (CEOF) Observer Representation, is presented. The observer has a robust CV subsystem that enables gyro-aided epipolar geometry as described in Section 3.3.3. The CV subsystem does not need any prior information about the structure being filmed (e.g. flat terrain, known roll, pitch and height between UAV and terrain), as it is not affected by the distance and shape of the terrain being filmed. The observer structure is shown in Figure 5.3. The CEOF NO requires angular velocity as input to the CV subsystem. Bias compensated angular velocity measurements are acquired from the gyroscope. Before describing the CEOF NO representation, some motivation, disadvantages and assumptions are stated.

5.5.1 Motivation and Advantages

The motivation and advantages for using the CEOF Observer Representation will now be listed.

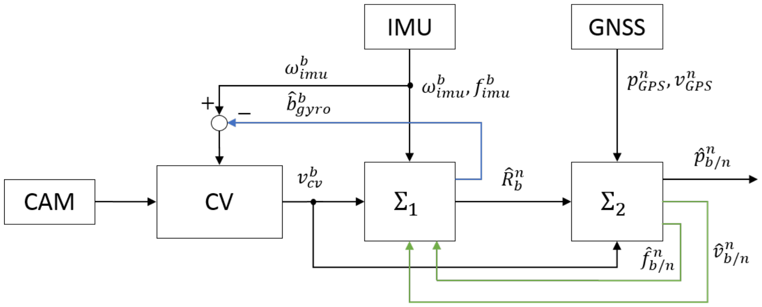


Figure 5.3: Block diagram of the Continuous Epipolar Optical Flow Observer Representation. Σ_1 represents the attitude observer, and Σ_2 the translational motion observer. For the feedback indicated with green, USGES stability has been proved. The stability of the bias feedback in blue has not yet been assessed.

- The CEOF Observer Representation does not assume flat terrain (unlike the GTOF Observer Representation), and can therefore be used at lower altitudes and in more hilly environments. This makes this observer representation applicable for more real world situations than the Ground Truth Optical Flow Observer Representation.
- No information about the distance down to the terrain is required.
- No external sensor measuring the attitude or altitude relative to the ground is required.

5.5.2 Disadvantages

Some disadvantages are also present with the CEOF observer.

- The CV subsystem of the CEOF Observer Representation outputs a dimensionless measure of the velocity, and does not provide any information about the scale of the velocity.
- The CEOF Observer Representation requires that the UAV has some forward velocity, but this is reasonable as a fixed wing UAV always will have some forward velocity when in flight.
- The CV subsystem assumes continuous OF, although the OF algorithm approximates this by the discrete OF.
- A feedback is introduced by the bias correction.

5.5.3 Assumptions

Some assumptions need to be stated for the CEOF nonlinear observer.

Assumption 5.6. *Angular rates are measured by the IMU meaning that $\boldsymbol{v}^b_{\text{cv}}$ is depending only on known values.*

Assumption 5.7. *The UAV has forward body-fixed velocity, $v_x > 0$.*

Assumption 5.8. *The OF algorithm provides linearly independent optical flow vectors, as defined in Definition 8.*

5.5.4 Observer Representation

Given N OF vectors, and provided Assumptions 5.1-5.2 and 5.6-5.8, the equations of the CEOF NO are stated as equation (5.2)-(5.3) together with the following CV subsystem.

$$\text{CV}\begin{cases} \boldsymbol{v}^b_{\text{cv}} = \frac{\boldsymbol{v}_e}{\|\boldsymbol{v}_e\|} \\ \boldsymbol{v}_e = \frac{\boldsymbol{v}^b_{b/n}}{v_x} = [1, (A_c^+ \boldsymbol{b}_c)^T]^T \\ \boldsymbol{u}^c_j \times \left((\dot{\boldsymbol{u}}^c_j)^T + \left[\boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{gyro}} \right]_\times \boldsymbol{u}^c_j \right) = [c_{x,j}, c_{y,j}, c_{y,j}]^T, j = 1...N \end{cases} \tag{5.7}$$

With A_c and $\boldsymbol{b}_c$ as defined in equation (3.55).

Where $\boldsymbol{u}^c = K^{-1}\underline{\boldsymbol{x}}^m$ is the back projected image coordinate, and $\dot{\boldsymbol{u}}^c = K^{-1}\dot{\underline{\boldsymbol{x}}}^m$ is the back projected OF. K is the projection matrix as defined in equation (3.4).

5.5.5 Stability Proof

Provided Assumptions 5.1-5.2 and 5.6-5.8, the following theorem can be stated to describe the stability of the CEOF observer, if assuming that $\hat{b}^b_{\text{gyro}}$ is kept constant in equation (5.7)

Theorem 2. ***(Stability of the CEOF observer)*** *Let σ be chosen to ensure stability according to Lemma 1 in Grip et al. (2012*a*) and define $H_K(s) = (Is - A_w + K_w C_w)^{-1} B_w$. There exists a set (0,c) such that, if K_w is chosen such that $A_w - K_w C_w$ is Hurwitz, and $\|H_K(s)\|_\infty < \gamma$, for $\gamma \in (0, c)$, then the origin of the error dynamics equation (5.2)-(5.3) and equation (5.7), provided Assumptions 5.1-5.2 and 5.6-5.8, is USGES* *when the initial conditions satisfy* $\|\hat{b}^b_{\text{gyro}}(0)\| \leq L_{\hat{b}}$.

Proof. The proof is based on Fusini et al. (2014), where we have replaced M with the new CV subsystem from equation (5.7). We must show that $\boldsymbol{v}^b_{\text{cv}}$ is uniquely defined. Then it follows from Theorem 1 that the origin of the error dynamics equation (5.2)-(5.3) together with equation (5.7) is USGES.

$\boldsymbol{v}^b_{\text{cv}}$ has a one to one mapping to the scaled body-fixed velocity with unit forward component $\boldsymbol{v}_e$. Hence the uniqueness of $\boldsymbol{v}^b_{\text{cv}}$ can be shown by the uniqueness of $\boldsymbol{v}_e$. Moreover if the sign of v_x is known, then $\boldsymbol{v}^b_{\text{cv}} = \text{normalize}(\boldsymbol{v}^b_{b/n})$. From Assumption 5.7 $v_x > 0$, hence the uniqueness of $\boldsymbol{v}^b_{\text{cv}}$ can be shown by the uniqueness of $\boldsymbol{v}_e$. $\boldsymbol{v}_e = [1, (A_c^+ \boldsymbol{b}_c)^T]^T$ has a unique solution if and only if the rank of A_c is full (Kreyszig, 2006). Given that the OF algorithm extracts features such that Assumption 5.8 is not violated, then A_c has full rank, and $\boldsymbol{v}_e$ is uniquely determined. Hence $\boldsymbol{v}^b_{\text{cv}}$ is uniquely determined, and it follows from Theorem 1 that the system is USGES. □

5.6 Epipolar Optical Flow Observer Representation

In this section the Epipolar Optical Flow (EOF) Observer Representation, will be presented. This observer representation is very similar to the CEOF observer from Section 5.5, but uses the discrete epipolar constraint from Section 3.3.2 to calculate the normalized body-fixed linear velocity.

The observer structure is shown in Figure 5.4. An IMU can be configured to output incremental angles ($\delta\dot{\Theta}_{b/n}$) as seen in Figure 5.4. The motivation behind the EOF observer is however that its CV subsystem is discrete in its nature, as it is derived assuming discrete OF. Recall that the OF algorithms measures the discrete OF. Before describing the EOF Observer Representation, some motivation, disadvantages and assumptions are stated.

5.6.1 Motivation and Advantages

The motivation and advantages for using the EOF Observer Representation can be summarized with the following list:

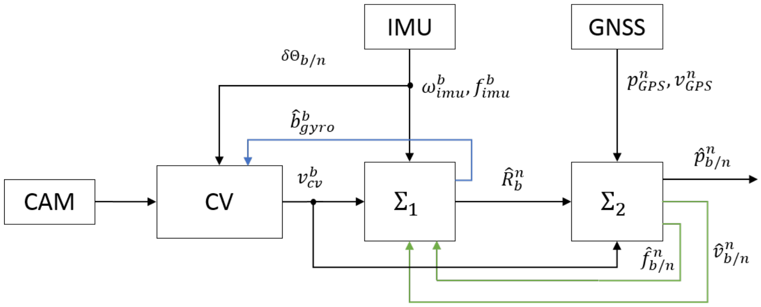


Figure 5.4: Block diagram of the Epipolar Optical Flow Observer Representation. Σ_1 represents the attitude observer, and Σ_2 the translational motion observer. For the feedback indicated with green, USGES stability has been proved.

- The EOF Representation does not assume flat terrain (unlike the GTOF Observer Representation), and can therefore be used at lower altitudes and in rugged environments. This makes this observer representation applicable for more real world situations than the GTOF Observer Representation.

- The EOF observer assumes discrete OF in the CV subsystem. The OF algorithms produces indeed measures of the discrete OF, and not the OF, meaning one should get an accurate measurement of the direction of translation of the UAV.

- No information about the distance down to the terrain is required.

5.6.2 Disadvantages

There are also some disadvantages the EOF observer, and these are now emphasized:

- The CV subsystem of the EOF Observer Representation outputs a dimensionless measure of the translation, and does not provide any information about the scale of the translation.

- The normalized body-fixed linear velocity is approximated by the normalized direction of translation. This is a good approximation if the time between two images are short, but when this time increases then the approximation might be rather poor.

- The CV subsystem of the EOF Observer Representation requires that the UAV has some forward translation between images are taken, but this is reasonable as a fixed wing UAV always will have some forward velocity when in flight.

5.6.3 Assumptions

Together with Assumptions 5.1-5.2, the following assumptions are stated for the EOF nonlinear observer:

Assumption 5.9. *The OF algorithm provides linearly independent pairs of image features, as defined in Definition 7.*

Assumption 5.10. *Incremental angles are measured by an IMU such that δR as defined in equation* (3.24) *is known, and consequently $\boldsymbol{v}^b_{\text{cv}}$ depend only on known values.*

Assumption 5.11. *The UAV has forward body-fixed velocity, $v_x > 0$, and the time between two consecutive images is sufficiently low, such that the body-fixed position-change will be positive in x-direction between two consecutive images, that is $t_x > 0$.*

Assumption 5.11 restrict the types of maneuvers that can be allowed. The UAV cannot rotate and translate such that the position of the UAV at time $t - \Delta t$ (when the last image was taken) appear behind the UAV at time t, as this would violate $t_x > 0$.

5.6.4 Observer Representation

Given Assumptions 5.1-5.2 together with Assumptions 5.9-5.11, the equations of the EOF NO representation are stated as equation (5.2)-(5.3) together with the following CV subsystem. Please notice that Σ_1 and Σ_2 remains unchanged from the GTOF and CEOF NO, as the CV subsystem is the only part that is changed.

$$\text{CV}\begin{cases}\boldsymbol{v}^b_{\text{cv}} = \text{sign}(t_x)\frac{\boldsymbol{t}_e}{\|\boldsymbol{t}_e\|} \\ \boldsymbol{t}_e = \frac{\boldsymbol{t}}{t_x} = [1, (A_d^+\boldsymbol{b}_d)^T]^T, \quad t_x \neq 0 \\ \left(\delta R\boldsymbol{u}^c_j(t_k)\right) \times \boldsymbol{u}^c_j(t_{k+1}) = [d_{x,j}, d_{y,j}, d_{y,j}]^T, j = 1...N\end{cases} \tag{5.8}$$

with A_d and $\boldsymbol{b}_d$ as defined in equation (3.31). $\boldsymbol{u}^c_j(t_k) = K^{-1}\underline{\boldsymbol{x}}^m_j(t_k)$, K being the projection matrix from equation (3.4), $\underline{\boldsymbol{x}}^m_j(t_k) = [r_j(t_k), s_j(t_k), 1]^T$ being the position of feature j at time t_k in homogeneous coordinates, $\boldsymbol{u}^c_j(t_{k+1}) = K^{-1}\underline{\boldsymbol{x}}^m_j(t_{k+1})$, $\underline{\boldsymbol{x}}^m_j(t_{k+1}) = [r_j + dr_j, s_j + ds_j, 1]^T$ being the position of feature j at time t_{k+1}, δR being the rotation matrix describing the rotation of the camera from time t_k to time t_{k+1}, as described in 3.3.2.

5.6.5 Stability Proof

Provided Assumptions 5.1-5.2 and Assumptions 5.9-5.11, the following theorem can be stated to describe the stability of the proposed observer

Theorem 3. ***(Stability of the EOF observer)*** *Let σ be chosen to ensure stability according to Lemma 1 in Grip et al. (2012*a*) and define $H_K(s) = (Is - A_w + K_w C_w)^{-1} B_w$. There exists a set (0,c) such that, if K_w is chosen such that $A_w - K_w C_w$ is Hurwitz, and $\|H_K(s)\|_\infty < \gamma$, for $\gamma \in (0, c)$, then the origin of the error dynamics equation* (5.2)-(5.3) *with computer vision subsystem as defined in equation* (5.8)*, provided Assumptions 5.1-5.2 and 5.9-5.11 and that $\lim_{\Delta t \to 0} \frac{\boldsymbol{p}^b_{b/n}(t_k+\Delta t) - \boldsymbol{p}^b_{b/n}(t_k)}{\Delta t} = \boldsymbol{v}^b_{b/n}$ with t_k being the time when image k is captured and Δt being the time between two consecutive images, is USGES when the initial conditions satisfy $\|\hat{b}^b_{\mathrm{gyro}}(0)\| \leq L_{\hat{b}}$.*

Proof. The proof is based on Fusini et al. (2014), where we have replaced M with the new computer vision subsystem. We must show that $\boldsymbol{v}^b_{\mathrm{cv}}$ is uniquely defined. Then it follows from Theorem 1 that the origin of the error dynamics equation (5.2)-(5.3) with the CV subsystem as defined in equation (5.8) is USGES.

Provided $\lim_{\Delta t \to 0} \frac{\boldsymbol{p}^b_{b/n}(t_k+\Delta t) - \boldsymbol{p}^b_{b/n}(t_k)}{\Delta t} = \boldsymbol{v}^b_{b/n}$, then $\boldsymbol{v}^b_{\mathrm{cv}} = \boldsymbol{v}^b_{b/n}$, $\boldsymbol{v}^b_{\mathrm{cv}}$ has a one to one mapping to $\boldsymbol{t}_e$ if the sign of t_x is known. From Assumption 5.10 $t_x > 0$, hence the uniqueness of $\boldsymbol{v}^b_{\mathrm{cv}}$ can be shown by the uniqueness of $\boldsymbol{t}_e$. $\boldsymbol{t}_e = [1, (A_d^+ \boldsymbol{b}_d)^T]^T$ has a unique solution if and only if the rank of A_d is full (Kreyszig, 2006). Given that the computer vision algorithm extracts features such that Assumption 5.11 is not violated, then A_d has full rank, and $\boldsymbol{t}_e$ is uniquely determined. Hence $\boldsymbol{v}^b_{\mathrm{cv}}$ is uniquely determined, and it follows from Theorem 1 that the system is USGES. □

5.6.6 Discrete Representation

The CV of the EOF NO approximates the normalized body-fixed linear velocity by the direction of translation (the change in position). For numerical and implementational reasons it might be smarter to use change in estimated position, $\hat{\boldsymbol{p}}^n_{b/n}(t) - \hat{\boldsymbol{p}}^n_{b/n}(t - \Delta t)$, as reference vector in the EOF NO. EOF(Δp) NO denotes the NO with EOF CV, and change in position, $\hat{\boldsymbol{p}}^n_{b/n}(t)$, as reference vector. No stability analysis is performed for the EOF(Δp) NO, but an experimental validation is given in Section 8.3. The output injection term $\hat{J}$ for the EOF(Δp) NO then takes the form:

$$\hat{J}(\boldsymbol{v}^b_{\mathrm{cv}},\ \Delta\hat{\boldsymbol{p}}^n_{b/n}, \boldsymbol{f}^b_{\mathrm{imu}}, \hat{\boldsymbol{f}}^n_{b/n}, \hat{R}) := \hat{A}_n A_b^T - \hat{R} A_b A_b^T \tag{5.9a}$$

$$A_b := \left[\frac{\boldsymbol{f}^b_{\mathrm{imu}}}{\|\boldsymbol{f}^b_{\mathrm{imu}}\|}, \frac{\boldsymbol{f}^b_{\mathrm{imu}}}{\|\boldsymbol{f}^b_{\mathrm{imu}}\|} \times \frac{\boldsymbol{v}^b_{\mathrm{cv}}}{\|\boldsymbol{v}^b_{\mathrm{cv}}\|}, \frac{\boldsymbol{f}^b_{\mathrm{imu}}}{\|\boldsymbol{f}^b_{\mathrm{imu}}\|} \times \left(\frac{\boldsymbol{f}^b_{\mathrm{imu}}}{\|\boldsymbol{f}^b_{\mathrm{imu}}\|} \times \frac{\boldsymbol{v}^b_{\mathrm{cv}}}{\|\boldsymbol{v}^b_{\mathrm{cv}}\|} \right) \right] \tag{5.9b}$$

$$\hat{A}_n := \left[\frac{\hat{\boldsymbol{f}}^n_{b/n}}{\|\hat{\boldsymbol{f}}^n_{b/n}\|}, \frac{\hat{\boldsymbol{f}}^n_{b/n}}{\|\hat{\boldsymbol{f}}^n_{b/n}\|} \times \frac{\Delta\hat{\boldsymbol{p}}^n_{b/n}}{\|\Delta\hat{\boldsymbol{p}}^n_{b/n}\|}, \frac{\hat{\boldsymbol{f}}^n_{b/n}}{\|\hat{\boldsymbol{f}}^n_{b/n}\|} \times \left(\frac{\hat{\boldsymbol{f}}^n_{b/n}}{\|\hat{\boldsymbol{f}}^n_{b/n}\|} \times \frac{\Delta\hat{\boldsymbol{p}}^n_{b/n}}{\|\Delta\hat{\boldsymbol{p}}^n_{b/n}\|} \right) \right] \tag{5.9c}$$

$$\Delta\hat{\boldsymbol{p}}^n_{b/n} = \hat{\boldsymbol{p}}^n_{b/n}(t) - \hat{\boldsymbol{p}}^n_{b/n}(t - \Delta t) \tag{5.9d}$$

5.7 Comparison of the Different Observer Representations

The different NOs differ only in the CV subsystem, hence a comparison can be done only with respect to the CV subsystem.

The GTOF observer relies on the OF algorithm to provide a sufficient number of OF vectors, such that M has full rank and equation (3.10) can be used to find $\boldsymbol{v}^b_{\mathrm{cv}} = \boldsymbol{v}^b_{b/n}$. The EOF and CEOF have similar requirements. The EOF observer relies on that the OF algorithm provides at least two linearly independent pairs of image features, as defined in Definition 7, while CEOF needs at least two linearly independent OF vectors as defined in Definition 8. In general all NO representations needs to have some distribution of the image features, such that the pseudo-inverse of M from equation (5.5), A_d from equation (5.8) and A_c from equation (5.7) exists.

The GTOF observer depends on knowing the roll, pitch and altitude of the UAV relative to the ground to calculate the body-fixed linear velocity. This could be measured by an inclinometer and an altimeter. The EOF does only depend on the relative rotation in the time between two consecutive images are taken. This can be recovered from the gyroscope by configuring it to output incremental angles, or by integrating the angular velocity in the time between two images are taken. As the time between two consecutive images are relatively low and the bias of the measurements are small the relative orientation will not drift away. The CEOF observer relies only on the angular velocity, which can be extracted from a gyroscope. The gyroscope is a standard sensor in low cost IMUs, while inclinometer and altimeter often must be added as separate sensors.

The GTOF observer does require the terrain being filmed to be flat. This requirement is not present in the CEOF and EOF observers, and might be the biggest advantage with these.

The GTOF and CEOF observers are derived assuming continuous OF, that is $\dot{r}$

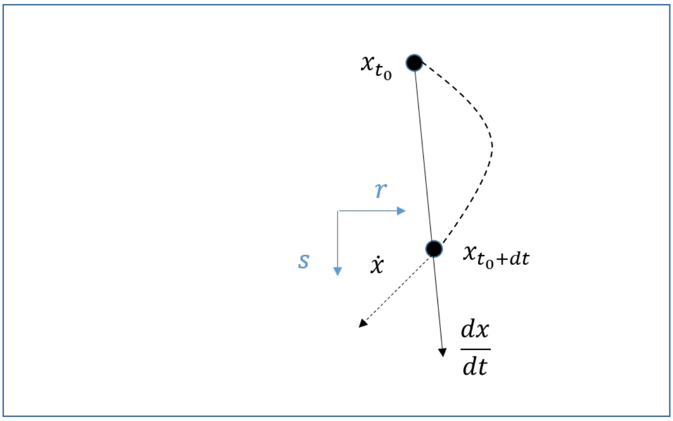


Figure 5.5: The GTOF and the CEOF nonlinear observers approximate the optical flow by the discrete optical flow vectors. As can be seen, big rotations will cause the discrete optical flow to differ much from the optical flow. This is handled by using sufficiently high frame rate for the camera.

and $\dot{s}$, and yields body-fixed linear velocity. However the OF algorithm measures the discrete OF $\frac{\Delta r}{\Delta t}$ and $\frac{\Delta s}{\Delta t}$. For the GTOF and the CEOF the OF is approximated by the discrete OF, meaning that the discretization is done with the OF vectors. The EOF observer is derived assuming discrete OF, and makes use of Δr and Δs directly. The EOF uses these discrete OF vectors to calculate a normalized vector of the direction of translation. The direction of translation is used to approximate the direction of the body-fixed velocity, meaning the discretization is done with the velocity vector itself. Hence both the GTOF, CEOF and the EOF observers does some discretization to get measures of the body-fixed velocity. The GTOF and CEOF discretization is illustrated in Figure 5.5, while the EOF discretization is illustrated in Figure 5.6.

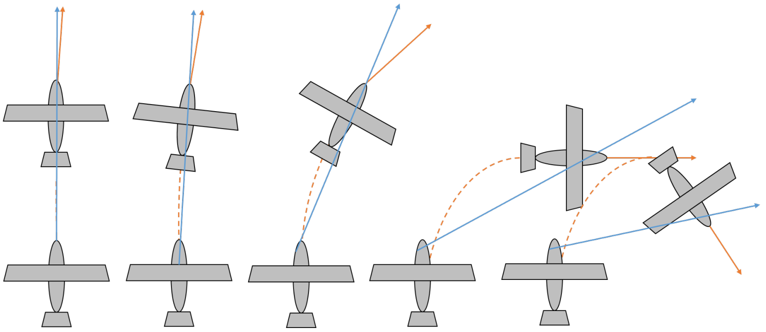


Figure 5.6: The EOF nonlinear observer does a discretization step on the velocity vector from the computer vision subsystem. The orange arrow illustrates the body-fixed velocity, while the blue indicates the direction of translation. As can be seen, the error between these two vectors will increase when flying steep turns, or when having a big time interval between images are captured. This is handled by a sufficiently large frame rate for the camera.

Part III

Methods

Chapter 6

Experiment

The objective of the experiment is to compare the performance of the nonlinear observers (NO) from Chapter 5 with the multiplicative extended Kalman filter (MEKF) from Chapter 4. A UAV flight test is conducted to collect real measurement data. The performance of the different NOs and MEKFs are compared to a reference. The reference is an extended Kalman filter (EKF) implemented on the onboard UAV autopilot. The experimental results are organized in three cases, which will be explained more closely in Section 6.4.1-6.4.3. The cases are:

Case 1: Ground Truth Optical Flow observers Two NOs and one MEKF, all with with GTOF CV, are compared to each other and a reference.

Case 2: Continuous Epipolar Optical Flow observers A NO and a MEKF utilizing the CEOF CV are tested with real world data. They are compared to each other and to a reference.

Case 3: Epipolar Optical Flow observers Three navigation systems are tested. The EOF NO, a variant of the EOF NO and a MEKF are compared to each other and a reference.

The chapter is organized as follows. First a description of a UAV flight experiment is presented in Section 6.1. Then computer vision and observer tuning parameters are highlighted in Section 6.2. In Section 6.3 different evaluation criteria are defined. Then three different cases are motivated and described in Section 6.4.

Figure 6.1: The team after the flight test at Eggemoen

6.1 UAV Flight Test

In order to collect data for testing, a flight experiment have been conducted. The experiment was performed the 6^{th} of February 2015 at Eggemoen, Norway - a sunny white winter day. A Penguin B UAV, produced by UAV Factory, was equipped with a custom payload module in order to record the desired data, namely camera images and sensor data from IMU and GPS. The UAV and the team performing the flight test are shown in Figure 6.1. The camera was set to capture images at a rate of ten frames per second(FPS), and the images were captured at a resolution of 1600×1200 pixels. The payload module was placed so that the camera was pointing straight down (towards the ground) when the UAV has zero roll and pitch. OF algorithms are deployed on the images from the flight in order to generate OF vectors that can be used by the observers. The IMU, a STIM300, was set to log data on 300Hz. The IMU was configured to output angular rates during the flight test. The GPS logged measurements at 5Hz. Table 6.1 summarizes the measurement rates of the payload sensors. Velocity measurements are obtained from the GPS by differentiation of position measurements. For more information about the payload setup see Appendix E. The results presented in this section has a time axis from 1050 seconds to 1300 seconds. This is the time since the data logging began. Only the significant part involving the flight is presented.

The cruise altitude of the UAV was approximately 150 meters above ground level,

Table 6.1: Measurement rates of the sensors in the payload

Sensor	Rate
IMU	300Hz
CV	10Hz
GPS	5Hz

and the flight lasted for approximately four minutes. Earth-Centered Earth-Fixed (ECEF) position measurements from GPS, accelerometer-, gyroscope- and inclinometer-measurements from an IMU, as well as camera images were logged with a time stamp and stored on-board the UAV. GPS measurements were converted to North-East-Down coordinates after the flight before they were used in the experimental testing. The flight test was conducted over a relatively flat area, but the area around the airfield at Eggemoen has a lot of trees, meaning the actual terrain altitude will vary a bit.

6.2 Implementation for Off-line Testing

This section presents some important parameters for the computer vision subsystem as well as the NO and the MEKF.

6.2.1 Computer Vision

The camera projection matrix was verified with the Matlab `computer vision toolbox`. The camera and lens ware tested to be very accurate, with negligible distortion. Therefore no distortion correction was done with the images. The images from the camera on the UAV were processed at their original resolution of 1600×1200 pixels. The Template Matching and SIFT OF algorithms, as described in Section 3.2.1 and Section 3.2.2 respectively, are implemented in C++ with the open source library OpenCV (Bradski, 2000). The OF vectors from these two algorithms are used as input to the CV system of the different observers. By using two OF algorithms, the chance of having at least two or three OF vectors increases (recall from Chapter 3 that the GTOF method needs at least three OF vectors, while the EOF and CEOF method needs at least two). For the Template Matching algorithm, there are some important parameters to consider listed in Table 6.2. For implementation details and tuning values for the SIFT algorithm it is referred to Appendix D.1.

Table 6.2: Various parameters for the Template Matching algorithm

Description	**Value**
Image height	1200 pixels.
Image width	1600 pixels.
Template height	90 pixels.
Template width	120 pixels.
Matching method	normalized cross correlation.
Points of interest	The centre of 12 equally spaced regions in the image.

6.2.2 Multiplicative Extended Kalman Filter

The covariance parameters of the MEKF used in the experimental testing are listed in Table 6.3. The initial error covariance matrix $\bar{P}_0 = E[(\boldsymbol{v}(0) - \hat{\boldsymbol{v}}(0))(\boldsymbol{v}(0) - \hat{\boldsymbol{v}}(0))^T]$ with $\boldsymbol{v} = [\delta\boldsymbol{\epsilon}^T, (\boldsymbol{b}^b_{\text{gyro}})^T, (\boldsymbol{p}^n_{b/n})^T, (\boldsymbol{v}^n_{b/n})^T, (\boldsymbol{b}^b_{\text{acc}})^T]$, with values from the table. Process noise covariance matrix is $R = \text{diag}\left[\sigma^2_{\delta\boldsymbol{\epsilon}}, \sigma^2_{\boldsymbol{b}^b_{\text{gyro}}}, \sigma^2_{\boldsymbol{p}^n_{b/n}}, \sigma^2_{\boldsymbol{v}^n_{b/n}}, \sigma^2_{\boldsymbol{b}^b_{\text{acc}}}\right]$ and the measurement noise covariance matrix is $Q = \text{diag}\left[\sigma^2_{\boldsymbol{v}^b_{\text{cv}}}, \sigma^2_{\boldsymbol{f}^b_{\text{imu}}}, \sigma^2_{\boldsymbol{p}^n_{\text{GPS}}}, \sigma^2_{\boldsymbol{v}^n_{\text{GPS}}}\right]$ with values defined in Table 6.3. Some comments on the covariance parameters are appropriate. The covariance of 10^{-3} on GPS position measurement may sound extreme low, but have been found to work well in the MEKF with the dataset from the experiment. Since the velocity measurement is obtained by differentiation of the GPS measurement, it should in general not be very accurate. However a noise covariance of 10^{-2} for the GPS velocity measurements is found to yield satisfactory results. Further implementation details about the MEKF are found in Appendix D.2.

6.2.3 Nonlinear Observer

The observer gains are chosen in compliance with the Lyapunov stability from Theorem 1 on page 63, and are listed in Table 6.4. Further implementation details about the NO are found in Appendix D.3.

6.3 Evaluation Criteria

CV measurements are normalized before use. When a three dimensional vector is normalized, the resulting vector only has two degrees of freedom. This is because any vector with unit length can be constructed from two of its elements. As the measurement provide information in two degrees of freedom, it is interesting to express the CV measurement by the crab and flight path angles. The measured

Table 6.3: MEKF covariance parameters in experimental testing

Error covariance		
Variable $\boldsymbol{v}$	**Value $E[(\boldsymbol{v}(0) - \hat{\boldsymbol{v}}(0))(\boldsymbol{v}(0) - \hat{\boldsymbol{v}}(0))^T]$**	**Variance in**
$\delta\boldsymbol{\epsilon}$	$[10^{-7}, 10^{-7}, 10^{-7}]^T$	Attitude
$\boldsymbol{b}^b_{\text{gyro}}$	$[10^{-12}, 10^{-12}, 10^{-12}]^T$	Gyro bias
$\boldsymbol{p}^n_{b/n}$	$[10^{-4}, 10^{-4}, 10^{-4}]^T$	NED position
$\boldsymbol{v}^n_{b/n}$	$[10^{-7}, 10^{-7}, 10^{-7}]^T$	NED velocity
$\boldsymbol{b}^b_{\text{acc}}$	$[10^{-11}, 10^{-11}, 10^{-11}]^T$	Accelerometer bias
Process noise		
Variable $\boldsymbol{v}$	**Value $\sigma^2_{\boldsymbol{v}}$**	**Variance in**
$\delta\boldsymbol{\epsilon}$	$[5 \cdot 10^{-7}, 5 \cdot 10^{-7}, 5 \cdot 10^{-6}]^T$	$\delta\boldsymbol{\epsilon}$
$\boldsymbol{b}^b_{\text{gyro}}$	$[5 \cdot 10^{-13}, 5 \cdot 10^{-13}, 1 \cdot 10^{-12}]^T$	Gyro bias
$\boldsymbol{p}^n_{b/n}$	$[10^{-4}, 10^{-4}, 10^{-4}]^T$	NED position
$\boldsymbol{v}^n_{b/n}$	$[10^{-4}, 10^{-4}, 10^{-4}]^T$	NED velocity
$\boldsymbol{b}^b_{\text{acc}}$	$[10^{-12}, 10^{-12}, 10^{-12}]^T$	Accelerometer bias
Measurement noise		
Variable $\boldsymbol{v}$	**Value $\sigma^2_{\boldsymbol{v}}$**	**Variance in**
$\boldsymbol{v}^b_{\text{cv}}$	$[2 \cdot 10^{-3}, 2 \cdot 10^{-3}, 2 \cdot 10^{-3}]^T$	Computer vision
$\boldsymbol{f}^b_{\text{imu}}$	$[2 \cdot 10^{-3}, 2 \cdot 10^{-3}, 2 \cdot 10^{-3}]^T$	Accelerometer
$\boldsymbol{p}^n_{\text{GPS}}$	$[10^{-3}, 10^{-3}, 10^{-3}]^T$	GPS position
$\boldsymbol{v}^n_{\text{GPS}}$	$[10^{-2}, 10^{-2}, 10^{-2}]^T$	GPS velocity

crab and flight path angles are compared to the estimated crab and flight path angle from the Piccolo autopilot EKF (the EKF does not really estimate these angles, but the attitude and velocity estimates can be used to find the angles). Although there is no guarantee of the accuracy of the estimates from the Piccolo autopilot, it is reason to think that it provides estimates not far from the true value. The UAV was during parts of the flight controlled by the Piccolo autopilot with success, and this implies that the EKF estimated correct navigation states. Two quantities are therefore defined. $\tilde{\chi}$ is a measure of the crab angle error of the measurement from the CV, while $\tilde{\gamma}$ is a measure of the flight path angle error. Figure 6.2 illustrates $\tilde{\gamma}$ and $\tilde{\chi}$.

$$\tilde{\chi} = \arcsin\left(v_{\text{cv}} - v_{\text{ref}}\right) \tag{6.1}$$

$$\tilde{\gamma} = \arcsin\left(w_{\text{cv}} - w_{\text{ref}}\right) \tag{6.2}$$

In order to evaluate the performance of the different NOs and MEKFs, the root mean squared (RMS) error in the estimated states will be assessed. The Piccolo autopilot EKF is used as a reference. The log from the Piccolo is only at 1 Hz. The Piccolo data is resampled to 300 Hz with linear interpolation before it is used as a reference. Although the accuracy of the Piccolo autopilot EKF is unknown, it is assumed to produce estimates that are not far from the correct values. As the validity of the Piccolo autopilot EKF is unknown, the RMS values cannot be used

Table 6.4: Nonlinear observer gains for experimental testing

Variable	Value	Description
	Common Nonlinear Observer Parameter Values	
L_b	$2^\circ = 2/180 \cdot \pi\mathrm{rad}$	Upper bound on norm of gyro bias
$L_{\hat{b}}$	$2.1^\circ = 2.1/180 \cdot \pi\mathrm{rad}$	Upper bound on norm of estimated gyro bias
K_P	$\mathrm{diag}[0.08, 0.04, 0.06]$	Symmetric positive definite gain matrix for attitude
k_I	0.0001	Positive scalar gain for gyro bias estimate
σ	1	Scaling factor tuned to achieve stability
K_{pp}	$30I_{3\times3}$	Translational gains
K_{pv}	$2I_{3\times3}$	-
K_{vp}	$0.01I_{3\times3}$	-
K_{vv}	$20I_{3\times3}$	-
$K_{\xi p}$	$I_{3\times3}$	-
$K_{\xi v}$	$50I_{3\times3}$	-

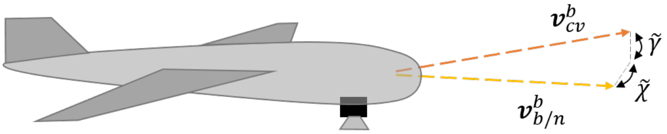


Figure 6.2: Crab angle error and flight path angle error are measures of the accuracy of the computer vision velocity measurement.

to state whether one observer is better than the other. The RMS is however used to give some indication of the validity of the estimates. As the observers needs some time to converge, the RMS is measured from after time 1150 seconds (100 seconds after the flight begins). The RMS error is defined as:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x}(i) - x_{\mathrm{ref}}(i))^2} \tag{6.3}$$

The observers will also be compared with respect to their computationally efficiency, by the execution time. The execution time is the average time of one iteration of the respective observer. The execution time is found by the `tic` and `toc` functions in Matlab. The observers are all tested on a desktop computer with specifications given in Table 6.5.

6.4 Description of Cases

All cases are evaluated with sensor measurements from the UAV flight. A description of each case and motivation will be presented in the forthcoming.

Table 6.5: Specification of computer used to test the various observers

Manufacturer	Dell
OS	Windows 7 Enterprise
RAM	16 GB
Processor	Intel(R) Core i7 - 3.40GHz
System Type	64-bit

6.4.1 Case 1: Ground Truth Optical Flow Observers

This case aims at validating the theory in practice, by looking at the performance of the GTOF NO (described in Section 5.4) and a MEKF with real experimental data. It is interesting to compare the performance of the GTOF NO from Fusini et al. (2014) with a traditional MEKF. Moreover it is interesting to see whether introducing a feedback to the CV in the GTOF NO will improve the performance of the observer. In this case the following three observers are implemented, and tested on real sensor data from a UAV flight:

- The **GTOF NO** from Section 5.4. CV uses inclinometer measurements of the attitude.
- The **GTOF(f) NO** is similar to the GTOF NO, but uses feedback of the estimated attitude in the CV subsystem.
- The **MEKF** from Chapter 4 with the same CV as the GTOF NO with inclinometer measurements.

The MEKF uses the GTOF CV as described in Section 3.3.1. Recall that GTOF CV needs to know the attitude of the UAV relative to the ground. The CV in the MEKF uses an inclinometer to obtain the attitude. The NO is, as described above, implemented with two different variants of the GTOF CV. The first variant uses inclinometer measurements as input to CV, and is addressed as GTOF NO. The second variant uses attitude feedback from the observer to CV, and is addressed as GTOF(f) NO. Recall that a stability proof and experimental validation have been presented only for the first variant, using inclinometer measurements (Fusini et al., 2014, 2015). In this case an experimental validation of the GTOF(f) NO with attitude feedback to CV will be presented. As no altimeter was available from the conducted flight experiment, the UAVs height relative to the taxiing area on the airfield is used to approximate the height relative to the ground. The height of the UAV is recovered by: $c_z^n = [0,0,1]\boldsymbol{p}_{\text{GPS}}^n$ and $c_z^n = [0,0,1]\hat{\boldsymbol{p}}_{b/n}^n$ for GTOF and GTOF(f) respectively. The results from this case is presented in Section 8.1.

6.4.2 Case 2: Continuous Epipolar Optical Flow Observers

This case aims to test the performance of the proposed CEOF NO (Section 5.5) with the experimental data from the UAV flight. The CEOF NO is compared to a CEOF MEKF (The CEOF MEKF uses the same CV as the CEOF NO). The motivation behind choosing this case is to validate the proposed CEOF NO experimentally. Furthermore it is interesting to see its performance compared to a MEKF. The following two observers are tested and evaluated:

- The **CEOF NO** from Section 5.5.
- The **MEKF** from Chapter 4 with the same CV as the CEOF NO.

The results from this case is presented in Section 8.2.

6.4.3 Case 3: Epipolar Optical Flow Observers

The purpose of this case is to validate the proposed EOF NO from Section 5.6 in practice. Moreover it is also the purpose to compare the EOF NO with the proposed discrete NO from Section 5.6.6, namely the EOF(Δp) NO. These two observers are further compared to a MEKF utilizing the EOF CV as described in Section 3.3.2. The following three observers are compared with sensor data from the UAV test flight:

- The **EOF NO** from Section 5.6.
- The **EOF**(Δp) **NO** from Section 5.6.6.
- The **MEKF** from Chapter 4 with the same CV as the EOF NO.

As the IMU was configured to output angular rates, the rotation matrix δR, relating the UAV orientation between two consecutive images, is found by integrating the bias corrected (using estimated bias) angular velocity measurements. The results from this case is presented in Section 8.3.

Chapter 7

Simulation

This chapter presents simulations of a UAV flying over a terrain mimicking a Norwegian coastline environment. The objective of is to compare the performance of the GTOF NO and CEOF NO when flying over rugged and elevated terrain. The UAV flight test described in Chapter 6 is conducted over a relatively flat area. In order to evaluate the different observers with more rugged terrain, a simulator is implemented. Moreover, these two observers are compared to a nonlinear observer without computer vision. The observer without CV is addressed as NoCV, as it has no CV. The NoCV observer is implemented in order to evaluate whether introducing CV in the observer will improve the performance of the observer. NoCV is realized by removing the CV subsystem, and using a pseudo measurement of body-fixed velocity directed straight forward (setting $v^b_{\text{cv}} = [1, 0, 0]^T$). This means that the NoCV observer uses only IMU and GNSS measurements. The observers will be compared to a known reference. The different NOs considered in the simulation are:

- The GTOF NO described in Section 5.4.
- The CEOF NO described in Section 5.5.
- The NoCV NO (a NO without CV).

The chapter is divided in three main parts. First a description of the implemented simulator is presented, then the observer implementation is highlighted. The chapter ends with description of a case study.

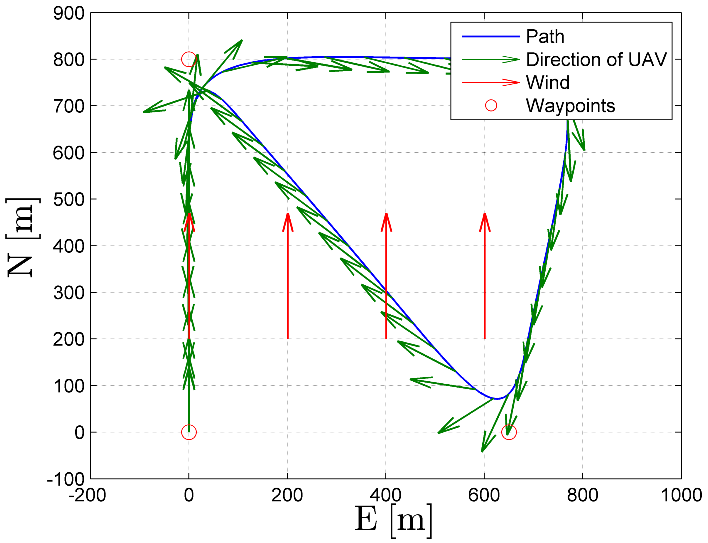


Figure 7.1: Waypoints are made to generate a path that the UAV follows. By adding wind of 5m/s directed north to the simulation, the UAV experiences some crab angle when not flying straight north or south.

7.1 Simulator

A simulator is implemented in Matlab to test the different NOs when flying over realistic rugged terrain.

7.1.1 Simulating a UAV Flight

Multiple waypoints for the UAV are defined and is the basis for the generation of a path for the UAV to follow. The waypoints contains 3D position, pitch angle of UAV and time of arrival. A linear interpolation is done between these points in order to obtain a path in time. A fourth order lowpass filtering is done on the path of 3D points and pitch to get a smooth path that the UAV can follow. NED velocity of the UAV is found by differentiating the position coordinates of the smooth path. Some wind is added to simulation (in the simulation presented here wind of 5m/s directed straight north is implemented). Yaw is generated from the NED velocity of the UAV and the wind. Figure 7.1 displays the waypoints, simulated path, wind and the direction of the UAV when flying along the path. Roll is generated as a function of the yaw rate. Linear and angular velocity is then generated by differentiation. Linear acceleration is found by differentiating the expression for linear velocity.

Table 7.1: Measurement rates of the sensors in the simulation

Sensor	**Rate**
IMU	100Hz
Inclinometer	100Hz
CV	25Hz
GNSS	5Hz

7.1.2 Sensor Data

Sensor data must be generated before running the observer. A gyroscope, accelerometer, inclinometer, GNSS and CV is simulated. The GNSS is simulated to measure {n} position and velocity and CV is simulated to measure the OF. The gyroscope, accelerometer and inclinometer are configured to output measurements with a rate of 100 Hz. The camera is simulated to take 25 frames per second. The camera simulator is described in detail in Appendix D.4. The GNSS is configured to output measurements at 5Hz. The measurement rates are listed in Table 7.1.

Noise is added to the position measurement from GNSS. The GNSS noise is modelled as a Gauss-Markov process by $\nu(k+1) = e^{-K_{\text{GNSS}}\Delta T}\nu(k) + \eta_{\text{GNSS}}$, with noise parameters given in Table 7.2.

Table 7.2: Gauss-Markov error model parameters for GNSS position measurements.

Direction	*Std. dev.* η_{GNSS}[m]	$1/K_{\text{GNSS}}$[s]	ΔT_{GNSS}[s]
North	0.21	360	0.2
East	0.21	360	0.2
Down	0.4	360	0.2

White noise is added to the IMU, inclinometer, camera and velocity from GNSS sensor data by the multivariate normal random noise-function, `mvnrnd`, in Matlab. Inclinometer measurements are denoted $\boldsymbol{\Theta}^b_{\text{incl}} = [\phi, \theta]^T$. The measurement from CV is normalized before it is used in the NO, meaning that the measurement from the altimeter provide some scale information about the body-fixed velocity that will not be used, hence no noise is added to the altimeter. The following mean and covariance are used:

$$\begin{aligned}
\boldsymbol{w}_{\boldsymbol{\omega}^b_{\text{imu}}} &\sim \mathcal{N}(0_{3\times1}, \Sigma_{\boldsymbol{\omega}^b_{\text{imu}}}), \quad \Sigma_{\boldsymbol{\omega}^b_{\text{imu}}} = (0.135°/\text{s})^2 I_{3\times3} \\
\boldsymbol{w}_{\boldsymbol{f}^b_{\text{imu}}} &\sim \mathcal{N}(0_{3\times1}, \Sigma_{\boldsymbol{f}^b_{\text{imu}}}), \quad \Sigma_{\boldsymbol{f}^b_{\text{imu}}} = (1.29\cdot10^{-3}\text{g})^2 I_{3\times3} \\
\boldsymbol{w}_{\boldsymbol{\Theta}^b_{\text{incl}}} &\sim \mathcal{N}(0_{2\times1}, \Sigma_{\boldsymbol{\Theta}^b_{\text{incl}}}), \quad \Sigma_{\boldsymbol{\Theta}^b_{\text{incl}}} = (0.18°)^2 I_{2\times2} \\
\boldsymbol{w}_{\boldsymbol{v}^n_{\text{GNSS}}} &\sim \mathcal{N}(0_{3\times1}, \Sigma_{\boldsymbol{v}^n_{\text{GNSS}}}), \quad \Sigma_{\boldsymbol{v}^n_{\text{GNSS}}} = (0.21\text{m/s})^2 I_{3\times3}
\end{aligned}$$

No bias on the accelerometer is assumed, and a constant bias is assumed on the gyroscope. The gyroscope bias is given the following value

$$\boldsymbol{b}^b_{\text{gyro}} = \begin{bmatrix} 0.1°/\text{s} \\ -0.3°/\text{s} \\ -0.35°/\text{s} \end{bmatrix}$$

White noise is also added to the OF data from the simulated camera. Every extracted feature is given some white noise with variance, $\sigma^2_{\Delta r} = \sigma^2_{\Delta s} = \sigma^2_{\Delta} = (4.5 \cdot 10^{-5}\text{mm})^2$. As two corresponding features is needed to get an OF vector, the resulting noise of the OF vector has variance $\sigma_{OF} = \sigma^2_{\Delta} I_{2\times2}$. On a camera chip with 1600 × 1200 pixels and dimension 7.2 × 5.4 mm, this would yield noise with a small variance of $\sigma^2_{OF} = 2 \cdot (0.01 \times \text{pixels})^2$.

7.1.3 Terrain Simulation

In order to evaluate the performance of the different observer representations with realistic environments, a terrain model is generated. The terrain model is a matrix, Z, with values corresponding to the elevation profile of the terrain. The terrain model is also called the elevation profile of the terrain, as it describes the elevation of the terrain. At position x, y of the matrix the elevation h of the terrain at x meters north and y meters east is found. A point on the surface of the terrain will have NED coordinate $x, y, -h$. The terrain model has a resolution of 1m × 1m meter. The terrain model is covering 1km × 1km. When the UAV flies out of this range, the elevation of the terrain is considered to be zero. The terrain model is implemented to imitate a Norwegian coastline as illustrated in Figure 7.2.

7.2 Observer Implementation

Three observers are implemented: the GTOF NO from Fusini et al. (2014) (described in Section 5.4), the proposed CEOF NO described in Section 5.5 and a NO without CV, named NoCV NO. The NoCV NO is similar to the CEOF and GTOF NO, but using a pseudomeasurement of the body-fixed linear velocity rather than having a CV system to measure the velocity. The NoCV NO assumes no crab and flight path angle by imposing $\boldsymbol{v}^b_{\text{cv}} = [1, 0, 0]^T$. Further observer implementation details are found in Appendix D. The parameters and gains for the observers are listed in Table 7.3. All observers are initialized with $\hat{\boldsymbol{b}}^b_{\text{gyro}} = 0_{3\times1}$ and $\hat{R}^b_n = I_{3\times3}$. The velocity and position estimates are initialized by the first GNSS velocity and position measurements.

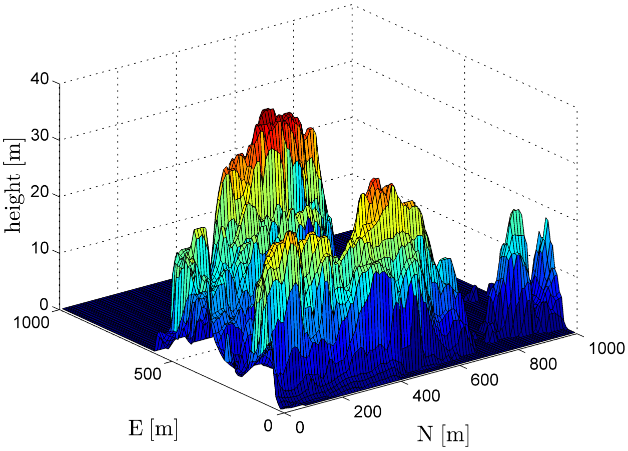


Figure 7.2: The terrain model implemented in the simulator is a coastline.

Table 7.3: Nonlinear observer gains for testing on simulated data.

Variable	**Value**	**Description**
	Common Parameter Values	
L_b	$2^\circ = 2/180 \cdot \pi \text{rad}$	Upper bound on norm of gyro bias
$L_{\hat{b}}$	$2.1^\circ = 2.1/180 \cdot \pi \text{rad}$	Upper bound on norm of estimated gyro bias
σ	1	Scaling factor tuned to achieve stability
K_{pp}	$\text{diag}[5, 5, 0.7]$	Translational gains
K_{pv}	$50 I_{3\times3}$	
K_{vp}	$\text{diag}[0.1, 0.1, 0.01]$	
K_{vv}	$10 I_{3\times3}$	
$K_{\xi p}$	$0.1 I_{3\times3}$	
$K_{\xi v}$	$5 I_{3\times3}$	
	GTOF and CEOF Parameter Values	
K_P	$I_{3\times3}$	Symmetric positive definite gain matrix for attitude
k_I	0.03	Positive scalar gain for gyro bias estimate
	NoCV Parameter Values	
K_P	$\text{diag}[1, 0.2, 0.1]$	Symmetric positive definite gain matrix for attitude
k_I	0.01	Positive scalar gain for gyro bias estimate

7.3 Description of Case 4: Navigating in the Coastline

An elevation profile of the coastline in Norway, covering 1 × 1km, is acquired. The elevation model has a resolution of one times one meter. The elevation model is shown together with the UAV flight path in Figure 7.3. This elevation model mimics very well a real environment which can be found when navigating in the Norwegian coastline. The motivation behind choosing this terrain is to see how the different observer representations perform when flying over a terrain with both flat and rugged areas.

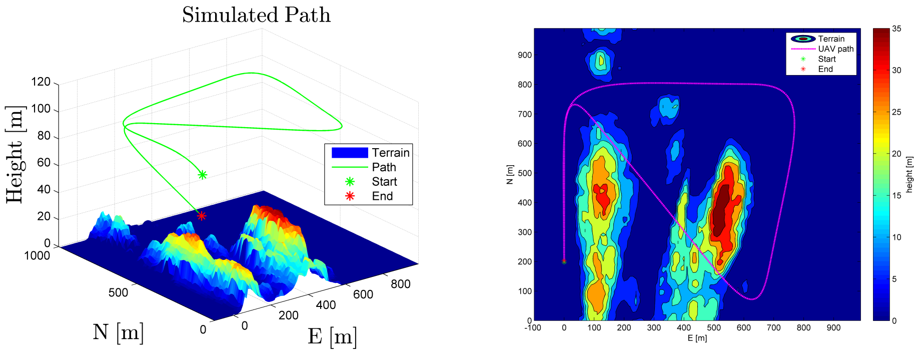


Figure 7.3: UAV flight path over the coastline.

In the Norwegian coastline it is always windy. Therefore wind with direction straight north with a magnitude of 5m/s is simulated. This wind causes the UAV to have a crab angle when not flying straight north or south. As the CV measures motion relative to the ground, and not the airspeed, one does not have to consider sideslip angles.

The UAV path can be divided into the following segments:

1. Start (20-40 seconds): The UAV flies along the skerries towards the ocean in north.

2. Ocean (40-110 seconds): The UAV flies above the sea, navigating on straight lines and taking three right turns.

3. Non-planar (110-180 seconds): The UAV flies in over the skerries back to the starting point.

4. End (180-200 seconds): The UAV takes a steep turn left, before it flies towards south along the skerries. At the same time the UAV is losing altitude.

This case seeks to evaluate how the GTOF, CEOF and NoCV NO performs when flying over non-horizontal planar terrain. Moreover the objective is to see whether introducing CV improves the accuracy of the observer. The observers are compared to a known reference.

Part IV

Results and Discussion

Chapter 8

Experiment Results

This chapter presents the results from the experiment described in Chapter 6.

8.1 Case 1: Ground Truth Optical Flow Observers

This section presents the results from the case described in Section 6.4.1. The case is organized in four parts: CV measurements, Attitude Estimates, Velocity and Position Estimates and Bias Estimates.

8.1.1 Computer Vision Measurements

The measured crab and flight path angle, compared with the reference EKF, are presented in Figure 8.1. The mean and RMS of $\tilde{\chi}$ and $\tilde{\gamma}$ for the GTOF and GTOF(f) NO are shown in Table 8.1 and Table 8.2 respectively. Only GTOF and GTOF(f) NO are presented in the figure. This is because GTOF NO and GTOF MEKF uses the identical CV (depends on the same OF vectors, and the same inclinometer measurements), which leads to χ and γ of GTOF NO and MEKF coinciding. From the figure it is seen that GTOF(f) NO has more noise than GTOF NO in the beginning. The CV of GTOF(f) NO depends on the estimated attitude. The noise in χ and γ is therefore not a surprise, as the observer need some time to converge to the correct attitude (an incorrect attitude estimate yield incorrect feedback, which in turn yields erroneous CV measurements). However after approximately 1100 seconds, the GTOF(f) NO seem to have somewhat less noise than GTOF NO. Both the crab and flight path angle error are reduced when using attitude feedback

Table 8.1: Mean and RMS in $\tilde{\chi}$ and $\tilde{\gamma}$, given in degrees, from GTOF CV

CV	Mean Error	RMS
$\tilde{\chi}$	-1.9452	10.0278
$\tilde{\gamma}$	-1.1624	3.1848

Table 8.2: Mean and RMS in $\tilde{\chi}$ and $\tilde{\gamma}$, given in degrees, from GTOF(f) CV

CV	Mean Error	RMS
$\tilde{\chi}$	-2.5509	6.248
$\tilde{\gamma}$	-1.1024	2.7006

instead of measured attitude. This implies that using feedback of the attitude is a viable option to using inclinometer measurements.

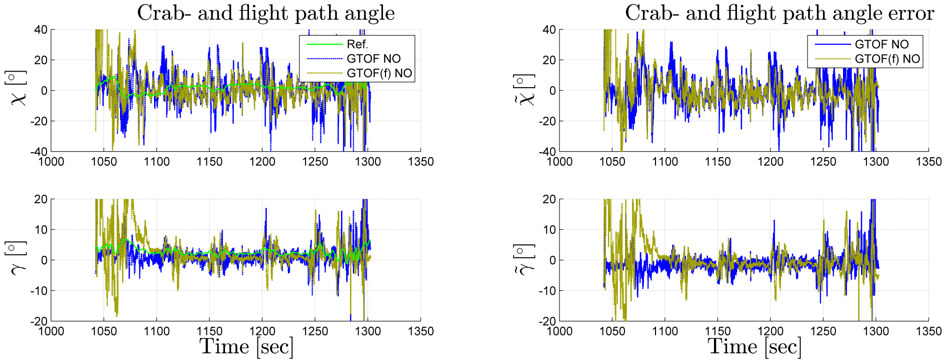


Figure 8.1: Crab and flight path angle from computer vision using ground truth optical flow computer vision. GTOF NO is the ground truth optical flow nonlinear observer, while GTOF(f) NO is the ground truth optical flow nonlinear observer with attitude feedback to computer vision subsystem.

8.1.2 Attitude Estimates

Figure 8.2 displays the estimated attitude by the GTOF NO, GTOF(f) NO and GTOF MEKF. Table 8.3, Table 8.4 and Table 8.5 presents the mean and RMS error of the attitude estimates. It is no major difference between the GTOF(f) NO, GTOF NO and the GTOF MEKF. It can be argued that the NOs have comparable results to the MEKF in estimating attitude. All observers are initiated with $\hat{R}_b^n = I_{3\times3}$, or equivalently roll, pitch and yaw equal to zero. It is seen that the attitude estimates from the GTOF and GTOF(f) NO converge faster than the GTOF MEKF. The GTOF and GTOF(f) NO use approximately 50 seconds to converge, while the GTOF MEKF uses around 70 seconds. However, after converging, the observers seem to produce very similar estimates.

Table 8.3: Mean error and RMS error in attitude, given in degrees, for GTOF NO

Attitude	Mean Error	RMS
Roll	1.5038	2.0074
Pitch	1.2737	1.5605
Yaw	-1.1617	2.1727

Table 8.4: Mean error and RMS error in attitude, given in degrees, for GTOF(f) NO

Attitude	Mean Error	RMS
Roll	1.2734	1.8144
Pitch	1.1233	1.4318
Yaw	-0.54055	1.6781

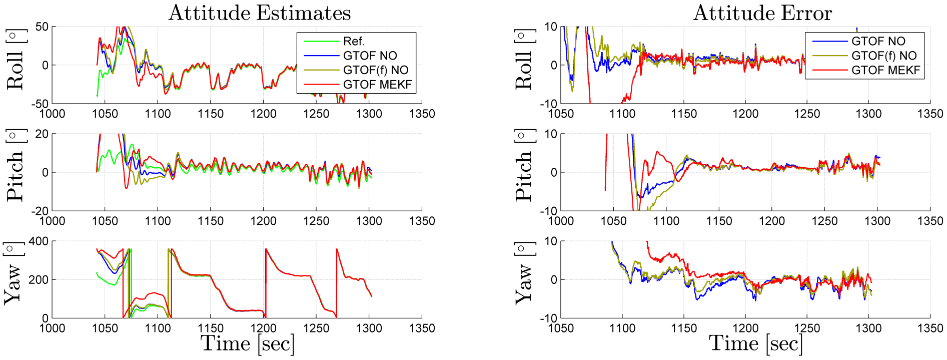


Figure 8.2: Estimated attitude by the GTOF observers.

8.1.3 Velocity and Position Estimates

Estimated velocity and position are displayed in Figure 8.3. The velocity and position of the observers are initialized with the first measurements from the GPS. The GTOF and GTOF(f) NO manage to maintain a very accurate estimate of the velocity throughout the entire flight, while the GTOF MEKF need some time to initialize. Moreover it is seen that the GTOF and GTOF(f) NO present some small spikes in the velocity estimates in the time between 1130-1150 seconds. This is due to inaccurate measurements of the velocity from the GPS. From the figure

Table 8.5: Mean error and RMS error in attitude, given in degrees, for GTOF MEKF

Attitude	Mean Error	RMS
Roll	0.68945	1.4093
Pitch	1.5012	1.7397
Yaw	0.24208	1.8321

the GTOF and GTOF(f) NO position estimate coincide. The GTOF and GTOF(f) NO appear more accurate than the GTOF MEKF in the first part of the flight. This is seen from both the North-East and the height-plot. From the height-plot it is seen that the GTOF MEKF and GTOF NO provide similar estimates from 1100 seconds to the end.

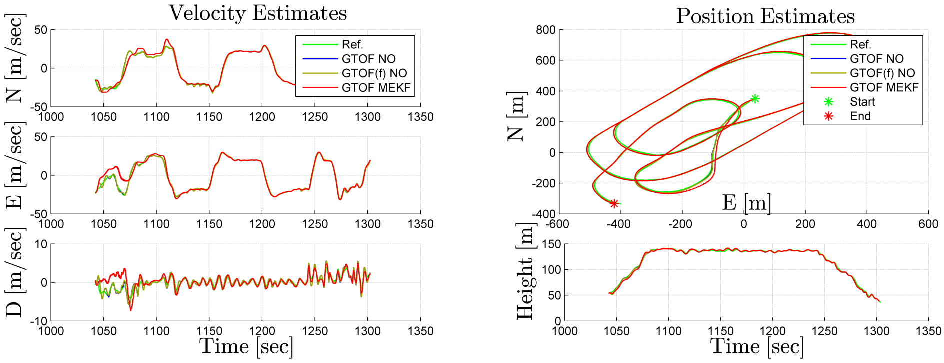


Figure 8.3: Estimated velocity and position by the GTOF observers.

8.1.4 Bias Estimates

The gyro bias estimate is shown in Figure 8.4. The bias is initialized by averaging the gyroscope before take-off. No reference of the gyro bias is available, hence it is not possible to say something about the correctness of the estimates. It is assumed that the bias is constant or slow varying, which is reflected in the estimates. The GTOF NO, GTOF(f) NO and GTOF MEKF do provide estimates with little dynamics. The estimates from GTOF MEKF, GTOF and GTOF(f) NO are very close to each other, implying that they all provide estimates that are within the correct range. Estimated accelerometer bias from the GTOF MEKF is seen in Figure 8.4. The estimated accelerometer bias converges to some constant value. No reference of the correct accelerometer bias is present, hence nothing can be said about the correctness of the estimate, although the stationary values look reasonable. The magnitude of the estimates of gyro and accelerometer bias appears to be reasonable.

8.2 Case 2: Continuous Epipolar Optical Flow Observers

This section presents the results from the case described in Section 6.4.2.

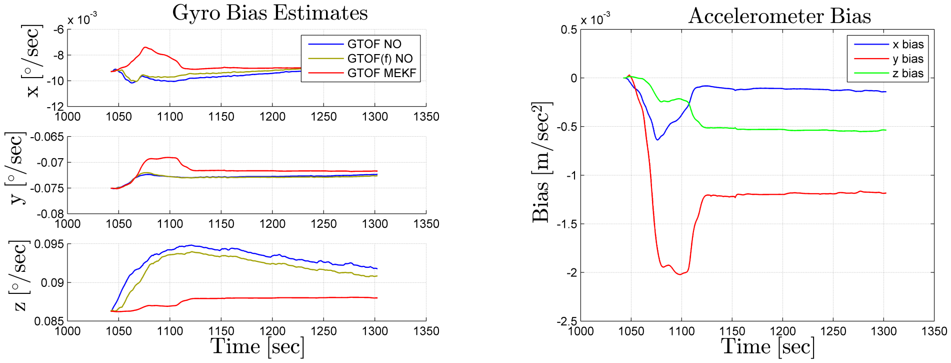


Figure 8.4: Estimated gyro bias and accelerometer bias estimated from MEKF.

Table 8.6: Mean and RMS in $\tilde{\chi}$ and $\tilde{\gamma}$, given in degrees, for CEOF NO

CV	Mean Error	RMS
$\tilde{\chi}$	-2.4691	3.7791
$\tilde{\gamma}$	-0.36728	2.6349

8.2.1 Computer Vision Measurements

The measurements from CV are shown in Figure 8.5. Moreover the RMS and mean error are presented in Table 8.6 and Table 8.7 for the CEOF NO and CEOF MEKF respectively. The CV in CEOF NO and CEOF MEKF differ in the estimated gyro bias that is used in the feedback to CV. It is seen that the CV measurements from CEOF NO and CEOF MEKF coincide, indicating that the CEOF NO and CEOF MEKF have equal estimates of the gyro bias. The crab angle error is varying around zero with values around ±15° while the flight path angle error has values between ±10°.

8.2.2 Attitude Estimate

From Figure 8.6 it is seen that the attitude estimated by CEOF NO converges to the reference attitude after approximately 50 seconds. The CEOF NO and the MEKF produce estimates very similar to the reference attitude from the Piccolo

Table 8.7: Mean and RMS in $\tilde{\chi}$ and $\tilde{\gamma}$, given in degrees, for CEOF MEKF

CV	Mean Error	RMS
$\tilde{\chi}$	-2.4722	3.7799
$\tilde{\gamma}$	-0.3656	2.6333

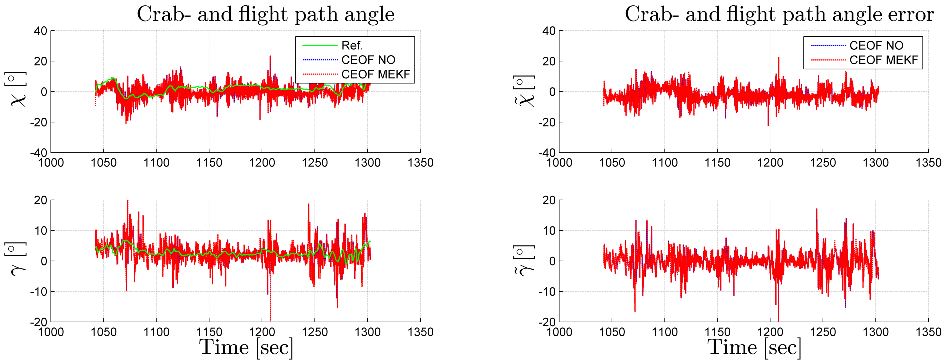


Figure 8.5: Crab and flight path angle from computer vision using continuous epipolar optical flow computer vision. The CEOF observers (NO and MEKF) use the same computer vision, causing the crab and flight path angles to coincide.

Table 8.8: Mean error and RMS error in attitude, given in degrees, for CEOF NO

Attitude	Mean Error	RMS
Roll	1.367	1.9218
Pitch	0.98635	1.3539
Yaw	-0.68797	1.7861

autopilot EKF. RMS and mean error values are seen in Table 8.8 and Table 8.9. No the RMS values of major differences are seen between the MEKF and NO. The mean errors in attitude from MEKF and NO are all below 1.5°.

8.2.3 Velocity and Position Estimates

Figure 8.7 displays the estimated velocity and position. It is seen that the CEOF NO and MEKF both produce reasonable estimates of the velocity. As for the GTOF NO, the CEOF NO has some spikes in the velocity estimates at the time between 1130-1150 seconds. This is due to spikes in the GPS velocity measurements. Moreover it is seen that the CEOF NO and MEKF have similar results in the position estimates.

Table 8.9: Mean error and RMS error in attitude, given in degrees, for CEOF MEKF

Attitude	Mean Error	RMS
Roll	0.68079	1.4078
Pitch	1.4955	1.7382
Yaw	0.33517	1.8633

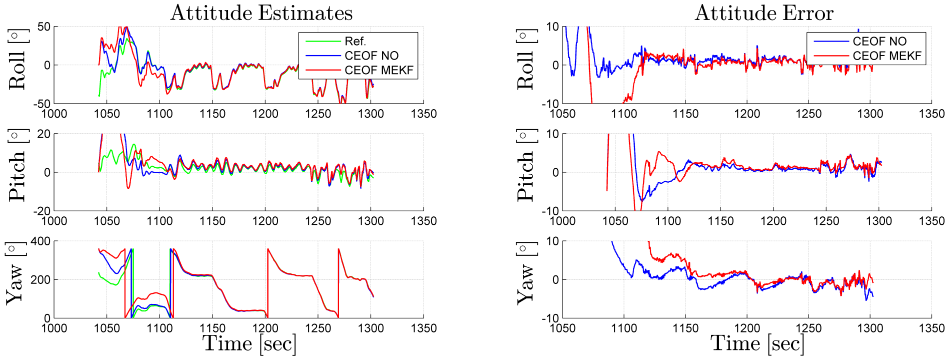


Figure 8.6: Estimated attitude from the CEOF observers

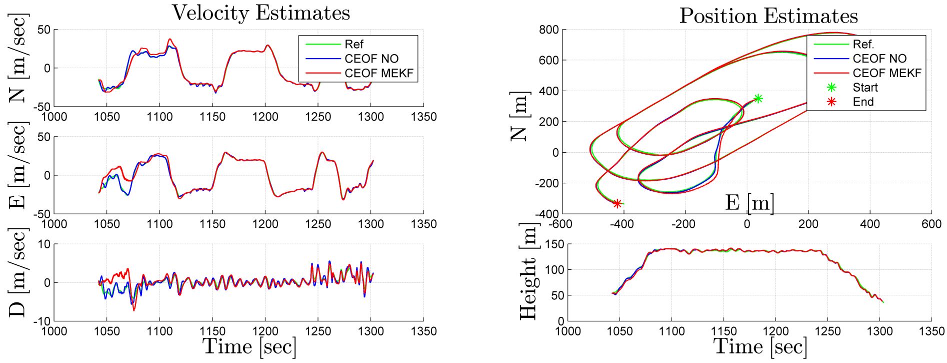


Figure 8.7: Estimated velocity and position by the CEOF observers.

8.2.4 Bias Estimates

The gyro bias is presented in Figure 8.8. The CEOF NO and MEKF estimates gyro bias that is very similar. It is difficult to say something about the validity of the bias estimate as no reference of the bias is present. The gyro bias is initialized by averaging the gyroscope before flight. It is seen that the estimated bias does not deviate much from the initial values which are assumed to be close to the correct bias. Estimated accelerometer bias from the CEOF MEKF is seen in Figure 8.8. The estimated bias converges to a constant value.

8.3 Case 3: Epipolar Optical Flow Observers

This section presents the results from the case described in Section 6.4.3.

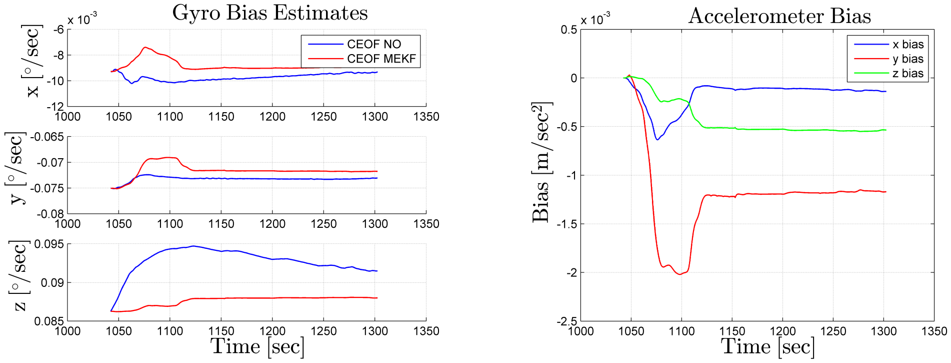


Figure 8.8: Estimated gyro and accelerometer bias by the CEOF observers.

Table 8.10: Mean and RMS in $\tilde{\chi}$ and $\tilde{\gamma}$, given in degrees, for EOF NO

CV	Mean Error	RMS
$\tilde{\chi}$	-2.5473	3.8952
$\tilde{\gamma}$	-0.55941	2.613

8.3.1 Computer Vision Measurements

The crab and flight path angle error are displayed in Figure 8.9. The crab and flight path angle error values of EOF(Δp) NO are the UAV direction of translation compared to the reference direction of translation. The reference direction of translation is found by $\Delta \boldsymbol{p}^b_{b/n} = R^b_n(\boldsymbol{p}^n_{b/n}(t_k + \Delta t) - \boldsymbol{p}^n_{b/n}(t_k))$ from the reference EKF. The RMS and mean error values of χ and γ is seen in Table 8.11,Table 8.12 and Table 8.10 for the EOF NO, EOF(Δp) NO and EOF MEKF respectively. It is seen that there is no improvement in crab and flight path angle error when using the direction of translation as reference vector, compared to the EOF NO and MEKF.

8.3.2 Attitude Estimate

From Figure 8.10 it is seen that the EOF NO, EOF(Δp) NO and the EOF MEKF produces reasonable estimates of the attitude, close to the reference attitude from

Table 8.11: Mean and RMS in $\tilde{\chi}$ and $\tilde{\gamma}$, given in degrees, for EOF(Δp) NO

CV	Mean Error	RMS
$\tilde{\chi}$	-5.5176	5.4402
$\tilde{\gamma}$	-2.1223	3.7364

Table 8.12: Mean and RMS in $\tilde{\chi}$ and $\tilde{\gamma}$, given in degrees, for EOF MEKF

CV	**Mean Error**	**RMS**
$\tilde{\chi}$	-2.5505	3.896
$\tilde{\gamma}$	-0.55781	2.6114

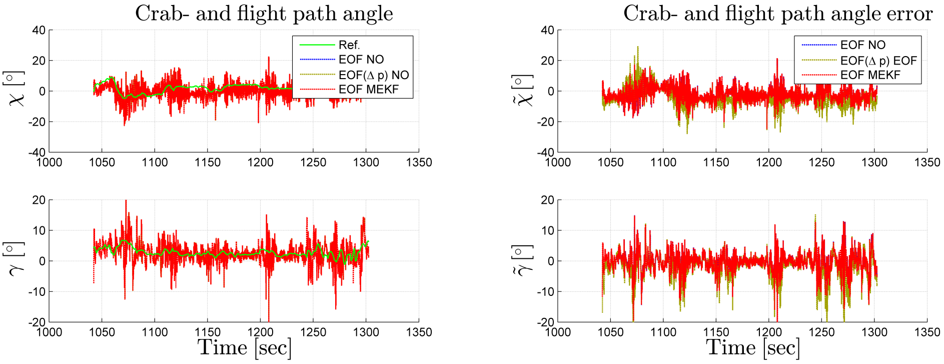


Figure 8.9: Crab and flight path angle from computer vision using epipolar optical flow computer vision. The EOF NO, EOF(Δp) NO and EOF MEKF use the same computer vision (except the estimated bias feedback from the observers to the computer vision), causing the crab and flight path angles to coincide in the plot to the left. The EOF(Δp) measurement is however compared to the change in position of the UAV, yielding an other crab and flight path error angle.

the Piccolo autopilot EKF. The RMS and mean error values are presented in Table 8.14,Table 8.15 and Table 8.13 for the EOF NO, EOF(Δp) NO and EOF MEKF respectively.

8.3.3 Velocity and Position Estimates

It is seen, from Figure 8.11, that the EOF NO, EOF(Δp) NO and EOF MEKF produce accurate estimates of the velocity. It is seen that the MEKF uses more time than EOF NO and EOF(Δp) NO to make the position estimate converge. After time 1100 seconds, both the NO and MEKF estimate the position accurately.

Table 8.13: Mean error and RMS error in attitude, given in degrees, for EOF NO

Attitude	**Mean Error**	**RMS**
Roll	1.3571	1.9207
Pitch	0.97988	1.3608
Yaw	-0.61974	1.731

Table 8.14: Mean error and RMS error in attitude, given in degrees, for EOF(Δp) NO

Attitude	Mean Error	RMS
Roll	1.349	1.915
Pitch	0.97634	1.3621
Yaw	-0.61352	1.7167

Table 8.15: Mean error and RMS error in attitude, given in degrees, for EOF MEKF

Attitude	Mean Error	RMS
Roll	0.68035	1.4076
Pitch	1.4953	1.7381
Yaw	0.33608	1.8618

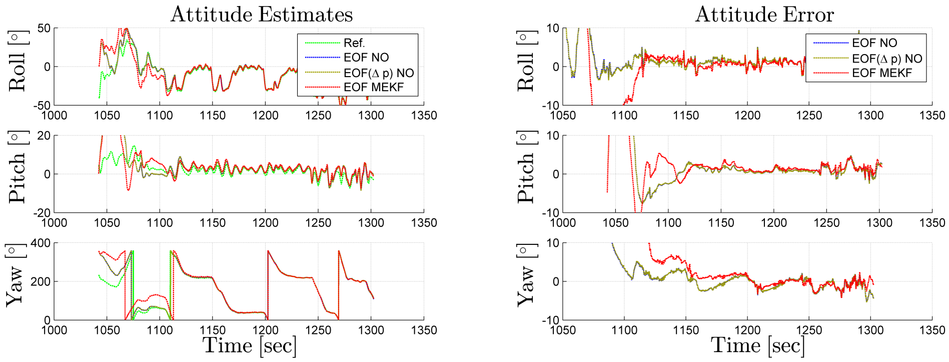


Figure 8.10: Estimated attitude and error in attitude estimate. It is seen that the estimated attitude from EOF NO and EOF(Δp) NO coincide.

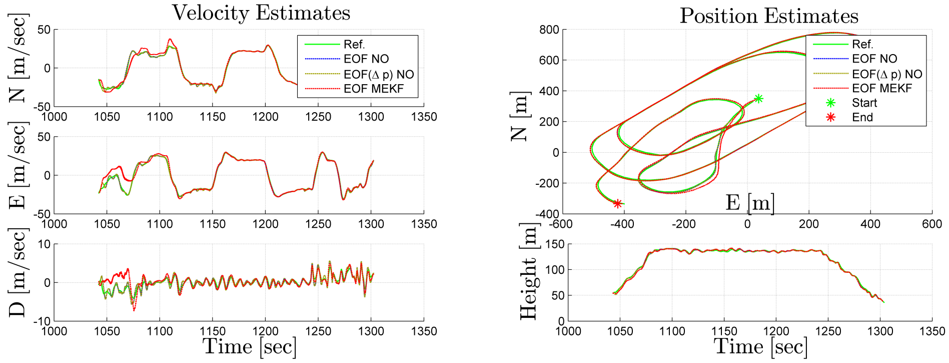


Figure 8.11: Estimated velocity. Estimated position.

8.3.4 Bias Estimates

The gyro bias estimate is displayed in Figure 8.12. Nothing can be said about the validity of the gyro bias estimates, as no reference is available. The estimates are very similar for the EOF NO, EOF(Δp) NO and MEKF. The MEKF estimates accelerometer bias. As can be seen from Figure 8.12, the estimated accelerometer bias converges to some constant value.

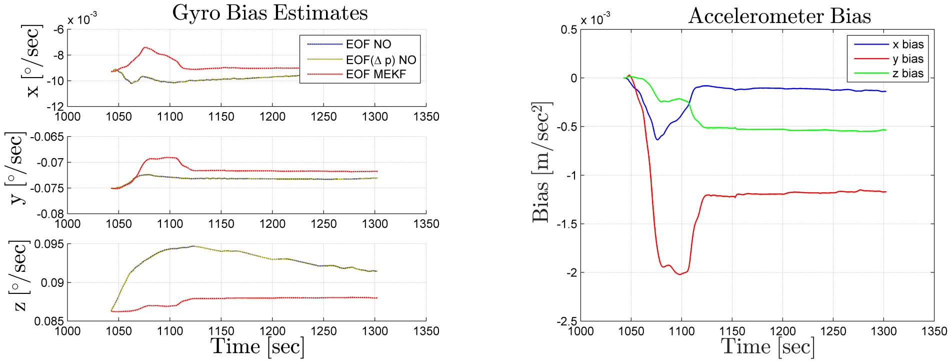


Figure 8.12: Estimated gyro and accelerometer bias. It is seen that the estimated gyro bias from EOF NO and EOF(Δp) NO coincide.

8.4 Comparing Case 1, 2 and 3

RMS errors of the attitude, position and velocity estimate, calculated from the reference EKF, from all cases are presented together in Table 8.16. The table lists the RMS values of every observer tested in the experimental results. The table is organized in two main categories, the NOs and the MEKFs. Table 8.16 also presents execution time of the observers. RMS values for the gyro- and accelerometer bias are not listed in Table 8.16 as no reference of these quantities are available.

From the attitude RMS it is noticed that the MEKF in general has lower RMS values than the NO in roll and yaw, while the NO has lower RMS in pitch. The results indicates that the NO has similar performance as the MEKF.

It is interesting to see that the proposed CV systems, CEOF and EOF, increase the overall accuracy of the observers. This is seen first and foremost from the values of the error angles $\tilde{\chi}$ and $\tilde{\gamma}$, but also in the attitude estimates. Furthermore it is seen that using attitude feedback with the GTOF CV increases the accuracy of the CV, compared to using inclinometers. Using Δp as reference vector does not improve the accuracy of the EOF observer significantly, although there is some

Table 8.16: Root Mean Squared (RMS) errors in the estimates, compared to the autopilot EKF. GTOF(f) indicates that a feedback of the attitude is introduced from the attitude estimate to the computer vision (CV) subsystem. EOF(Δp) indicates that Δp is used as reference vector. The execution time is given in milliseconds pr. iteration.

		Nonlinear Observer					MEKF		
		GTOF	**GTOF(f)**	**EOF**	**EOF(Δp)**	**CEOF**	**GTOF**	**EOF**	**CEOF**
$\mathbf{\Theta}_{b/n}$ (° RMS)	**Roll**	2.0074	1.8144	1.9207	1.915	1.9218	1.4093	1.4076	1.4078
	Pitch	1.5605	1.4318	1.3608	1.3621	1.3539	1.7397	1.7381	1.7382
	Yaw	2.1727	1.6781	1.731	1.7167	1.7861	1.8321	1.8618	1.8633
$\boldsymbol{p}^n_{b/n}$ (m RMS)	**North**	12.3018	12.3234	12.303	12.3037	12.3029	12.875	12.8796	12.8793
	East	10.9365	10.9303	10.933	10.9334	10.9327	11.2075	11.2105	11.2105
	Down	1.4334	1.4527	1.4329	1.4326	1.4327	1.8559	1.8556	1.8557
$\boldsymbol{v}^n_{b/n}$ (m/s RMS)	**North**	1.0506	1.0672	1.0696	1.0693	1.0684	2.8042	2.8016	2.8016
	East	1.2251	1.2789	1.2345	1.2359	1.2338	5.0044	5.0036	5.0037
	Down	0.71407	0.72591	0.71149	0.71163	0.71133	1.2767	1.2764	1.2765
CV (° RMS)	$\tilde{\chi}$	10.0278	6.248	3.8952	5.4402	3.7791	10.0278	3.896	3.7799
	$\tilde{\gamma}$	3.1848	2.7006	2.613	3.7364	2.6349	3.1848	2.6114	2.6333
Exec. time	(ms)	0.2100	0.2042	0.2120	0.2148	0.2121	0.8267	0.8442	0.8401

minor changes in the attitude RMS values.

From the execution time field it is seen that the NO is more computationally effective than the MEKF. The MEKF has an average execution time about 0.8-0.9 milliseconds pr. iteration, while the NO executes in about 0.2 milliseconds. The results show that the NO is four times faster than the MEKF.

Chapter 9

Simulation Results

This chapter presents the results from the simulation described in Chapter 7.

9.1 Case 4: Navigating in the Coastline

This section presents the results from the case described in Section 7.3.

9.1.1 Computer Vision Measurements

The GTOF and CEOF NOs use the normalized body-fixed linear velocity measurements from CV. This gives valuable information about how the UAV moves relative to itself. OF is used to generate these measurements. The GTOF and CEFO NOs use the same GNSS and IMU measurements. Furthermore they calculate OF vectors in the same way. They are however different in the way they utilizes these OF vectors to calculate the body-fixed linear velocity as described in Section 5.4 and Section 5.5. Figure 9.1 shows the measured normalized body-fixed linear velocities from GTOF and CEOF. The NoCV does not actually measure the velocity, but produces a pseudomeasurement implying that the UAV has only straight forward motion. This is seen from Figure 9.1, where NoCV outputs constantly $\boldsymbol{v}_{cv}^{b} = [1, 0, 0]^T$. The CEOF calculates a normalized velocity that is almost identical to the true value. This means that the CEOF succeeds in producing very accurate measurements.

As one normalizes the measurement, no information about the scale of the velocity is acquired from CV. Moreover, the normalization makes the vector measurements from CV to only have two degrees of freedom (this is due to the fact that any unit vector can be constructed from any two numbers with total norm less than or equal to one). As the CV measurements only provides information about the direction of travel, it is reasonable to compare the direction of the body-fixed velocity of the UAV. This direction is given by the crab and flight path angle. The error in measured crab and flight path angles are seen in Figure 9.1.

The crab and flight path angle errors in the measured body-fixed velocities are seen in Figure 9.1. These error angles come from erroneous measurements of the body-fixed velocity. The NoCV does not have any CV, and does therefore not measure any body-fixed velocity, but rather uses a pseudomeasurement $\boldsymbol{v}_{cv}^{b} = [1, 0, 0]^{T}$. The NoCV assumes that the UAV has only forward motion, and this in turn causes the crab and flight path error angles to increase when having some sideways or up/down motion. When the UAV flies over ocean (which is planar horizontal), the GTOF nonlinear observer has very small, or no, error angles. but when flying over the rugged non-planar terrain it produces erroneous measurement of the body-fixed velocity. The CEOF observer is unaffected by the structure of the terrain, but some increased noise is seen when the UAV is taking steep turns. This is especially seen at time 180 seconds.

9.1.2 Attitude Estimates

The attitude estimates and errors are seen in Figure 9.2. The observers are initialized with $\hat{R}_{b}^{n} = 0$. The UAV is simulated to have initially zero roll and yaw and approximately seven degrees pitch. This means that the roll and yaw angles are already assigned the correct value before the other estimates have converged. The pitch estimate of the GTOF and CEOF observer converges after approximately 10 seconds, while NoCV uses around 40 seconds to converge to the correct value.

The CEOF observer produces correct estimates throughout the entire flight, and no reduced performance is seen when flying over the rugged non-planar terrain. The CEOF observer has however a slightly more distinct deviation from the real attitude when the UAV takes a steep turn, at 180 seconds. This is seen from the attitude error in roll and pitch, and is also the case for the GTOF observer. The GTOF observer fails to produce correct estimates in yaw in the start-segment. When flying over the ocean the GTOF observer produces very accurate estimates, which is as accurate as the CEOF observer. When flying over the non-planar rugged terrain, the GTOF observer produces erroneous estimates of both roll, pitch and yaw. This implies that the GTOF observer is sensitive to bad measurements from CV. It is visible that the GTOF observer fails to produce correct attitude estimates when flying over non horizontal planar terrain. NoCV produces incorrect estimates when the UAV has crab and flight path angles, that is sideways or up/down-ways

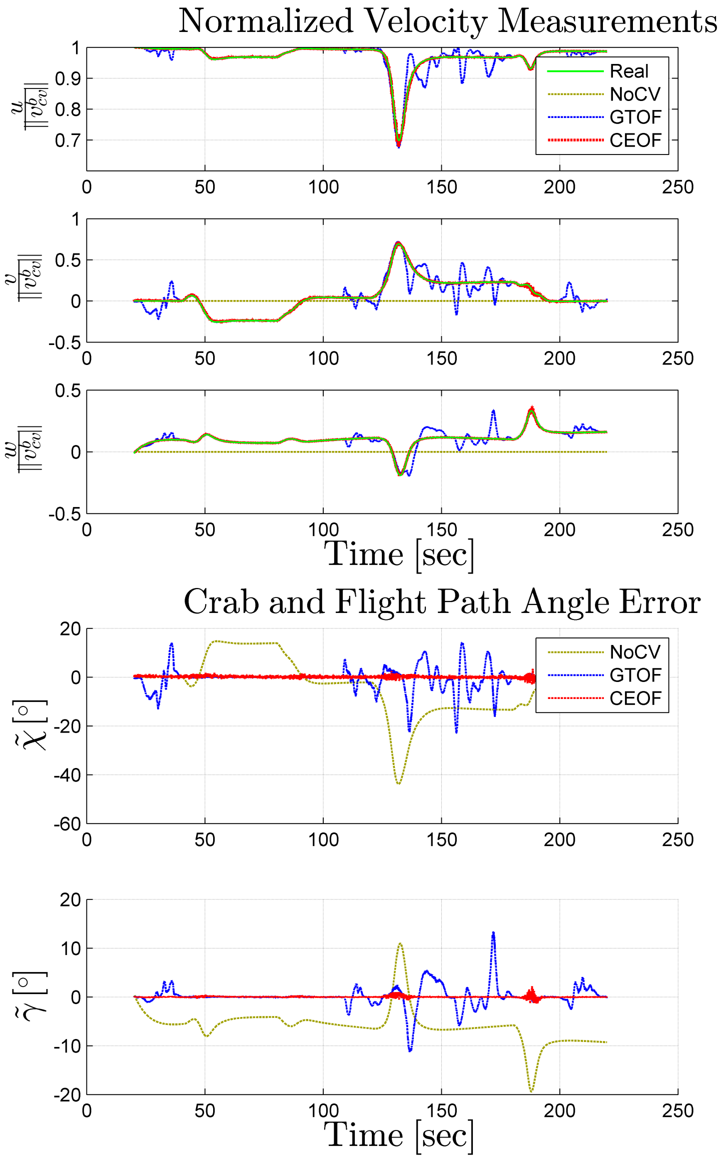


Figure 9.1: The CEOF produces very accurate measurements of the normalized linear velocity, and is identical to the true value throughout the entire flight. This is seen as the red line (CEOF) lies behind the green line (true value) the entire time. The error in measured crab and flight path angle show that the CEOF has a low noise level, the GTOF fails in the time between 110 to 220 seconds, while NoCV fails every time the UAV has som crab/flight path angle.

motion. This is a result of a pseudomeasurement deviating much from the real value. The effect is seen mostly in the Yaw, where the error is at most more than 50°. This is a result of the UAV having a significant crab angle. Also the pitch and roll are estimated incorrectly due to the lack of CV measuring the direction of the UAV.

9.1.3 Position and Velocity Estimates

The position and velocity estimates are shown in Figure 9.3. From the figure it is seen that the NoCV, GTOF and the CEOF observer produce very similar translational estimates. The estimates are all close to the real values. It is hard to say whether one observer is better than another. This is however expected, as the GNSS measurements are the most significant factor affecting the position and velocity estimates. No difference is seen in the observers when flying over the ocean or the non-planar terrain.

9.1.4 Gyro Bias Estimates

The estimated gyro bias is displayed in Figure 9.4. It is seen that the CEOF observer produces the most accurate estimates of the bias, and that convergence is reached approximately at time 100 seconds. The GTOF estimates correct bias when flying over the ocean, but when flying over non-planar terrain the GTOF observer fails in estimating correct bias. The GTOF CV measures incorrect crab and flight path angle of the UAV, and the nonlinear observer thinks that this is a correct measurement picking up some motion of the UAV. As the gyroscope does not provide measurements describing the same motion, the observer thinks that the gyro bias is the reason that the gyroscope does not measure this motion. This in turn causes the nonlinear observer to estimate not just incorrect attitude, but also incorrect gyro bias. As for the attitude, it is seen that the NoCV nonlinear observer fails to estimate the correct gyro bias. This is because the pseudomeasurement of the body-fixed linear velocity sometimes deviates much from the real value.

9.1.5 Overall Performance

Table 9.1 lists the performance of the NoCV, GTOF and CEOF nonlinear observers in means of the Root Mean Squared (RMS) error between estimates and true values. The RMS-measure is defined in equation (6.3). It is be seen that CV improves the RMS in the crab angle and flight path angle errors, $\tilde{\chi}$ and $\tilde{\gamma}$. This is as expected since the NoCV observer assumes that the UAV has only forward motion.

The GTOF and CEOF nonlinear observer utilize measurements from IMU, GNSS

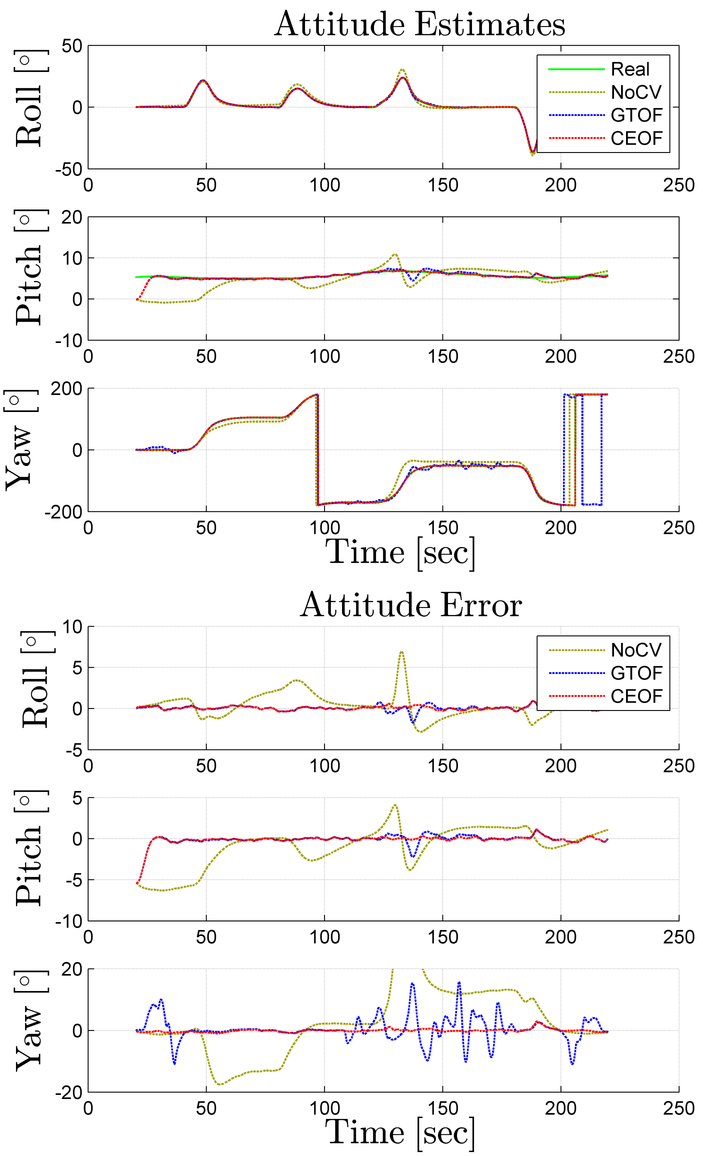


Figure 9.2: Estimated attitude. The CEOF observer estimates the attitude accurately throughout the flight. The GTOF observer produces estimates that are very accurate sometimes, while incorrect when flying over the non-planar areas. The NoCV observer fails to estimate correct attitude when the UAV have some motion that is not straight forward.

and CV. As the same IMU and GNSS measurements are used in the NoCV, GTOF and CEOF observer, it is expected that the CV errors will reflect the error in the

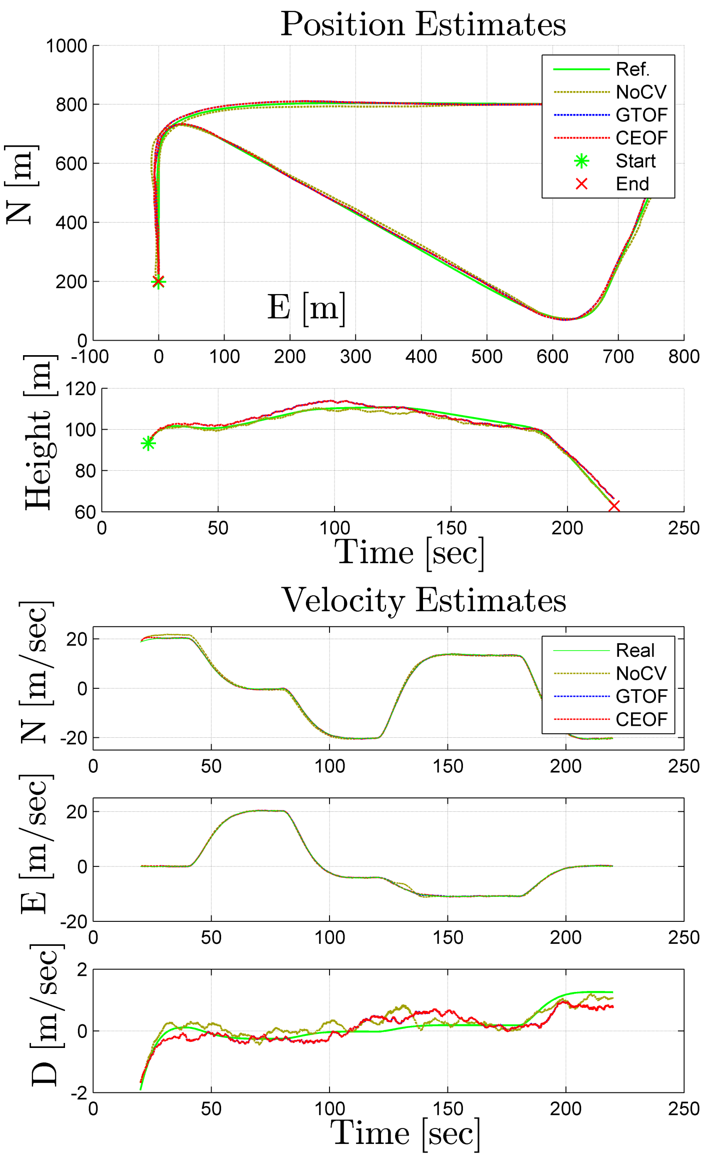


Figure 9.3: Position and velocity estimates. NoCV, GTOF and CEOF all produce translational estimates very similar to the reference.

other estimates. From the table it is seen that by introducing CV the RMS in the yaw estimate reduces from around 17° to approximately 5° with GTOF and under 1° with CEOF. Introducing CV yields a significant improvement. Moreover the RMS of the estimate of the gyro bias is reduced when introducing CV. This is seen especially in yaw-direction, where the RMS is reduced from $0.1°/\sec$ for NoCV

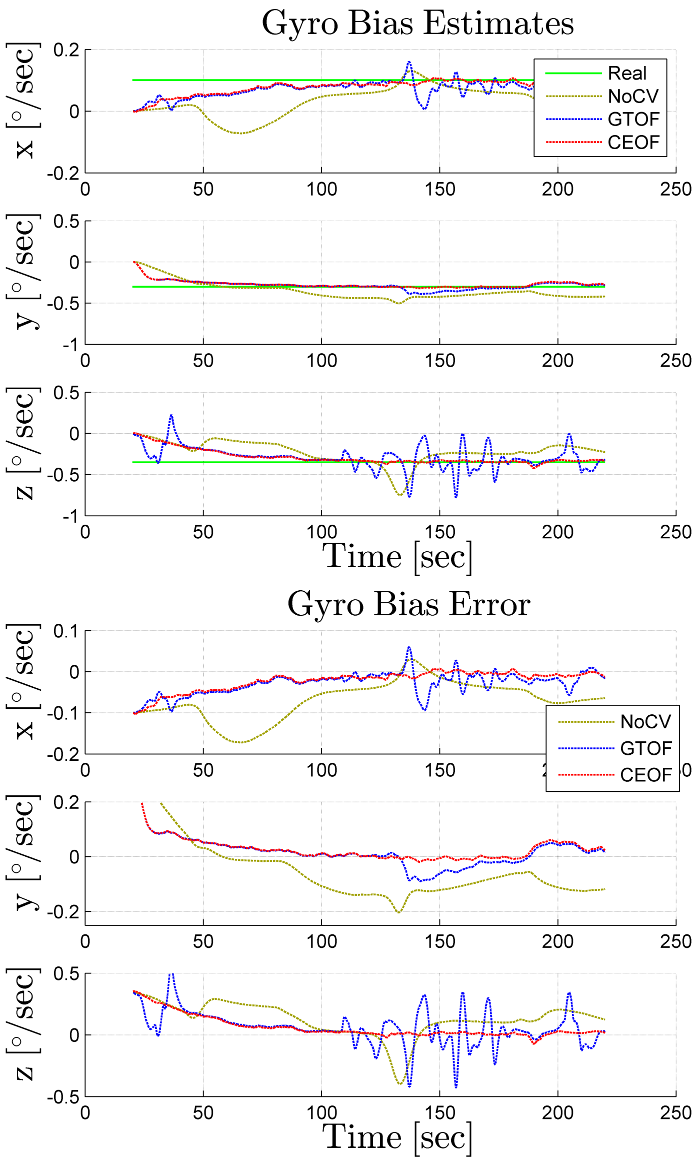


Figure 9.4: Estimated bias and error in estimate.

to approximately 0.01°/ sec for CEOF (recall that the gyro bias is in magnitude 0.1-0.3°/ sec). The position and velocity RMS are quite similar for all observers, as expected.

Table 9.1: Root Mean Squared (RMS) errors in the estimate of the states, when navigating over different terrain. NoCV denotes a non-linear observer without computer vision. GTOF denotes the ground truth optical flow observer, while CEOF denotes the continuous epipolar optical flow observer representation. The gyro bias converges after approximately 100 second, hence the RMS values of the attitude and bias is considered from 100 seconds after start. Mean is the average RMS of the attitude, position and velocity and gyro bias. $\tilde{\chi}$ and $\tilde{\gamma}$ are the crab and flight path angle error as defined in equation (6.1) and equation (6.2).

Nonlinear Observer		**NoCV**	**GTOF**	**CEOF**
$\hat{R}_b^n$ (° RMS)	**Roll**	1.5567	0.34947	0.22886
	Pitch	1.5253	0.44828	0.19555
	Yaw	16.9411	4.9635	0.48152
$\boldsymbol{p}_{b/n}^n$ (m RMS)	**North**	6.9063	5.6735	5.6326
	East	5.1357	4.8273	4.811
	Down	1.5013	2.1648	2.1641
$\boldsymbol{v}_{b/n}^n$ (m/s RMS)	**North**	0.66703	0.17651	0.17562
	East	0.32544	0.082529	0.076372
	Down	0.23738	0.21957	0.21979
$\boldsymbol{b}_{\text{gyro}}^b$ (°/s RMS)	**Roll**	0.033247	0.017695	0.003933
	Pitch	0.081939	0.032418	0.010337
	Yaw	0.10986	0.10292	0.010542
CV (° RMS)	$\tilde{\chi}$	12.1701	4.7665	0.4782
	$\tilde{\gamma}$	6.8836	2.2023	0.1772

Chapter 10

Discussion

This chapter seeks to discuss some of the findings from the experiment and the simulation. The chapter is divided in to two sections, one for the experimental part and one for the simulation.

10.1 Experiment

From the experimental results in Chapter 8 it is difficult to say something about the accuracy of the different observers. Furthermore it is difficult to state whether the MEKF or NO performs better. This is due to the fact that the accuracy of the reference is somewhat unknown. However some considerations should be highlighted.

From the results it is seen that, independently of the computer vision (CV) being used, the MEKF needs more time to converge than the NO. This might imply that the NO has better convergence properties than the MEKF, but the tuning of the observers have to be highlighted. The convergence rate of the observer is tuning dependent. The MEKF estimates could converge faster to the correct values if the initial error covariance matrix was tuned differently. As the convergence rate is tuning dependent, it is not possible to state that one of the observers have better convergence properties than the other.

Moreover there is not much difference in the different observers evaluated. Recall from the theory that the GTOF NO will fail in the presence of non-horizontal-planar terrain. The experiment is performed at an area where the terrain is very flat, and the main disadvantage with GTOF does not appear. Moreover the UAV flight path is not very acrobatic, meaning there is no significant differences between the direction of translation, and the direction of the body-fixed velocity. Therefore no significant difference between CEOF, EOF and EOF(Δp) are seen. From the $\tilde{\chi}$ and $\tilde{\gamma}$ values of EOF(Δp) NO and EOF NO it is no reason to state that using the change in position as reference vector increases the accuracy, compared to using the velocity. It is however believed that one would see some improvement with EOF(Δp) over EOF if the UAV was doing steeper turns.

Some small spikes appear in the velocity estimates of all NOs at time between 1130-1150 seconds. This is due to spikes in the velocity measurements from the GPS. That indicates that the tuning of the NO might trust the velocity measurement to much.

10.2 Simulation

From the Figures 9.1-9.4 and the numerical results from Table 9.1 in Chapter 9, it can be seen that CV improves the performance of the nonlinear observer. It is seen that CV improves the RMS in the crab angle and flight path angle errors, $\tilde{\chi}$ and $\tilde{\gamma}$. This is expected since the NoCV observer assumes that the UAV has only forward motion. The GTOF and CEOF nonlinear observer utilizes measurements from IMU, GNSS and CV. As the same IMU and GNSS measurements are used in the NoCV, GTOF and CEOF observer, it is expected that the CV errors will reflect the error in the other estimates. The most significant differences is seen in the attitude and gyro bias estimates.

It is seen that the NoCV fails to produce correct pitch, yaw and gyro bias estimates. The reason why the pitch estimate is erroneous is because the NO uses both accelerometer-measurements and the measurement of body-fixed velocity to determine the pitch. Recall that the NoCV NO does not actually measure the body-fixed velocity, but rather approximating it by assuming that the UAV has only forward motion and using the pseudomeasurement $\boldsymbol{v}_{cv}^{b} = [1, 0, 0]^T$. This is a good approximation when the UAV has small or no crab/flight path angles, but is rather poor when the UAV have maneouvers that violate this. The accelerometer measurements are accurate, but from the flight path angle errors it is seen that the body-fixed velocity pseudomeasurement claims that the UAV has no downwards velocity. The fact is that the UAV has a positive pitch, but the body-fixed velocity pseudomeasurement implies that the pitch of the UAV is zero. This means that the CV inputs incorrect information to the observer, which causes the observer to produce wrong estimates.

An interesting phenomenon occurs when the UAV is taking steep turns, and is especially seen when the UAV is taking the steep turn at 180 seconds. During turns, the measured body-fixed velocity from both GTOF and CEOF appear more noisy. This affects the CEOF NO the most, but the same phenomenon is present for the GTOF. The reason for this will now be explained and justified. The GTOF CV is aided by an inclinometer, while the CEOF observer is aided by a gyroscope. As these sensors are independent of each other (although they both measures angular quantities) it is not reasonable to think that the problem is caused by noise on these sensors. The same OF vectors, on the other hand, are used by both GTOF and CEOF CV, and is the only common input to the CEOF and GTOF observer. Recall, from Section 3.2, that the continuous OF, $\dot{\boldsymbol{x}}^m$, is approximated by the discrete OF $\frac{d\boldsymbol{x}^m}{dt}$. This is because any OF algorithm computes the discrete OF and not the continuous OF. Discrete optical flow is a good approximation when the motion of a feature in the image plane is small, but when the motion is large the quality of this approximation reduces. The problem seems therefore to be in the restriction of the camera and the OF algorithms. The fact that the camera is a discrete sensor (it does not take images with infinite frame rate) is causing the body-fixed velocity measurements to be noisy.

From the results, it is clear that the GTOF nonlinear observer has problems when flying over non-planar terrain. This is due to the assumption of all points lying in a horizontal-plane. From simulation results, not presented in this thesis, it is seen that GTOF works with horizontal-planar terrain independent of the measured distance to the ground. When using CV for attitude estimation, only the normalized body-fixed velocity is used in the nonlinear observer. The distance to the ground plane does not affect the normalized velocity measurements from GTOF. The scaled distance to any feature is recovered from the attitude and altitude of the UAV, which implies two things: 1) If the terrain is not planar it is impossible to say something about the distance to all features, and 2) If the plane is not horizontal then the inclinometer does not actually measure the attitude of the UAV relative to the terrain being filmed by the camera. This means that 1) non-planar terrain will cause trouble, and 2) even planar terrain, which is not aligned with the horizon, will cause GTOF to fail. The GTOF will however work with terrain that is horizontal-planar independent of distance to the ground.

Some considerations about the simulation should be mentioned. No real physics are taken into account when simulating the UAV path, e.g. lift is not dependent of airspeed in the simulator. This means that the path might be unrealistic. However, as far as the author can conclude this will not affect the validity of the evaluation of the NO performance. The conducted simulation is done with very little noise on the OF vectors. It would have been interesting to see the effect on the GTOF and CEOF NO when increasing the noise level on the OF vectors. Moreover a statistical analysis should be done to see the effect of how noise in OF vectors, gyroscope and inclinometer affects the body-fixed velocity measurements from CV. This is however considered outside the scope of this thesis.

Part V

Closing Remarks

Chapter 11

Conclusion

In this thesis different variants of a nonlinear observer (NO) and the multiplicative extended Kalman filter (MEKF) have been implemented and tested on both simulated and real world data. Furthermore two new nonlinear observer representations have been proposed, namely the continuous epipolar optical flow (CEOF) NO, and the epipolar optical flow (EOF) NO. This chapter will conclude the work and summarize the most important findings.

11.1 Overview

Experimental data have been collected in a UAV test flight. The images from the flight test have been processed with optical flow (OF) algorithms to generate information that can be used by computer vision (CV). The sensor data have been used to test and compare different observers, namely different variants of the NO and different variants of the MEKF. The experimental data have been used to compare the performance of the NO with the MEKF.

Simulated data was used to show how the different CV subsystem affects the performance of the observers, in the presence of more rugged and elevated terrain. The CEOF and GTOF NO were compared to a NO without CV assuming no crab and flight path angle. It was seen that utilizing CV in the NO improves the accuracy of the observer.

The Continuous Epipolar Optical Flow Nonlinear Observer (CEOF NO) and Epipolar Optical Flow Nonlinear Observer (EOF NO) have been proposed. These NOs have proven to be a robust option to the Ground Truth Optical Flow (GTOF) NO presented in Fusini et al. (2014). It have been proved that the proposed NOs have the same stability properties as the GTOF NO, namely USGES in the origin of the error dynamics.

11.2 Findings

The experimental results showed that the NO has comparable results to the MEKF when estimating the attitude, position and velocity of a fixed-wing UAV. It is however hard to say something about the accuracy of the observers as the accuracy of the reference is unknown. However as the reference is calculated by an extended Kalman filter, it is reason to believe that the reference is more or less correct. It has been seen that the NO is in general four times less computationally demanding than the MEKF. Furthermore negligible differences are seen in the computation time of the GTOF NO and the proposed CEOF NO and EOF NO.

From the simulation results it was shown that the CEOF NO yield more accurate estimates of attitude and gyro bias than the GTOF NO and a NO without CV. The CEOF NO demonstrated to be independent of the terrain. The GTOF NO showed comparable performance to the CEOF NO when flying over planar horizontal terrain, but flying over rugged terrain caused the GTOF NO to produce incorrect estimates. It is seen that the GTOF NO does not perform well when flying over terrain that is not horizontal planar. The GTOF NO assumes that the terrain being filmed is flat and horizontal. This assumption limits the validity of the GTOF NO. Although ocean and some agricultural fields are completely flat, the assumption of flat horizontal terrain is a fundamental limitation of the GTOF NO. The main advantage of the GTOF NO versus the CEOF NO is that the GTOF NO preserves the scale of the body-fixed velocity measurement, which in turn can be used in e.g dead reckoning. Moreover it has been argued for not needing to have an altimeter to measure the distance to the ground if only using the normalized body-fixed velocity.

It has been demonstrated that the CEOF NO is robust and independent of the structure being filmed, yielding accurate estimates of the states. An article on experimental validation of the GTOF observer has been written and accepted to the ICUAS'15 conference. A second paper presenting and comparing the proposed CEOF NO with the GTOF NO and a NO without CV has been written and submitted to AIAA SciTech 2016.

Chapter 12

Future Work

This chapter is going to discuss the most important factors for future work with respect to the findings in this report.

A stability proof of the CEOF NO with constant gyro bias in feedback to CV is presented in this thesis, but the stability properties of the CEOF NO is not analysed when having a varying gyro bias in the feedback. A stability proof including varying gyro bias in feedback should be the next step in the analysis of the CEOF NO.

A new UAV experiment should be performed at lower altitude or at a location with more rugged terrain. This would give experimental data that can be used to validate the simulated results from this thesis. Experimental data can be used to show whether the CEOF NO performs better than the GTOF NO with real world data.

A discrete Lyapunov analysis should be done for the observer. The camera is a discrete sensor (it measures the discrete optical flow) and any computer implementation is discrete, arguing for a discrete representation of the nonlinear observer. It should be investigated how increasing the frame rate of the camera affects the performance of the observers. A discrete observer representation argues for the discrete Epipolar Optical Flow Nonlinear Observer as described in Section 5.6.

More sensors could be added to the nonlinear observer. The payload module already contains an altimeter and an inclinometer. The altimeter could be used to aid the down position estimate of the observer, while the inclinometer can be used to aid the attitude estimate.

The simulator implemented in this thesis should be expanded and generalized.

The simulator is a powerful tool for vision-based navigation, making it possible to simulate a UAV with camera flying over any terrain. The simulator makes it possible to get realistic IMU, GNSS and optical flow measurements without having to perform a real UAV test flight. The camera simulator provides realistic measures of optical flow. The UAV flight path and kinematics path of the simulator should be improved by adding a more realistic model of the UAV.

The GTOF NO has a big advantage of actually preserving the scale of the velocity measurements. It should be investigated whether it is possible to modify CEOF NO to enable the altimeter to measure velocity with the scale preserved. Moreover the GTOF NO can be expanded to combine GTOF with simultaneous localization and mapping (SLAM) in order to recover the depth in the images. Then the assumption of flat horizontal terrain can be omitted.

Part VI

Appendices

Appendix A

Submitted Paper to AIAA SciTech 2016

The following paper was submitted for The 2016 American Institute of Aeronautics and Astronautics Science and Technology Forum and Exposition (AIAA SciTech) in San Diego, California, US. Any decision on acceptance/rejection is not yet taken, but is expected before 25th of August 2015.

A Vision-aided Nonlinear Observer for Fixed-wing UAV Navigation

Jesper Hosen*, Håkon H. Helgesen*, Lorenzo Fusini†, Thor I. Fossen† and Tor A. Johansen†
*Norwegian University of Science and Technology
Department of Engineering Cybernetics
Trondheim, Norway
E-mail: {jesperhosen, haakon.helgesen}@gmail.com
†Norwegian University of Science and Technology
Centre for Autonomous Operations and Systems
Trondheim, Norway
E-mail: {lorenzo.fusini, thor.fossen, tor.arne.johansen}@itk.ntnu.no

Abstract—**This paper presents a vision-aided uniformly semi-globally exponentially stable (USGES) nonlinear observer for estimation of attitude, gyro bias, position, velocity and specific force of a fixed-wing Unmanned Aerial Vehicle (UAV). The nonlinear observer uses measurements from an Inertial Measurement Unit (IMU), a Global Navigation Satellite System (GNSS) receiver, and a video camera. This paper present a nonlinear observer representation with a computer vision (CV) system without any assumptions related to the depth in the images and the structure of the terrain being recorded. The CV utilizes a monocular camera and the continuous epipolar constraint to calculate body-fixed linear velocity. The observer is named a Continuous Epipolar Optical Flow (CEOF) nonlinear observer. Experimental data from a UAV test flight and simulated data are presented showing that the CEOF nonlinear observer has robust performance. Experimental results are compared with an Extended Kalman Filter (EKF) and illustrate that the estimates of the states converges accurately to the correct values. Results show that using the proposed CV in addition to IMU and GNSS improves the accuracy of the estimates. The CV provides accurate information about the direction of travel of the UAV, which improves the attitude and gyro bias estimate.**

I. Introduction

The use of Unmanned Aerial Vehicles (UAV) has in the last decade gained an increasingly interest, and already plays a major role in military use. The field of applications for UAVs will grow even more in the future, and the demands for robustness, safety and reliability are considered to be crucial. Robust navigation is one of the most important parts when working with UAVs. A challenge in navigation systems is to maintain accurate estimates of the states with low-cost measurement units. The output of such low-cost sensors are typically contaminated by noise and bias. As it is desirable to have low energy consumption on UAVs, it is necessary to find light weight navigation systems with good performance. The Kalman filter has been the preferred filter algorithm, but in recent years nonlinear observers, like the nonlinear complementary filter, have gained increased attention [1]–[6].

The use of cameras for navigational purposes is expected to grow quickly since video cameras are lightweight, energy efficient and the prices are constantly decreasing. As magnetometers are sensitive to disturbances, such as electromagnetic fields [7], cameras might be a good alternative or complementary to the magnetometer. The camera images can be used to output the body-fixed velocity of a UAV [8], but depend on favourable atmospheric conditions, light and detection of visual stationary features.

Computer Vision and Optical flow (OF) have been used for different applications in UAV navigation including indoor maneuvering [9], [10], linear and angular velocity estimation [8], [11], [12] and obstacle avoidance [9], [13]–[17], as well as height above the ground estimation in [18]. [19], [20] uses OF in landing assistance for UAVs without external sensor inputs. OF from a single camera is used in [21], [22] to estimate body axes angular rates of an aircraft as well as wind-axes angles. [12], [23], [24] have used OF as input in Kalman filter-based navigation systems, fusing OF measurements with acceleration and angular velocity measurements. [25], [26] have used camera as sensor for navigating in GPS-denied environments.

Attitude estimation has received significant attention as a stand-alone problem [1], [27]–[35]. In addition, other researchers have integrated Inertial Navigation System (INS), magnetometer/compass and GNSS to estimate the navigation states of a vehicle. [4] expanded the vector-based observer proposed by [1] and [32] to include GNSS velocity measurements. [27] and [28] built globally exponentially stable (GES) attitude estimators based on multiple time-varying reference vectors or a single persistently exiting vector. A similar observer was developed in [5], [36] to include also gyro bias and GNSS integration. [3] extended [36] to use linear velocity and specific force as reference vectors. [3] proved that feedback of estimated NED velocity and specific force in NED from the translational motion observer to the attitude observer, yield USGES in the origin of the error dynamics.

In this paper the observer presented in [3] is denoted as Ground Truth Optical Flow (GTOF) nonlinear observer. By assuming known distance to every feature in the camera image, the body-fixed velocity was recovered from the relationship between ego-motion and theoretical optical flow. This relationship is called the GTOF relationship between velocity and OF. The distance to every point was recovered by assuming flat horizontal terrain coinciding with NED, measured distance to the terrain by a laser altimeter and measured attitude of the UAV relative to NED by an inclinometer. The assumption of flat and horizontal terrain will cause the CV in the GTOF nonlinear observer to produce erroneous velocity measurements in the case of flying over rugged terrain. Therefore it is desirable to exchange the CV of the GTOF nonlinear observer with a CV system with no requirement of flat horizontal terrain.

Optical flow (OF) describes how objects in an image plane moves between two consecutive images. The motion in the image plane is caused by relative motion between the camera and the visual features being detected. In the simplest case it could be understood as the pixel displacement of a single feature between two successive images. The OF can be represented as multiple vectors describing the change in the image plane in time. Several methods exists for determining the OF of a series of images [37]–[40].

A camera fixed to a UAV can be used to recover the motion of the vehicle relative to the scene. An effective principle for recovering ego-motion of a camera is epipolar geometry. Epipolar geometry has been applied in e.g. navigation, landing and collision avoidance [12], [23], [41]–[45]. [46] presented the epipolar constraint in the continuous case. [47] and [48] have used the continuous epipolar constraint to recover the velocity of a UAV.

In this paper OF vectors together with the continuous epipolar constraint [46] are used to calculate the normalized body-fixed velocity of the UAV, and fed into the nonlinear observer as a reference vector. The use of the continuous epipolar constraint eliminates the dependency on the depth in the image. In practice this means that prior information about the distance and structure of the terrain are not required any more. Thus the observer is applicable when flying over any terrain.

A. Contribution of this Paper

This paper presents a more robust CV subsystem for the nonlinear observer from [3]. In [3] the ground truth optical flow (GTOF) relationship between motion and OF were used to recover the ego-motion of the UAV. A fundamental restriction from [3] was that the distance to every feature corresponding to an OF vector must be known in order to calculate the body-fixed linear velocity. The CV in this paper utilizes epipolar geometry [49] and only depends on the angular velocity of the UAV. Furthermore it works without knowing the distance to the features in the image. To the authors knowledge, this is the first time the continuous epipolar constraint has been employed in a USGES nonlinear observer.

Experimental and simulated results show that the proposed CEOF observer has comparable performance with the GTOF observer from [3] when flying over flat horizontal terrain. Simulations show that the proposed CEOF observer is structure independent, and that it outperforms the GTOF observer when flying above rugged and elevated terrain. Moreover, results show that using CV increases the accuracy of the estimates, compared to using only IMU and GNSS measurements. This is particularly clear in the attitude, as CV provides information about the direction of the body-fixed velocity. A pure IMU and GNSS approach assumes zero crab and flight path angle, and thus looses important information about the attitude. The experimental results are compared to an EKF, while the simulated results are compared to the known reference. The results imply that the CEOF observer is a robust option to the GTOF nonlinear observer.

The last contribution is a stability proof showing that the CEOF observer has the same stability properties as the GTOF observer, namely a USGES equilibrium point at the origin of the error dynamics.

II. Notation and Preliminaries

Matrices and vectors are represented by uppercase and lowercase letters respectively. X^{-1} and X^{+} denote the inverse and the pseudoinverse of a matrix respectively, X^T the transpose of a matrix or vector, $\hat{X}$ the estimated value of X, and $\tilde{X} = X - \hat{X}$ the estimation error. $\| \cdot \|$ denotes the Euclidean norm, $I_{n\times n}$ the identity matrix of order n, and $0_{m\times n}$ the $m \times n$ matrix of zeros. A vector $x = [x_1, x_2, x_3]^T \in \mathcal{R}^3$ is represented in homogeneous coordinates as $\underline{x} = [x_1, x_2, x_3, 1]^T$. The function sat$(\cdot)$ performs a component-wise saturation of its vector or matrix argument to the interval $[-1, 1]$. The operator $[x]_\times$ transforms the vector x into the skew-symmetric matrix

$$[x]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

The inverse operation is denoted as vex$(\cdot)$, such that vex$([x]_\times) = x$. The determinant of a matrix A is denoted $\det(A)$. The skew symmetric part of a square matrix A is obtained by the operator $\mathbb{P}_a(A) = \frac{1}{2}(A - A^T)$.

The North-East-Down, camera- and the body-fixed reference frames are used in this paper as shown in Fig. 1: the body-fixed frame are denoted {B} and the North-East-Down (NED) frame denoted {N} (Earth-fixed, considered inertial), while the camera frame is denoted {C}. The rotation from {B} to {N} is represented by the matrix $R_b^n \in SO(3)$, with $SO(3)$ representing the Special Orthogonal group. The image plane is denoted {M}. {B} and {C} are assumed to be aligned, ie. the camera is strapped to the body.

A vector decomposed in {B} and {N} has superscript ^{b} and ^{n} respectively. The subscript of a vector indicates which frame is measured relative to what. For instance $p^n_{b/n}$ is the position of {B} relative to {N} expressed in {N}. The camera location w.r.t. {N} is described by $c^n = [c^n_x, c^n_y, c^n_z]^T$. A point in the environment expressed w.r.t. {N} is $p^n = [x^n, y^n, z^n]^T$. The same point expressed in {C} is $p^c = [x^c, y^c, z^c]^T$. It will also be assumed that every point is fixed w.r.t. {N}. The Greek letters ϕ, θ, and ψ represent the roll, pitch, and yaw angles respectively, defined according to the zyx convention for principal rotations [6], and they are collected in the vector $\Theta_{b/n} = [\phi, \theta, \psi]^T$. A 2-D camera image has coordinates $x^m = [r, s]^T$, aligned with the y^b- and x^b-axis respectively (see Fig. 3). The corresponding homogeneous image coordinate is denoted $\underline{x}^m = [r, s, 1]^T$. The derivative $[\dot{r}, \dot{s}]^T$ of the image coordinates is the OF. The subscript $_{cv}$ indicates a quantity evaluated by means of the computer vision, $_{\text{imu}}$ indicates a quantity measured by the IMU, while $_{\text{GPS}}$ indicates that the quantity is measured by the GNSS.

A. Measurements and Sensors

The observer is designed to take use of a IMU, a GPS receiver and a video camera, providing the following measurements:

- *GPS*: NED position p^n and NED velocity v^n.
- *IMU*: biased angular velocity $\omega^b_{\text{imu}} = \omega^b_{b/n} + b^b_{\text{gyro}}$, where b^b_{gyro} represents the gyro bias, and specific force $f^b_{\text{imu}} = f^b_{b/n}$.
- *Camera*: 2-D projections $x^m = [r, s]^T$ onto the image plane {M} of points $[x^n, y^n, z^n]^T$ in {N}.

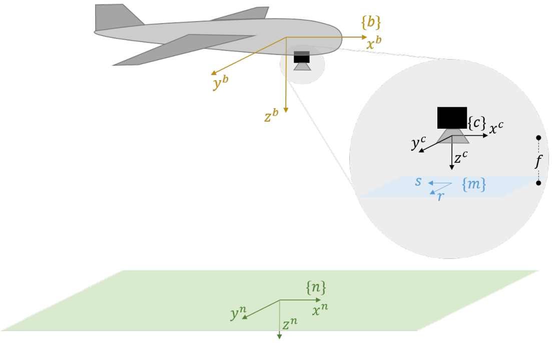


Fig. 1. Body frame is denoted {B}, camera frame is denoted {C}, and NED frame is denote {N}. Points in the terrain are projected by the pinhole camera model onto the image plane {M}, as illustrated by the blue plane.

Detailed information on the actual sensors employed in the experiment is presented in Section V.

III. Computer Vision

The observer presented in Section IV depends on body-fixed velocity measurements from the onboard camera. These measurements are generated through OF, therefore it is necessary to compute the OF vectors for consecutive images before these vectors are transformed to velocity measurements. The OF calculation and the transformation are presented in the forthcoming section.

A. *Optical flow computation*

There exist several methods for computing OF. For the experiment presented in Section V two specific methods are chosen. The first one is SIFT [39] which provided the overall best performance in [8]. The second method is a region matching-based method [8], namely template matching utilising cross-correlation [50]. SIFT uses a feature-based approach to compute OF. A set of features are extracted from two consecutive images with a feature detector. The detected features are then matched together to find common features in successive images. An OF vector is created from the displacement of each feature. The total number of such vectors in each image depends on the number of features detected and successfully matched.

It is desired to make sure that the OF algorithm produces at least two OF vectors to calculate the body-fixed velocity. It is not possible to guarantee a given number of vectors with SIFT since homogeneous environments, like snow or the ocean, increase the difficulty of finding distinct features. Therefore the OF vectors created by SIFT are combined with OF vectors from template matching [51]. The displacement of twelve templates, created symmetrically across the images, are used to find twelve OF vectors. Template matches below a given threshold are discarded and the corresponding OF vectors removed. Unreliable matches can occur in case of homogeneous terrain, changes in brightness or simply when the area covered by the template has disappeared from the image in the time between the capture of images.

The combination of two individual OF methods increases the probability of having OF vectors distributed across the whole image, as well as maintaining a high number of OF vectors. An example of OF vectors computed with SIFT and template matching from UAV test flights is displayed in Fig. 2.

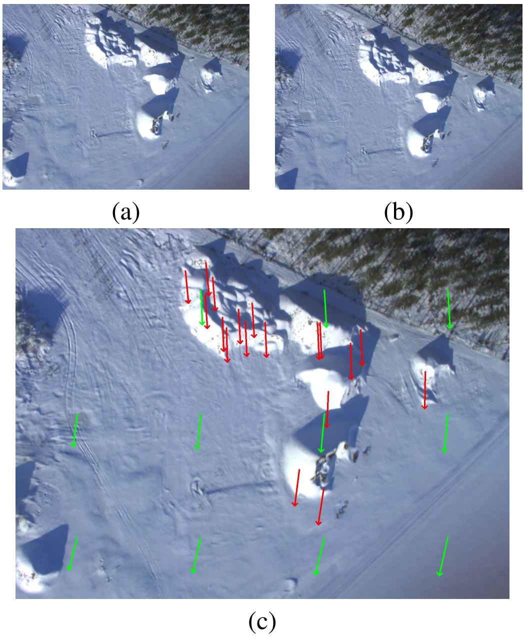
In case of mismatches, both methods create erroneous OF vectors. It is desired to locate and remove these vectors. Therefore a simple outlier detector is implemented before the vectors are used to calculate body-fixed velocities. The outlier detector utilizes a histogram to find the vectors that deviates from the mean with respect to direction and magnitude.

B. *Transformation from optical flow to velocity*

For the OF computations to be useful in the observer a transformation to body-fixed velocity is necessary. The transformation is motivated by the continuous epipolar constraint and the pinhole camera model [52]. The camera-fixed coordinate system, {C}, is related to {N} as illustrated in Fig. 3. The focal point of the camera is for simplicity assumed to coincide with the origin of {B}. A point p in the terrain is projected from {C} to {M} by the pinhole camera model by

$$\underline{x}^m = \frac{1}{z^c} K p^c \tag{1}$$

where $\underline{x}^m$ is the homogeneous image coordinate and K is a projection matrix mapping points in the



(a)

(b)

(c)

Fig. 2. a) Image captured at time t_0. b) Image captured at time $t_0 + \Delta t$. c) Optical flow vectors between image a) and b), generated by SIFT (red) and Template Matching (green).

camera frame to the image plane. It is defined as

$$K = \begin{bmatrix} 0 & f & 0 \\ -f & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where f is the focal length of the camera. The focal length of a camera can be verified by the computer vision toolbox in Matlab. The same toolbox can be used to estimate coefficients describing the distortion of the camera. These coefficients can be used to generate undistorted images. For the rest of this paper, it is assumed that the distortion is insignificant.

u^c is defined as the back-projected point lying on the projection ray between the origin of $\{\mathrm{C}\}$ and p^c with unity z-component

$$u^c = K^{-1}\underline{x}^m \tag{3}$$

Epipolar geometry [49] relates the motion of the camera frame with the motion in the image plane independent of the distance to the scene and the structure being recorded. By assuming that all

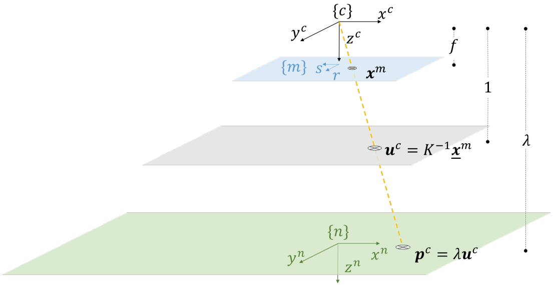

{c}
x^c
y^c
z^c
f
{m}
s
r
$\boldsymbol{x}^m$
1
$\boldsymbol{u}^c = K^{-1}\underline{\boldsymbol{x}}^m$
λ
{n}
y^n
x^n
z^n
$\boldsymbol{p}^c = \lambda \boldsymbol{u}^c$

Fig. 3. Pinhole camera model. The camera frame is denoted $\{\mathrm{C}\}$, image plane is illustrated in blue and denoted $\{\mathrm{M}\}$ and NED frame is illustrated in green and denoted $\{\mathrm{N}\}$. The gray plane is called the back projected plane. The back projected plane is located at unit length away from the camera frame in camera z-direction.

matched features are at rest w.r.t $\{\mathrm{N}\}$, the continuous epipolar constraint [46] can be expressed as

$$\left(\dot{u}^{cT} + u^{cT}\left[\omega^c_{c/n}\right]_\times^T\right)\left(v^c_{c/n} \times u^c\right) = 0 \tag{4}$$

where $\omega^c_{c/n}$ and $v^c_{c/n} = [v_x, v_y, v_z]^T$ are the angular and linear velocity of the camera relative to $\{\mathrm{N}\}$ expressed in $\{\mathrm{C}\}$, respectively. Note that the epipolar geometry has an inherited sign ambiguity due to the fact that the scale is not preserved. This means that it is only possible to determine the body-fixed velocity up to scale.

Using now the properties of a triple product [53], (4) can be rewritten as

$$\begin{aligned} &{v^c_{c/n}}^T\left(u^c \times \left(\dot{u}^{cT} + u^{cT}\left[\omega^c_{c/n}\right]_\times^T\right)^T\right) \\ &= {v^c_{c/n}}^T\left(u^c \times \left(\dot{u}^{cT} + \left[\omega^c_{c/n}\right]_\times u^c\right)\right) = 0 \end{aligned} \tag{5}$$

(5) might be rewritten as a linear equation in $v^c_{c/n}$. The crossproduct term is defined as:

$$c := u^c \times \left(\dot{u}^{cT} + \left[\omega^c_{c/n}\right]_\times u^c\right) = [c_x, c_y, c_z]^T$$

If the angular velocity is measured, then all quantities in the crossproduct term c are known. Using the definition of c, (5) is rewritten as

$${v^c_{c/n}}^T c = c^T v^c_{c/n} = 0 \tag{6}$$

Assuming that a fixed-wing UAV will never have zero forward velocity, then since $\{\mathrm{C}\}$ and $\{\mathrm{B}\}$ are aligned, one can divide (6) by the forward velocity component $v_x \neq 0$

$$\frac{1}{v_x}c^T\begin{bmatrix}v_x\\v_y\\v_z\end{bmatrix} = c^T\begin{bmatrix}1\\\frac{v_y}{v_x}\\\frac{v_z}{v_x}\end{bmatrix} = c_x + [c_y, c_z]\begin{bmatrix}\frac{v_y}{v_x}\\\frac{v_z}{v_x}\end{bmatrix} = 0$$

$$[c_y, c_z]\begin{bmatrix}\frac{v_y}{v_x}\\\frac{v_z}{v_x}\end{bmatrix} = -c_x \tag{7}$$

As can be seen from (7), one ends up with a linear equation. Assuming N features, the scaled body-fixed velocity with unity forward component can be found as:

$$v^c_{c/n} = v_x A^+ b, \quad v_x \neq 0$$

$$A = \begin{bmatrix}c_{y,1} & c_{z,1}\\ \vdots & \\ c_{y,N} & c_{z,N}\end{bmatrix}$$

$$b = -\begin{bmatrix}c_{x,1}\\ \vdots\\ c_{x,N}\end{bmatrix}$$

$$u^c_j \times \left(\dot{u}^{cT}_j + \left[\omega^c_{c/n}\right]_\times u^c_j\right) = [c_{x,j}, c_{y,j}, c_{z,j}]^T \tag{8}$$

This gives a correct solution only if A has full rank. This can only happen if the OF algorithm chooses linearly independent feature points and OF vectors as defined in Def. 1. Linearly independent OF vectors are in general obtained by not choosing all features from the same line in the image plane. $u^c_j = K^{-1}\underline{x}^m_j$ and $\dot{u}^c_j = K^{-1}[\dot{r}_j, \dot{s}_j, 0]^T$ are the back projected coordinate and OF of feature j respectively. Recall the sign ambiguity of the epipolar geometry, meaning that one must know the sign of v_x to recover the normalized linear velocity. For a fixed-wing UAV the forward velocity will always be greater than zero, $v_x > 0$.

Definition 1. ***Linearly Independent Optical Flow Vectors***
A pair of image features and their corresponding optical flow vectors x^m_1, $\dot{x}^m_1$ and x^m_2, $\dot{x}^m_2$, are said to be linearly independent if and only if the rank of A in (8) *is full, yielding $[v_y, v_z]^T = v_x A^+ b$ to be uniquely defined. The rank is full if and only if some 2×2 sub-matrix of A, $A_{2\times 2}$, has det($A_{2\times 2}$) $\neq 0$.*

IV. Observer Design

A. Kinematics

The kinematics of attitude, position, and velocity are described by

$$\dot{R}^n_b = R^n_b \left[\omega^b_{b/n}\right]_\times \tag{9a}$$

$$\dot{p}^n_{b/n} = v^n_{b/n} \tag{9b}$$

$$\dot{v}^n_{b/n} = f^n_{b/n} + g^n \tag{9c}$$

The objective is to estimate the attitude R^n_b, the position $p^n_{b/n}$, and the velocity $v^n_{b/n}$ with exponential convergence rate. In addition to this, an estimator for the gyro bias b^b_{gyro} is also provided.

B. Assumptions

The observer design by [3] is based on the following assumptions:

Assumption 1. *The gyro bias b^b_{gyro} is constant, and there exists a known constant $L_b > 0$ such that $\|b^b_{\text{gyro}}\| \leq L_b$.*

Assumption 2. *There exists a constant $c_{obs} > 0$ such that, $\forall t \geq 0$, $\|v^b_{cv} \times f^b_{\text{imu}}\| \geq c_{obs}$.*

Assumption 2 states that the UAV cannot have a specific force parallel to the velocity of the UAV. Furthermore neither the specific force nor the velocity can be identically equal to zero. In practice this condition restricts the types of maneuvers that ensure guaranteed performance of the proposed observer. This is however not a problem for fixed-wing UAVs as they always have forward speed to remain airborne. Moreover the observer does not converge while the vehicle is at rest without aiding from e.g. a magnetometer, but presents no issues during flight.

For the CEOF observer, two assumptions are introduced to ensure that CV can recover the body-fixed velocity.

Assumption 3. *The UAV has forward body-fixed velocity, $v_x > 0$.*

Assumption 4. *The OF algorithm provides at least two linearly independent OF vectors, as defined in Def.1.*

C. Observer Equations

Provided Assumptions 1-4 hold, the CEOF observer representation is stated as

$$\Sigma_1 \begin{cases} \dot{\hat{R}}{}_b^n = \hat{R}_b^n S(\omega_{\text{imu}}^b - \hat{b}_{\text{gyro}}^b) + \sigma K_P \hat{J} \\ \dot{\hat{b}}{}_{\text{gyro}}^b = \text{Proj}(\hat{b}_{\text{gyro}}^b, -k_I \text{vex}(\mathbb{P}_a(\hat{R}_s^T K_P \hat{J}))) \end{cases} \tag{10}$$

$$\Sigma_2 \begin{cases} \dot{\hat{p}}{}_{b/n}^n = \hat{v}_{b/n}^n + K_{pp}(p_{\text{GPS}}^n - \hat{p}_{b/n}^n) \\ \qquad + K_{pv}(v_{\text{GPS}}^n - \hat{v}_{b/n}^n) \\ \dot{\hat{v}}{}_{b/n}^n = \hat{f}_{b/n}^n + g^n + K_{vp}(p_{\text{GPS}}^n - \hat{p}_{b/n}^n) \\ \qquad + K_{vv}(v_{\text{GPS}}^n - \hat{v}_{b/n}^n) \\ \dot{\xi} = -\sigma K_P \hat{J} f_{\text{imu}}^b + K_{\xi p}(p_{\text{GPS}}^n - \hat{p}_{b/n}^n) \\ \qquad + K_{\xi v}(v_{\text{GPS}}^n - \hat{v}_{b/n}^n) \\ \hat{f}_{b/n}^n = \hat{R}_b^n f_{\text{imu}}^b + \xi \end{cases} \tag{11}$$

$$\text{CV} \begin{cases} v_{cv}^b = \text{sign}(v_x) \frac{v_e}{\|v_e\|} \\ v_e = \frac{v_{b/n}^b}{v_x} = [1, (A^+ b)^T]^T, \quad v_x \neq 0 \\ u_j^c \times \left(\dot{u}_j^{cT} + \left[\omega_{\text{imu}}^b - \hat{b}_{\text{gyro}}^b \right]_\times u_j^c \right) \\ \quad = [c_{x,j}, c_{y,j}, c_{y,j}]^T \end{cases} \tag{12}$$

The subsystem Σ_1 represents the attitude observer, whereas Σ_2 represents the translational motion observer. The CV gives (12), together with (8). $\sigma \geq 1$ is a scaling factor tuned to achieve stability, k_I is a positive scalar gain and K_P is a symmetric positive definite gain matrix. $\text{Proj}(\cdot,\cdot)$ represents a parameter projection [54] that ensures that $\|\hat{b}_{\text{gyro}}^b\|$ does not exceed a design constant $L_{\hat{b}} > L_b$ (see Appendix A), and $\hat{R}_s = \text{sat}(\hat{R}_b^n)$. $K_{pp}, K_{pv}, K_{vp}, K_{vv}, K_{\xi p}$, and $K_{\xi v}$ are observers gains, and g^n is the gravity vector in {N}. The matrix $\hat{J}$ is the output injection term, whose design is inspired by the TRIAD algorithm [55] and defined as

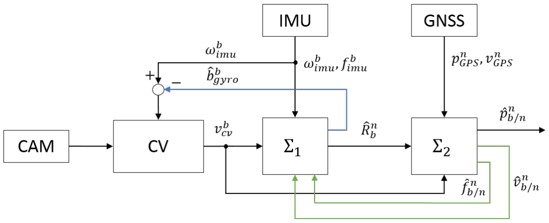


Fig. 4. Block diagram of the observer. Σ_1 represents the attitude observer, and Σ_2 the translational motion observer. The feedback illustrated in green have been proved to yield USGES stability of the nonlinear observer. The stability of the gyrob bias feedback illustrated in blue has not been analysed.

$$\hat{J}(v_{cv}^b, \ \hat{v}_{b/n}^n, f_{\text{imu}}^b, \hat{f}_{b/n}^n, \hat{R}_b^n) := \hat{A}_n A_b^T - \hat{R}_b^n A_b A_b^T \tag{13a}$$

$$A_b := [f_{\text{imu}}^b, \ f_{\text{imu}}^b \times v_{cv}^b, \ f_{\text{imu}}^b \times (f_{\text{imu}}^b \times v_{cv}^b)] \tag{13b}$$

$$\hat{A}_n := [\hat{f}_{b/n}^n, \ \hat{f}_{b/n}^n \times \hat{v}_{b/n}^n, \ \hat{f}_{b/n}^n \times (\hat{f}_{b/n}^n \times \hat{v}_{b/n}^n)] \tag{13c}$$

The system Σ_1–Σ_2 is a feedback interconnection, as illustrated by Fig. 4.

D. Stability Proof

The error dynamics of the nonlinear observer can be written in a compact form as

$$\Sigma_1 \begin{cases} \dot{\tilde{R}}{}_b^n = R_b^n \left[\omega_{b/n}^b \right]_\times - \hat{R}_b^n \left[\omega_{\text{imu}}^b - \hat{b}_{\text{gyro}}^b \right]_\times - \sigma K_P \hat{J} \\ \dot{\tilde{b}}{}_{\text{gyro}}^b = -\text{Proj}(\hat{b}_{\text{gyro}}^b, -k_I \text{vex}(\mathbb{P}_a(\hat{R}_s^T K_P \hat{J}))) \end{cases} \tag{14a}$$

$$\Sigma_2 \left\{ \dot{\tilde{w}} = (A_w - K_w C_w)\tilde{w} + B_w \tilde{d} \right. \tag{14b}$$

where $\tilde{w} = [(\tilde{p}_{b/n}^n)^T, (\tilde{v}_{b/n}^n)^T, (\tilde{f}_{b/n}^n)^T]^T$ collects the estimated position, velocity and acceleration vectors, $\tilde{d} = \left(R_b^n \left[\omega_{b/n}^b \right]_\times - \hat{R}_b^n \left[\omega_{\text{imu}}^b - \hat{b}_{\text{gyro}}^b \right]_\times \right) f_{b/n}^b + \left(R_b^n - \hat{R}_b^n \right) \dot{f}_{b/n}^b$, and the four matrices in (14b) are defined as

$$A_w = \begin{bmatrix} 0_{6\times3} & I_6 \\ 0_{3\times3} & 0_{3\times6} \end{bmatrix}, \qquad B_w = \begin{bmatrix} 0_{6\times3} \\ I_3 \end{bmatrix},$$

$$C_w = \begin{bmatrix} I_6 & 0_{6\times3} \end{bmatrix}, \qquad K_w = \begin{bmatrix} K_{pp} & K_{pv} \\ K_{vp} & K_{vv} \\ K_{\xi p} & K_{\xi v} \end{bmatrix}.$$

The following theorem can be stated about the stability of the nonlinear observer (10)-(12), if assuming that $\hat{b}^b_{\text{gyro}}$ is kept constant in (12).

Theorem 1. ***(Stability of the CEOF observer)*** *Let σ be chosen to ensure stability according to Lemma 1 in [5] and define $H_K(s) = (Is - A_w + K_w C_w)^{-1} B_w$. There exists a set (0,c) such that, if K_w is chosen such that $A_w - K_w C_w$ is Hurwitz, and $\|H_K(s)\|_\infty < \gamma$, for $\gamma \in (0, c)$, then the origin of the error dynamics* (10)-(12)*, provided Assumptions 1-4, is USGES when the initial conditions satisfy $\|\hat{b}^b_{\text{gyro}}(0)\| \leq L_{\hat{b}}$.*

Proof: Proof is based on Theorem 1 in [3], where we have replaced M with the new computer vision subsystem from (12). We must show that v^b_{cv} is uniquely defined. Then it follows from Theorem 1 in [3] that the origin of the error dynamics (10)-(12) is USGES.

v^b_{cv} has a one to one mapping to the scaled body-fixed velocity with unit forward component v_e. Moreover if the sign of v_x is known, then $v^b_{cv} = \frac{v^b_{b/n}}{\|v^b_{b/n}\|}$. From Assumption 3 $v_x > 0$, hence the uniqueness of v^b_{cv} can be shown by the uniqueness of v_e. $v_e = [1, (A^+ b)^T]^T$ has a unique solution if and only if the rank of A is full [53]. Given that the computer vision algorithm extracts features such that Assumption 4 is not violated, then A has full rank, and v_e is uniquely determined. Hence v^b_{cv} is uniquely determined, and it follows from Theorem 1 in [3] that the system is USGES. ■

V. Experimental Results

An experiment is carried out to validate the theory in practice. The UAV employed is a UAV Factory Penguin-B, equipped with a custom-made payload that includes all the necessary sensors. The IMU is a Sensonor STIM300, a low-weight, tactical grade, high-performance sensor that includes gyroscopes, accelerometers, and inclinometers, all recorded at a frequency of 300 Hz. The chosen GPS receiver is a uBlox LEA-6T, which gives measurements at 5 Hz. The video camera is an IDS GigE uEye 5250CP provided with a 8mm lens. The camera is configured for a hardware-triggered capture at 10 Hz. The experiment has been carried out on 6 February 2015 at the Eggemoen Aviation and Technology Park, Norway, in a sunny day with good visibility, very little wind, an air temperature of about -8°C. The terrain is relatively flat and covered with snow.

The observer is evaluated offline with the flight data gathered at the experiment. It is implemented using first order forward Euler discretisation with a time-varying step depending on the interval of the data acquisition of the fastest sensor, namely the STIM300, and it is typically around 0.003 seconds. The gyro bias is initialized by averaging the gyroscope measurement at stand still before take-off. The position estimate is initialized by using the first GPS measurement, while the NED velocity is initialized by the difference between the two first consecutive GPS measurements. The various parameters and gains are chosen as $L_b = 2°/\text{s}$, $L_{\hat{b}} = 2.1°/\text{s}$, $\sigma = 1$, $K_P = \text{diag}[0.08, 0.04, 0.06]$, $k_I = 0.0001$, $K_{pp} = 30 I_{3\times3}$, $K_{pv} = 2 I_{3\times3}$, $K_{vp} = 0.01 I_{3\times3}$, $K_{vv} = 20 I_{3\times3}$, $K_{\xi p} = I_{3\times3}$, and $K_{\xi v} = 50 I_{3\times3}$.

The reference provided for the attitude, position, and velocity is the output of the EKF of the autopilot mounted on the Penguin-B. A reference for the gyro bias is not available.

All the images are processed with a resolution of 1600×1200 (width×height) pixels and in their original state, without any filtering. The lens distortion of the camera is not accounted for, and no correction is applied to the images. SIFT is implemented with the open source computer vision library (OpenCV) [56] with default settings. Each match is tagged with a value indicating the accuracy of the match, and the smallest of these values is considered to be the best match. To increase the reliability of the OF vectors, each match is compared to the best one. Every match with an uncertainty more than double the uncertainty of the best match is removed. Also the template matching algorithm is implemented with OpenCV. The size of the templates is chosen to be 120×90 pixels and a correlation of 99% is required in order for a template match to be considered reliable and not removed.

In addition to the CEOF and GTOF observer, a nonlinear observer without CV is implemented.

This is done by removing the CV subsystem in (12) from the nonlinear observer, and approximating the body-fixed linear velocity measurement by $v^b = [1, 0, 0]^T$. The nonlinear observer without CV is denoted NoCV. Although Theorem 1 does not cover feedback of the gyro bias estimate to CV in the CEOF nonlinear observer, this feedback is implemented. This is assumed to increase the accuracy without being destabilizing, as the bias estimator is tuned to have slow dynamics.

A. *Results*

The results presented here refer to a complete flight of the Penguin-B, from take-off to landing. The time on the x-axis is the elapsed time since the data logging began, and only the significant part involving the flight is presented. The maneuvers performed include flights on a straight line and turns with a large and small radius of curvature, approximately 200 m and 100 m.

Fig. 5 shows the measured body-fixed velocity from the GTOF CV. The measurements are contaminated by noise. The mean values are close to the reference, although the mean forward velocity (u) is slightly greater than the reference. The measured crab and flight path angle of the UAV are shown in Fig. 6. It is seen that both the GTOF and CEOF CV succeeds in measuring the correct direction, but GTOF has a larger noise level than CEOF.

Fig. 7 illustrates the estimated attitude. It can be seen that all observers need approximately 60 seconds to converge. The estimates of the roll angle are fairly similar for NoCV, GTOF and CEOF. The estimated pitch angle has a small offset for all nonlinear observers throughout the entire flight. The yaw angle estimate is almost identical for the NoCV, GTOF and CEOF. Fig. 8 and Fig. 9 illustrates the estimated velocity and position in {N}, and shows small differences for NoCV, GTOF and CEOF. The estimated gyro bias is seen in Fig. 10. No bias reference is available, but the estimated bias is close to equal for NoCV, GTOF and CEOF. The flight terrain is relatively flat and the UAV has small crab and flight path angle during the flight. Therefore the weaknesses of the GTOF and NoCV observer are not significant in the results. However the experimental results show that the nonlinear observers yield small deviations from the reference EKF, and that the CV give reasonable estimates of normalized body-fixed velocity.

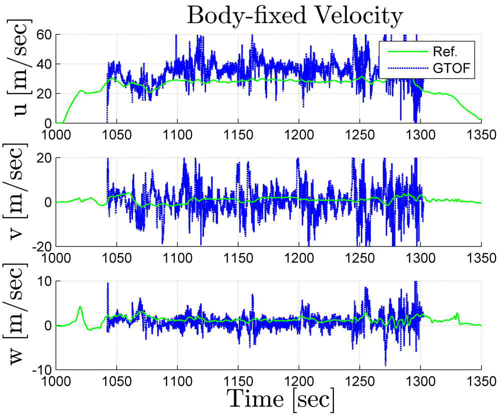


Fig. 5. Measured and estimated body-fixed velocity by GTOF and autopilot EKF respectively.

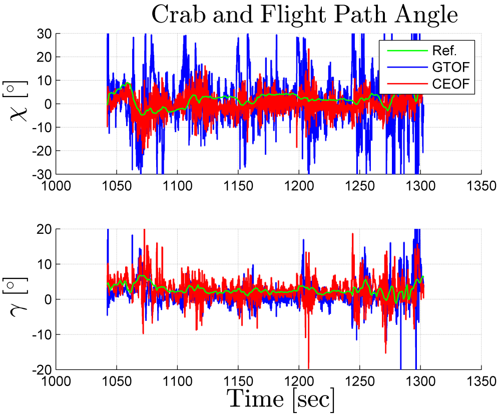


Fig. 6. Measured and estimated crab and flight path angle.

VI. Simulation Results

In order to evaluate the NoCV, GTOF and CEOF observer representations in the presence of more rugged terrain and to compare with an exactly known reference, a simulator is implemented in Matlab. An elevation profile of a coastline is generated, and a UAV flight is simulated.

The following parameters and gains are chosen identical for the NoCV, GTOF and CEOF observer: $L_b = 2°/\text{s}$, $L_{\hat{b}} = 2.1°/\text{s}$, $\sigma = 1$, $K_{pp} = \text{diag}[5, 5, 0.7]$, $K_{pv} = \text{diag}[50, 50, 50]$, $K_{vp} =$

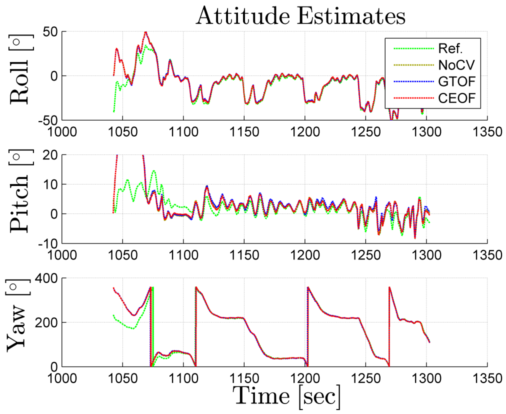


Fig. 7. Estimated attitude by the observers.

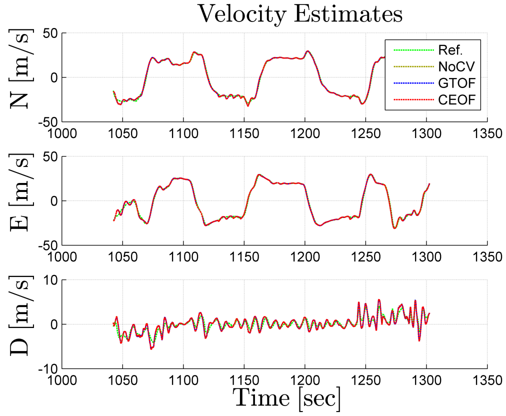


Fig. 8. Estimated velocity by the observers.

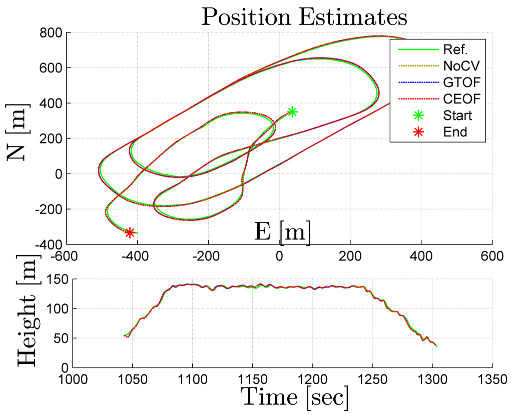


Fig. 9. Estimated position by the observers.

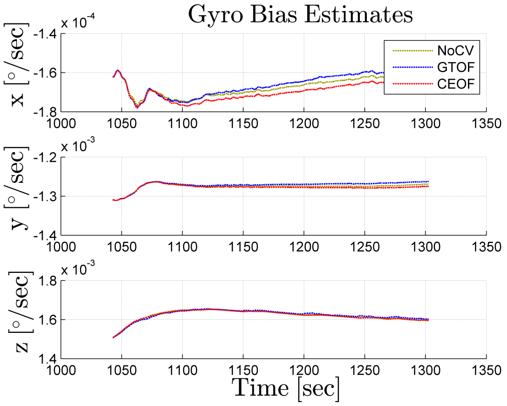


Fig. 10. Estimated gyro bias by the observers.

diag$[0.1, 0.1, 0.01]$, $K_{vv} = 10 I_{3\times3}$, $K_{\xi p} = 0.1 I_{3\times3}$, and $K_{\xi v} = 5 I_{3\times3}$. For the GTOF and CEOF observer $K_P = I_{3\times3}$ and $k_I = 0.03$ are chosen. The NoCV is tuned with $K_P = \text{diag}[1, 0.2, 0.1]$ and $k_I = 0.01$.

All observers are initialised with $\hat{R}^n_b = I_{3\times3}$, $\hat{b}^b_{\text{gyro}} = 0_{3\times1}$. $\hat{p}^n_{b/n}$ and $\hat{v}^n_{b/n}$ are initialised as the first GNSS position and velocity measurement respectively.

A. *UAV Path*

Linear and angular velocity, $v^b_{b/n}$ and $\omega^b_{b/n}$, are generated over a time interval of $200\,\text{sec}$. Wind directed straight north with magnitude 5m/s is simulated causing the UAV to have a crab angle. As the camera measures the velocity relative to the ground, one does not have to consider the sideslip angle. Kinematic equations are used to generate positions and attitude of the UAV.

$$v^n_{b/n} = R^n_b(\Theta_{b/n}) v^b_{b/n} \tag{15}$$

$$f^n_{b/n} = R^n_b(\Theta_{b/n})(\dot{v}^b_{b/n} + \omega^b_{b/n} \times v^b_{b/n}) - g^n \tag{16}$$

$$f^b_{b/n} = (R^n_b)^T(\Theta_{b/n}) f^n_{b/n} \tag{17}$$

$$\dot{\Theta}_{b/n} = T_\Theta(\Theta_{b/n}) \omega^b_{b/n} \tag{18}$$

$R^n_b(\Theta_{b/n})$ and $T_\Theta(\Theta_{b/n})$ being the rotation matrix between $\{\text{B}\}$ and $\{\text{N}\}$ and the angular transformation matrix respectively. The variables are integrated numerically with first order Euler integration

$$\Theta_{b/n}(k+1) = \Theta_{b/n}(k) + \delta t \dot{\Theta}_{b/n} \tag{19}$$

$$p^n_{b/n}(k+1) = p^n_{b/n}(k) + \delta t v^n_{b/n} \tag{20}$$

B. *Sensor Data*

Sensor data are generated before running the observer. A gyroscope, accelerometer, inclinometer, GNSS and CV are simulated. The GNSS is simulated to measure $\{\text{N}\}$ position and velocity, and CV is simulated to measure the OF. The gyroscope, accelerometer, inclinometer are configured to output measurements with a rate of 100 Hz. The GNSS is configured to output measurements at 5Hz. The noise on the position measurement from GNSS is modelled as a Gauss-Markov process by $\nu(k+1) = e^{-K_{\text{GNSS}}\Delta T}\nu(k) + \eta_{\text{GNSS}}$, with noise parameters given in Table I.

TABLE I. GAUSS-MARKOV ERROR MODEL PARAMETERS FOR GNSS POSITION MEASUREMENTS.

Direction	*Std. dev.* η_{GNSS}[m]	$1/K_{\text{GNSS}}$[s]	ΔT_{GNSS}[s]
North	0.21	360	0.2
East	0.21	360	0.2
Down	0.4	360	0.2

The camera is simulated to capture 25 frames per second. The camera extracts features and calculates OF as described in Appendix B. White noise is added to the IMU, inclinometer, camera and velocity from GNSS sensor data by the multivariate normal random noise-function, `mvnrnd`, in Matlab. Inclinometer measurements are denoted $\Theta^b_{incl} = [\phi, \theta]^T$. The following mean and covariance are used:

$$
\begin{aligned}
w_{\omega^b_{\text{imu}}} &\sim \mathcal{N}(0_{3\times1}, \Sigma_{\omega^b_{\text{imu}}}), \Sigma_{\omega^b_{\text{imu}}} = (0.135\,\text{deg})^2 I_{3\times3} \\
w_{f^b_{\text{imu}}} &\sim \mathcal{N}(0_{3\times1}, \Sigma_{f^b_{\text{imu}}}), \Sigma_{f^b_{\text{imu}}} = (1.29\cdot10^{-3}\text{g})^2 I_{3\times3} \\
w_{\Theta^b_{incl}} &\sim \mathcal{N}(0_{2\times1}, \Sigma_{\Theta^b_{incl}}), \Sigma_{\Theta^b_{incl}} = (0.18\,\text{deg})^2 I_{2\times2} \\
w_{v^n_{GNSS}} &\sim \mathcal{N}(0_{3\times1}, \Sigma_{v^n_{GNSS}}), \Sigma_{v^n_{GNSS}} = (0.21\text{m/s})^2 I_{3\times3}
\end{aligned}
$$

No bias on the accelerometer is assumed, and a constant bias is assumed on the gyroscope. The gyroscope is simulated with the following bias

$$
b^b_{\text{gyro}} = \begin{bmatrix} 0.1\text{deg/s} \\ -0.3\text{deg/s} \\ -0.35\text{deg/s} \end{bmatrix}
$$

White noise is also added to the OF data from the simulated camera. Every extracted feature is given white noise with variance, $\sigma^2_{dr} = \sigma^2_{ds} = \sigma^2_d = (4.5\cdot10^{-5}\text{mm})^2$. As two corresponding features are needed to get an OF vector, the resulting noise of the OF vector has variance $\sigma_{OF} = \sigma^2_d I_{2\times2}$. On a camera chip with 1600×1200 pixels and dimension 7.2×5.4 mm, this would yield a small variance of $(\sqrt{2}\cdot0.01\text{px})^2$ for the OF vector noise.

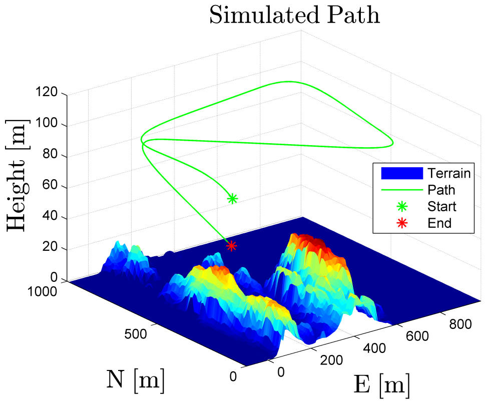


Fig. 11. The simulated UAV path and the elevation profile of the terrain model.

C. *Terrain Simulation*

In order to evaluate the performance of the GTOF and CEOF observer representations with a realistic environment, a terrain model is generated. The terrain model is a matrix, Z, with values corresponding to the elevation profile of the terrain. It is also called the elevation profile of the terrain, as it describes the elevation of the terrain. The terrain model is made to mimic a coastline, and has a resolution of $1\text{m}\times1\text{m}$ meter. The covered area is $1\text{km}\times1\text{km}$. At position x, y of the matrix the elevation h of the terrain at x meters North and y meters East is found. A point on the surface of the terrain will have NED coordinate $x, y, -h$. Fig. 11 displays the simulated UAV path and the terrain model.

D. *Results*

Fig. 12 shows the crab angle error and the flight path angle error in the measured normalized body-fixed velocity from CV. It can be seen that the GTOF fails to produce correct measurement of the body-fixed velocity when the terrain is non-planar (at time 110-220 seconds). Any crab and flight path angle of the UAV causes NoCV to fail as it assumes pure forward motion.

Fig. 13 and Fig. 14 show the attitude estimates and the error in the estimates. The NoCV observer fails to produce accurate estimates of the attitude. It is seen that the accuracy of the GTOF observer is heavily reduced when flying over the non-planar area. The CEOF observer on the other hand is not limited by the rugged terrain, and provides accurate estimates during the entire flight. The estimated and real gyro bias is displayed in Fig. 15. It is seen that the bias values from NoCV does not converge to the correct value. The shortcomings of the GTOF observer is again illustrated when the UAV flies over the non-planar area.

Fig. 16 and 17 show the real and estimated velocity and position. The estimates are close to the reference and quite similar for GTOF and CEOF. This is expected as the velocity and position measurements from GNSS have the largest influence on these estimates.

Table II provides numerical evaluation of the observers in means by the Root Mean Squared (RMS) error. The CEOF observer has lower RMS in the estimates of the attitude than the GTOF observer. NoCV has the least accurate estimates in attitude, and is outclassed by CEOF. There are no major differences in estimated position and velocity. However CV seem to slightly increase the accuracy in estimated position. The estimated gyro bias is most accurate with the CEOF and least accurate with NoCV. The crab angle and flight path angle error are reduced significantly with CV. This is because NoCV assumes zero crab and flight path angle, which is not the case.

Overall the CEOF observer proves to be much more reliable than GTOF and NoCV, with a robust and accurate performance. The GTOF performs better than NoCV, which supports the use of CV in the observer. However the validity of the GTOF observer is restricted to horizontal planar terrain, which limits the range of use in practice. CEOF is not restricted by the same limitations and thus more applicable in practice.

VII. Conclusions

In this paper two different vision-aided nonlinear observers, and one nonlinear observer without CV, for estimation of position, velocity and attitude have been evaluated on real experimental data obtained by flying a fixed-wing UAV with a custom-made payload of sensors. The nonlinear observers have also been tested on simulated data to compare the performance of the observers with the presence of non-planar terrain and with an exact known reference for comparison. The results show that using CV increases the accuracy of the nonlinear observer, especially in estimated attitude. This is because CV provides useful information about the direction of the body-fixed velocity. Furthermore the CEOF nonlinear observer has shown to be a more robust option than the GTOF nonlinear observer, as it is terrain independent.

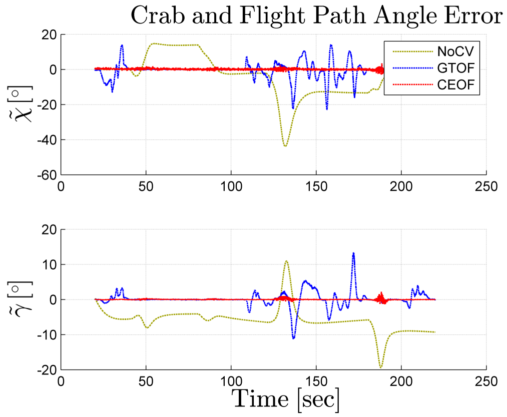


Fig. 12. Error in crab ($\tilde{\chi}$) and flight path angle ($\tilde{\gamma}$) for the measured normalized body-fixed velocity.

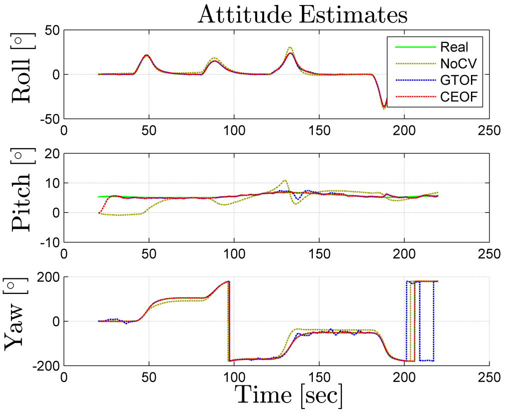


Fig. 13. Estimated attitude. When the UAV flies over the rugged terrain, the GTOF observer fails to produce correct estimates of the attitude.

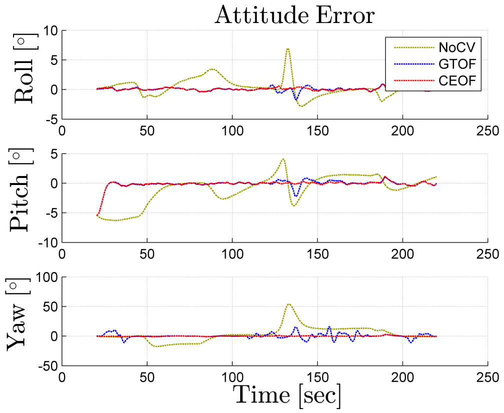


Fig. 14. Error in estimated attitude. When the UAV flies over the rugged terrain, the GTOF observer fails to produce correct estimates of the attitude. The NoCV observer fails to estimate correct pitch angle.

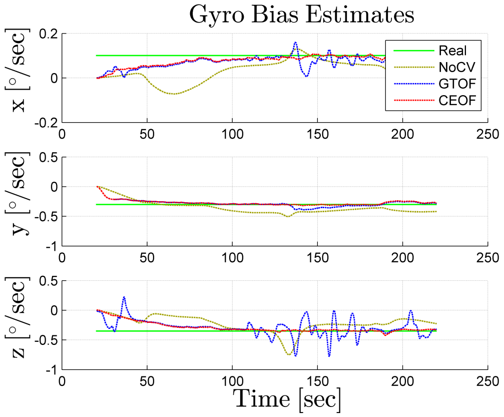


Fig. 15. Estimated gyro bias together with the real gyro bias. After 100 sec the gyro bias has converged. When flying over the rugged terrain the GTOF observer produces erroneous gyro bias estimates, while the CEOF observer is unaffected.

Acknowledgements

The authors are grateful for the assistance provided by the UAV engineers at NTNU and Maritime Robotics AS, in particular Lars Semb and Carl Erik Stephansen. Significant contributions to the construction of the UAV payload was made by the rest of the navigation team at NTNU, in particular Sigurd M. Albrektsen, Jakob M. Hansen and Kasper T. Borup.

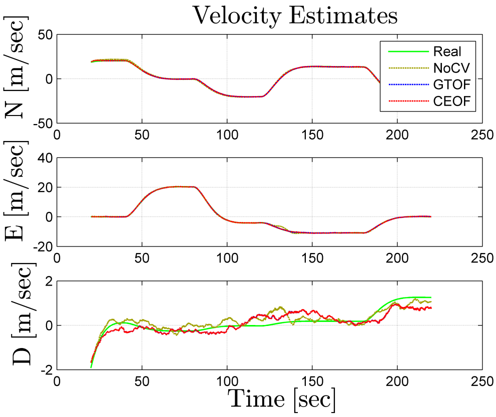


Fig. 16. Estimated velocity.

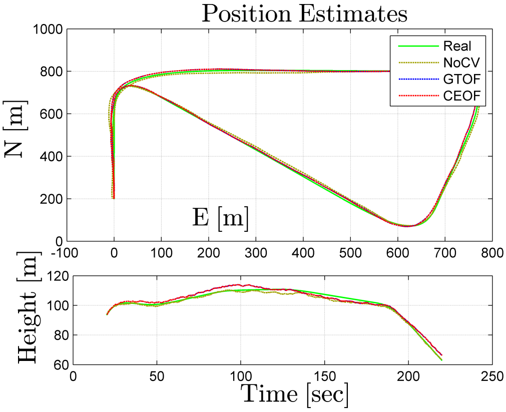


Fig. 17. Estimated position.

References

[1] R. Mahony, T. Hamel, and J. M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. Xx, pp. 1203–1218, 2008.

[2] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of Nonlinear Attitude Estimation Methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[3] L. Fusini, T. I. Fossen, and T. A. Johansen, "A Uniformly Semiglobally Exponentially Stable Nonlinear Observer for GNSS-and Camera-Aided Inertial Navigation," in *Proceedings of the 22nd IEEE Mediterranean Conference on Control and Automation (MED'14)*, 2014.

[4] M. D. Hua, "Attitude estimation for accelerated vehicles using {GPS/INS} measurements," *Control Engineering Practice*, vol. 18, no. 7, pp. 723–732, 2010.

[5] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi, "A nonlinear observer for integration of {GNSS and IMU} measurements with gyro bias estimation," *American Control Conference (ACC)*, pp. 4607–4612, 2012.

TABLE II. RMS values for the estimated states in the different cases using the ground truth optical flow (GTOF) and the continuous epipolar optical flow (CEOF) observer representation. $\tilde{\chi}$ and $\tilde{\gamma}$ are the crab angle- and flight path angle error in the body-fixed velocity measurement from the computer vision (CV), given in degrees. The gyro bias converges after approximately 100 second, hence the RMS values of the attitude and bias is considered from 100 seconds after start.

		Nonlinear Observer		
		NoCV	**GTOF**	**CEOF**
$\Theta_{b/n}$ (° RMS)	**Roll**	1.1255	0.48963	0.16426
	Pitch	1.2942	0.3906	0.15134
	Yaw	16.1414	4.9359	0.31014
$p^n_{b/n}$ (m RMS)	**North**	9.5084	6.9281	6.9506
	East	4.4114	4.2439	4.2623
	Down	1.2397	0.79603	0.80051
$v^n_{b/n}$(m/s RMS)	**North**	0.63969	0.1526	0.15025
	East	0.32982	0.10317	0.084081
	Down	0.1619	0.15283	0.15257
b^b_{gyro} (°/s RMS)	**Roll**	0.029879	0.018871	0.0054807
	Pitch	0.063454	0.021199	0.0062694
	Yaw	0.095493	0.10302	0.0088266
CV (° RMS)	$\tilde{\chi}$	12.1701	4.7667	0.46633
	$\tilde{\gamma}$	6.8836	2.2066	0.16025

[6] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2011.

[7] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing UAV," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 340–345, 2008.

[8] M. Mammarella, G. Campa, M. L. Fravolini, and M. R. Napolitano, "Comparing Optical Flow Algorithms Using 6-DOF Motion of Real-World Rigid Objects," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1752–1762, Nov. 2012.

[9] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "MAV navigation through indoor corridors using optical flow," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 3361–3368.

[10] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 20–25, 2011.

[11] D. Dusha, W. Boles, and R. Walker, "Attitude Estimation for a Fixed-Wing Aircraft Using Horizon Detection and Optical Flow," *Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, pp. 485–492, 2007.

[12] S. Weiss, R. Brockers, and L. Matthies, "4DoF drift free navigation using inertial cues and optical flow," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4180–4186, 2013.

[13] J. C. Zufferey and D. Floreano, "Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, pp. 2594–2599, 2005.

[14] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 302–309, 2005.

[15] P. C. Merrell, D.-J. Lee, and R. W. Beard, "Obstacle avoidance for unmanned air vehicles using optical flow probability distributions," *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 5609, no. 1, pp. 13–22, 2004.

[16] J. Conroy, G. Gremillion, B. Ranganathan, and J. S. Humbert, "Implementation of wide-field integration of optic flow for autonomous quadrotor navigation," *Autonomous Robots*, vol. 27, no. 3, pp. 189–198, 2009.

[17] F. Ruffier and N. Franceschini, "Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction," *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, no. April, pp. 2339–2346, 2004.

[18] P. C. Merrell, D.-J. Lee, and R. W. Beard, "Statistical analysis of multiple optical flow values for estimation of unmanned aerial vehicle height above ground," *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 5608, pp. 298–305, 2004.

[19] R. Brockers, S. Susca, D. Zhu, and L. Matthies, "Fully self-contained vision-aided navigation and landing of a micro air vehicle independent from external sensor inputs," *SPIE Defense, Security, and Sensing*, pp. 83 870Q–83 870Q–10, 2012.

[20] B. Herisse, F. X. Russotto, T. Hamel, and R. Mahony, "Hovering flight and vertical landing control of a VTOL Unmanned Aerial Vehicle using optical flow," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 801–806, 2008.

[21] J. J. Kehoe, A. S. Watkins, R. S. Causey, and R. Lind, "State estimation using optical flow from parallax-weighted feature tracking," *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, vol. 8, pp. 5030–5045, 2006.

[22] R. J. D. Moore, S. Thurrowgood, and M. V. Srinivasan, "Vision-only estimation of wind field strength and direction from an aerial platform," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4544–4549, 2012.

[23] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 957–964, 2012.

[24] D. A. Mercado, G. Flores, P. Castillo, J. Escareno, and R. Lozano, "GPS/INS/optic flow data fusion for position and Velocity estimation," *International Conference on Unmanned Aircraft Systems, ICUAS - Conference Proceedings*, pp. 486–491, 2013.

[25] M. Bibuli, M. Caccia, and L. Lapierre, "Path-following algorithms and experiments for an autonomous surface vehicle," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 7, no. 6, pp. 81–86, 2007.

[26] S. Ahrens, D. Levine, G. Andrews, and J. P. How, "Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2643–2648, 2009.

[27] P. Batista, C. Silvestre, and P. Oliveira, "{GES} Attitude Observers - {Part I}: Multiple General Vector Observations," *Proceedings of the 18th IFAC World Congress*, vol. 18, pp. 2985–2990, 2011.

[28] ——, “{GES} Attitude Observers - {Part II}: Single Vector Observations,” *Proceedings of the 18th IFAC World Congress*, vol. 18, pp. 2991–2996, 2011.

[29] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi, “Attitude Estimation Using Biased Gyro and Vector Measurements With Time-Varying Reference Vectors,” *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1332–1338, 2012.

[30] J. Guerrero-Castellanos, H. Madrigal-Sastre, S. Durand, L. Torres, and G. Muñoz Hernández, “A Robust Nonlinear Observer for Real-Time Attitude Estimation Using Low-Cost {MEMS} Inertial Sensors,” *Sensors*, vol. 13, no. 11, pp. 15 138–15 158, 2013.

[31] M. D. Hua, G. Ducard, T. Hamel, R. Mahony, and K. Rudin, “Implementation of a Nonlinear Attitude Estimator for Aerial Robotic Vehicles,” *IEEE Transactions on Control System Technology*, vol. 22, no. 1, pp. 201–213, 2014.

[32] R. Mahony, T. Hamel, J. Trumpf, and C. Lageman, “Nonlinear attitude observers on {SO(3)} for complementary and compatible measurements: A theoretical study,” *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 6407–6412, 2009.

[33] R. Mahony, M. Euston, J. Kim, P. Coote1, and T. Hamel, “A non-linear observer for attitude estimation of a fixed-wing unmanned aerial vehicle without {GPS} measurements,” *Transactions of the Institute of Measurement and Control*, vol. 33, no. 6, pp. 699–717, 2011.

[34] S. Salcudean, “A globally convergent angular velocity observer for rigid body motion,” *IEEE Transactions on Automatic Control*, vol. 36, no. 12, pp. 1493–1497, 1991.

[35] J. Thienel and R. M. Sanner, “A coupled nonlinear spacecraft attitude controller and observer with an unknown constant gyro bias and gyro noise,” *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 2011–2015, 2003.

[36] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi, “Globally exponentially stable attitude and gyro bias estimation with application to {GNSS/INS} integration,” *Automatica*, vol. 51, pp. 158–166, 2015.

[37] B.Lucas and T.Kanade, “An iterative image restoration technique with an application to stereo vision,” *Proc. DARPA Image Underst, Workshop*, pp. 121–130, 1981.

[38] B. K. Horn and B. G. Schunck, “Determining optical flow,” pp. 185–203, 1981.

[39] D. Lowe, “Object recognition from local scale-invariant features,” *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999.

[40] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded-Up Robust Features,” in *9th European Conference on Computer Vision*, vol. 110, 2008, pp. 346–359.

[41] D. D. Diel, P. DeBitetto, and S. Teller, “Epipolar Constraints for Vision-Aided Inertial Navigation,” *IEEE Workshop on Application of Computer Vision*, pp. 221–228, 2005.

[42] J. C. Bazin, C. Demonceaux, P. Vasseur, and I. S. Kweon, “Motion estimation by decoupling rotation and translation in catadioptric vision,” *Computer Vision and Image Understanding*, vol. 114, no. 2, pp. 254–273, 2010.

[43] M. Meingast, C. Geyer, and S. Sastry, “Vision based terrain recovery for landing unmanned aerial vehicles,” *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 2, 2004.

[44] M. Sanfourche, J. Delaune, and G. Besnerais, “Perception for UAV: Vision-based navigation and environment modeling,” *Aerospace Lab Journal*, pp. 1–19, 2012.

[45] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[46] Y. Ma, S. Soatto, J. Košecká, and S. Sastry, “An Invitation to 3D Vision,” *Springer*, 2001.

[47] V. Grabe, H. H. Bulthoff, and P. Robuffo Giordano, “Robust optical-flow based self-motion estimation for a quadrotor UAV,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 2153–2159, 2012.

[48] V. Grabe, H. H. Bulthoff, and P. R. Giordano, “On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow,” *2012 IEEE International Conference on Robotics and Automation*, pp. 491–497, 2012.

[49] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. The Press Syndicate of the University of Cambridge, 2003.

[50] J. N. Sarvaiya, S. Patnaik, and S. Bombaywala, “Image registration by template matching using normalized cross-correlation,” *International Conference on Advances in Computing, Control and Telecommunication Technologies*, pp. 819–822, 2009.

[51] C. S. Fuh and P. Maragos, “Region-based optical flow estimation,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Proceedings CVPR.*, pp. 130–135, 1989.

[52] S. Hutchinson, G. D. Hager, and P. I. Corke, “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[53] E. Kreyszig, *Advanced Engineering Mathematics*, 2006, vol. 53, no. 386.

[54] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: Wiley, 1995.

[55] M. D. Shuster and S. D. Oh, “Three-axis attitude determination from vector observations,” *Journal of Guidance, Control and Dynamics*, vol. 4, no. 1, pp. 70–77, 1981.

[56] G. Bradski, “The OpenCV Library,” *Dr Dobbs Journal of Software Tools*, vol. 25, pp. 120–125, 2000.

Appendix A
Parameter Projection

The parameter projection $\mathrm{Proj}(\cdot,\cdot)$ is defined as:

$$\mathrm{Proj}(\hat{b}^b,\tau)=\begin{cases}\left(I-\frac{c(\hat{b}^b)}{\|\hat{b}^b\|^2}\hat{b}^b\hat{b}^{bT}\right)\tau, & \|\hat{b}^b\|\geq L_b,\ \hat{b}^{bT}\tau>0\\ \tau, & \text{otherwise}\end{cases}$$

where $c(\hat{b}^b) = \min\{1, (\|\hat{b}^b\|^2 - L_b^2)/(L_{\hat{b}^b}^2 - L_{b^b}^2)\}$. This operator is a special case of that from Appendix E of [54].

Appendix B
Camera Simulator

A camera is simulated in order to get measurements of the OF. Here it is our objective to describe how a camera and OF algorithm can be simulated without having access to real images. The objective is to find the displacement of a projected point in the image plane between time t_{k-1} to t_k, that is $dr = r(t_k) - r(t_{k-1})$ and $ds = s(t_k) - s(t_{k-1})$. Lets first consider how one can choose features to project given the UAVs attitude, position and a elevation profile of the terrain. These features are the one that we wish to find the OF of.

At a time t_k a ray is drawn in the camera z-axis as shown in Figure 18. The ray intersects the ground plane at a point $t^n_{\text{centre}} = [x^n_{\text{centre}}, y^n_{\text{centre}}, 0]^T$ or expressed in $\{\text{C}\}$ $\underline{t}^c_{\text{centre}} = (T^n_c)^{-1}\underline{t}^n_{\text{centre}}$, T^n_c being the homogeneous transformation matrix relating $\{\text{C}\}$ and $\{\text{N}\}$. The point t^c_{centre} is named the "centre ground point".

Points are chosen deterministically around the centre ground point, t^c_{centre}, distributed on a plane perpendicular to the ray from $\{\text{C}\}$ to the centre ground point. This plane is named the field of view (FOV) plane. The points are distributed on the FOV plane, ranging from the centre ground point -30 to 30 meters in camera x-direction -40 to 40 meters in camera y-direction, separated with 10 meters in both dimensions. Lets call these points "FOV features" and denote them by p^c_{FOV}. The FOV features in camera coordinates is then defined as $p^c_{\text{FOV}} \in t^c_{\text{centre}} + [x, y, 0]^T, x \in [-30, -20, \dots, 20, 30], y \in [-40, -30, \dots, 30, 40]$. Lets now consider only one of the FOV features, and denote this FOV feature p^c_{FOV}.

The FOV feature p^c_{FOV} is transformed to $\{\text{N}\}$ by $\underline{p}^n_{\text{FOV}} = T^n_c \underline{p}^c_{\text{FOV}}$. Let the FOV feature be defined as $p^n_{\text{FOV}} = [x^n_{\text{FOV}}, y^n_{\text{FOV}}, z^n_{\text{FOV}}]$. The FOV feature is then projected onto the terrain by using $x^n_{\text{FOV}}, y^n_{\text{FOV}}$ and the elevation h at the $x^n_{\text{FOV}}, y^n_{\text{FOV}}$ coordinate of the elevation profile. The projected point is then $p^n = [x^n_{\text{FOV}}, y^n_{\text{FOV}}, h]^T$, which is called a "feature".

Now that the feature location in $\{\text{N}\}$ is found, it is in our interest to find the projection of this feature at time t_k and t_{k-1}. The camera moves between t_{k-1} and t_k, meaning the homogeneous transformation matrix T^n_c is time variant. The feature can then be transformed to $\{\text{C}\}$ by $\underline{p}^c(t_k) = (T^n_c)^{-1}(t_k)\underline{p}^n$ and $\underline{p}^c(t_{k-1}) = (T^n_c)^{-1}(t_{k-1})\underline{p}^n$. The points $p^c(t_k), p^c(t_{k-1})$ represents the feature on the surface of the terrain given in camera coordinates at time t_k and t_{k-1} respectively.

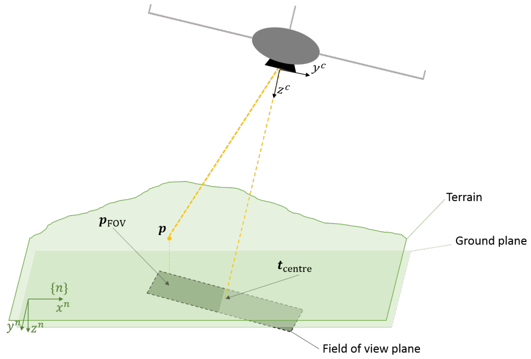


Fig. 18. Features on the surface of the terrain are chosen based on the attitude and position of the UAV. A ray along the camera z-axis intersects the ground plane at t_{centre}. A plane denoted field of view (FOV) is constructed perpendicular to the ray. FOV features are distributed along the FOV plane. Features are constructed with z^n-component from the elevation profile and x^n, y^n coordinate from the corresponding FOV feature. Features p are projected onto the image plane by the pinhole camera model to find the image plane coordinate $x^m = [r, s]^T$. This is done at time t_k and t_{k+1} with the same features, p, to get the discrete OF dr and ds.

The feature at time t_{k-1} and t_k can then be projected onto the image plane by the pinhole camera model from (1), yielding $x^m(t_k) = [r(t_k), s(t_k)]^T$ and $x^m(t_{k-1}) = [r(t_{k-1}), s(t_{k-1})]^T$. The discrete OF can then be found as $dr = r(t_k) - r(t_{k-1})$ and $ds = s(t_k) - s(t_{k-1})$.

Fig. 18 illustrates the relationship between the "centre ground point" t_{centre}, "FOV features" p_{FOV}, and "features" p.

Appendix B

Accepted Paper for the ICUAS'15 Conference

The following paper was submitted and accepted for The 2015 International Conference on Unmanned Aircraft Systems (ICUAS), and will be published 12th of June 2015 in Denver, Colorado, US.

Experimental Validation of a Uniformly Semi-globally Exponentially Stable Non-linear Observer for GNSS- and Camera-aided Inertial Navigation for Fixed-wing UAVs

Lorenzo Fusini, Jesper Hosen, Håkon H. Helgesen, Tor A. Johansen
and Thor I. Fossen
Norwegian University of Science and Technology
Centre for Autonomous Operations and Systems
Department of Engineering Cybernetics
Trondheim, Norway
E-mail: {lorenzo.fusini, tor.arne.johansen}@itk.ntnu.no, thor.fossen@ntnu.no,
{jesperho, hakonhhe}@stud.ntnu.no

Abstract—**This paper provides experimental validation of a uniformly semi-globally exponentially stable (USGES) non-linear observer for estimation of attitude, gyro bias, position, velocity and acceleration of a fixed-wing Unmanned Aerial Vehicle (UAV). The available sensors are an Inertial Measurement Unit (IMU), a Global Positioning System (GPS) receiver, a video camera, and an inclinometer. The UAV is flown with the sensor payload and all data is stored locally on a hard drive, which is recovered at the end of the flight. The non-linear observer is then tested offline with the recorded sensor data. An optical flow algorithm is used to calculate the UAV velocity based on the camera images, which is used as a reference vector of the body-fixed velocity in the attitude observer. The results are compared with an Extended Kalman Filter (EKF) and illustrate that the estimates of the unmeasured states converge accurately to the correct values, and that the estimates of the measured states have less noise than the measurements.**

I. Introduction

The estimation of position, velocity, and attitude of a vehicle at any time is commonly referred to as ”navigation”. The most used tool for this purpose has been the EKF, but in the last decades researchers have started to investigate new solutions, alternative to the Kalman filter, to the navigation problem, namely by developing non-linear observers with complete stability proofs and experimental validation. Non-linear observers have the advantage, over the EKF, of featuring a smaller computational footprint and often being globally exponentially stable (GES), a result that renders the observers robust to disturbances and initialization uncertainties. The problem of attitude estimation has received significant attention as a stand-alone problem [1]–[11]. In addition to this, other researchers have integrated Inertial Navigation System (INS), magnetometer/compass and GNSS to estimate the navigation states of a vehicle.

In [12] the authors expanded the vector-based observer proposed by [6] and [7] to include GNSS velocity measurements. [1] and [2] built globally exponentially stable (GES) attitude estimators based on multiple time-varying reference vectors or a single persistently exiting vector. A similar observer was developed in [13] and [14] to include also gyro bias and GNSS integration. An extension of this [15] replaced the rotation matrix with the unit quaternion for representing attitude, considered Earth rotation and curvature, a non-constant gravity vector, and included accelerometer bias estimation.

Another sensor commonly used in navigation is the camera. Low weight, low power consumption, and a wide range of machine vision software make it a viable choice for navigation purposes. Some drawbacks are its dependence on lighting and weather conditions, which directly affect the availability of features in the scene, and the difficulty in separating camera motion from moving objects in complex non-stationary environments.

Optical flow (OF) is how features in an image plane move between two consecutive images, caused by relative motion between the camera and the object being depicted. In the simplest case it could be understood as the pixel displacement of a single feature between two successive images. The OF can be represented as multiple vectors describing the change in the image plane in time. Several methods exists for determining the OF of a series of images, e.g. [16]–[19].

Machine vision and OF have been used for different applications in UAV navigation including indoor manoeuvring [20], [21], linear and angular velocity estimation [22]–[24], and obstacle avoidance [20], [25]–[29] as well as height above the ground estimation in [30]. [31], [32] use OF in assisting a landing of a UAV independent of external sensor inputs. OF from a single camera is used in [33], [34] to estimate body axes angular rates of an aircraft as well as wind-axes angles. [24], [35], [36] have used OF as input in Kalman filter-based navigation systems, fusing OF measurements with acceleration and angular velocity measurements. [37], [38] have used camera as sensor for navigating in GPS-denied environments.

A comparison of the performance of different methods of estimating the attitude of UAV based on machine vision is presented in [39], and different OF algorithms are evaluated in [23], [40] by estimating UAV velocity.

In [41] OF vectors are used to calculate the normalized body-fixed velocity of the UAV, and fed into the non-linear observer as a reference vector.

A. Contribution of this Paper

This paper provides experimental tests of a USGES non-linear observer for estimation of attitude, gyro bias, position, velocity, and acceleration of a fixed-wing UAV [41]. Exponential stability guarantees strong convergence and robustness properties, hence it is an important property to have in systems that are exposed to disturbances and uncertain initialization. The camera can sometimes replace the magnetometers: in small UAVs the magnetometers are heavily affected by disturbances and noise generated by the engine, while the camera is not conditioned by this.

The sensor data are logged during the UAV flight and used offline on a PC to test the observer. An OF algorithm is used to calculate the body-fixed velocity of the UAV based on the camera images, altitude and inclinometer data. To demonstrate their validity, the estimated states are compared with those evaluated via the EKF.

II. Notation and Preliminaries

Vectors and matrices are represented by lowercase and uppercase letters respectively. X^{-1}, X^{+}, and $\mathrm{tr}(X)$ denote the inverse, pseudoinverse, and trace of a matrix respectively, X^T the transpose of a matrix or vector, $\hat{X}$ the estimated value of X, and $\tilde{X} = X - \hat{X}$ the estimation error. $\|\cdot\|$ denotes the Euclidean norm, I_n the identity matrix of order n, and $0_{m\times n}$ the $m \times n$ matrix of zeros. A vector $x = [x_1, x_2, x_3]^T$ is represented in homogeneous coordinates as $\underline{x} = [x_1, x_2, x_3, 1]^T$. The function $\mathrm{sat}(\cdot)$ performs a component-wise saturation of its vector or matrix argument to the interval $[-1, 1]$. The operator $S(x)$ transforms the vector x into the skew-symmetric matrix

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

The inverse operation is denoted as $\mathrm{vex}(\cdot)$, such that $\mathrm{vex}(S(x)) = x$. For a square matrix A, its skew-symmetric part is represented by $\mathbb{P}_a(A) = \frac{1}{2}(A - A^T)$.

Two reference frames are considered in the paper: the body-fixed frame {B} and the North-East-Down (NED) frame {N} (Earth-fixed, considered inertial). The rotation from frame {B} to {N} is represented by the matrix $R_b^n \equiv R \in SO(3)$, with $SO(3)$ representing the Special Orthogonal group.

A vector decomposed in {B} and {N} has superscript ^{b} and ^{n} respectively. The camera location w.r.t. {N} is described by $c^n = [c_x^n, c_y^n, c_z^n]^T$. A point in the environment expressed w.r.t. {N} is $t^n = [x^n, y^n, z^n]^T$: note that a point located on the ground corresponds to $z^n = 0$ and such it will be throughout the paper. The same point expressed w.r.t. {B} is $t^b = [x^b, y^b, z^b]^T$. It will also be assumed that every point is fixed w.r.t. {N}. The greek letters ϕ, θ, and ψ represent the roll, pitch, and yaw angles respectively, defined according to the zyx convention for principal rotations [42], and they are collected in the vector $\Theta = [\phi, \theta, \psi]^T$. A 2-D camera image has coordinates $[r, s]^T$, aligned with the y^b- and x^b-axis respectively (see Fig. 2). The derivative $[\dot{r}, \dot{s}]^T$ of the image coordinates is the OF. The subscript $_F$ indicates a quantity evaluated by means of the OF.

A. *Measurements and Sensors*

The sensor payload consists of an IMU, a GPS receiver, a video camera, and an inclinometer, providing the following information:

- *GPS*: NED position p^n and, by differentiation, NED velocity v^n;
- *IMU*: biased angular velocity $\omega_m^b = \omega^b + b^b$, where b^b represent the gyro bias, and acceleration a^b;
- *camera*: 2-D projections $[r, s]^T$ onto the image plane of points $[x^n, y^n, z^n]^T$ from the 3-D world;
- *inclinometer*: roll ϕ and pitch θ angles.

Further information on the actual sensors employed in the experiment is presented in Section V.

III. Optical Flow

The observer presented in Section IV depends on velocity measurements from the on-board camera decomposed in the body-fixed frame. These measurements are generated from OF, therefore it is necessary to compute the OF vectors from consecutive images before these vectors are transformed to velocity measurements. The OF calculation and the transformation is presented in the forthcoming section.

A. *Optical flow computation*

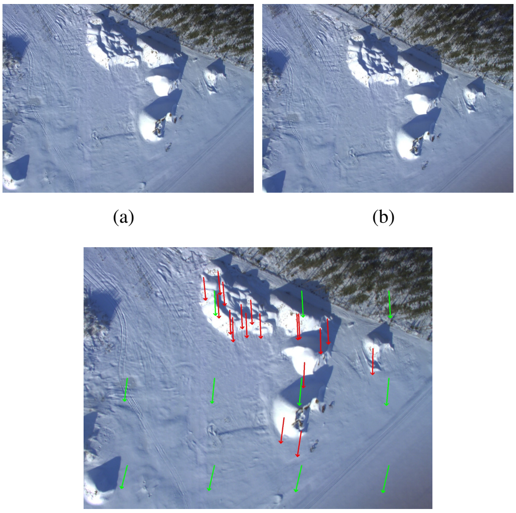
There exist several methods for computing OF. For the experiment presented in Section V two specific methods are combined. The first one is SIFT [18] which provided the overall best performance in [23]. SIFT uses a feature-based approach to compute OF. The total number of OF vectors in each image depends on the number of features detected and matched together. Since the transformation in Section III-B requires at least three OF vectors [41], it is necessary to make sure that this is handled. It is not possible to guarantee three OF vectors with SIFT since homogeneous environments, like snow or the ocean, increase the difficulty of finding distinct features. Therefore the OF vectors created by SIFT are combined with a second method, which is based on region matching [43].

The region matching method used here is a template matching approach based on normalized cross-correlation [44]. The displacements of twelve templates, created symmetrically across the images, are used to find twelve OF vectors. Template matches below a given threshold are discarded and the corresponding OF vector removed. Unreliable matches can occur in case of uniform terrain, changes in brightness or simply when the area covered by the template has disappeared from the image in the time between the capture of images. An example of OF vectors computed with SIFT and template matching is displayed in Fig. 4.

In case of mismatches, both methods will create erroneous OF vectors. It is desired to locate and remove these vectors. Therefore an outlier detector is implemented before the vectors are used to calculate body-fixed velocities. The outlier detector utilizes a histogram to find the vectors that deviate from the mean with respect to direction and magnitude.

B. *Transformation from optical flow to velocity*

For the OF computations to be useful in the observer, a transformation to body-referenced veloc-



(a)

(b)

(c)

Fig. 1. a) Image captured at time t_0. b) Image captured at time $t_0+\Delta t$. c) Optical flow vectors between image a) and b), generated by SIFT (red) and Template Matching (green).

ity is necessary. The transformation is motivated by [41] and the pinhole camera model is used [45]. The camera-fixed coordinate system is related to the body-fixed coordinate system through Fig. 2, where the downward-looking camera is aligned with the body z-axis. The focal point of the camera is for simplicity assumed to coincide with the origin of {B}.

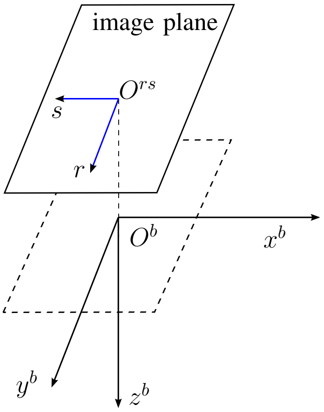


Fig. 2. Pinhole camera model. The camera is oriented downwards, while x^b is the direction of flight.

It is necessary to relate a point in the terrain expressed in {N} t^n to {B}, since points in the terrain are used to compute the body-referenced velocity. The matrix R_b^n and vector c^n represent a rotation and a translation between {N} and {B}. They can be merged to form a homogeneous 4×4-transformation T_b^n

$$T_b^n(\Theta) = \begin{bmatrix} R_b^n(\Theta) & c^n \\ 0_{1\times3} & 1 \end{bmatrix}$$

such that $\underline{t}^n = T_b^n(\Theta)\underline{t}^b$ where $\underline{t}^n$ and $\underline{t}^b$ is the same homogeneous point represented in {N} and {B} respectively. The transformation can be inverted to find t^b from t^n

$$\underline{t}^b = T_b^n(\Theta)^{-1}\underline{t}^n = \begin{bmatrix} R_b^n(\Theta)^T & -R_b^n(\Theta)^T c^n \\ 0_{1\times3} & 1 \end{bmatrix} \underline{t}^n \tag{1}$$

and t^b is now a function of x^n, y^n, z^n, c_x^n, c_y^n, c_z^n, ϕ, θ, ψ.

The relationship between t^b and its projection onto the image plane is expressed by the well-known pinhole camera model [45] [46].

$$\begin{bmatrix} r \\ s \end{bmatrix} = \frac{f}{z^b} \begin{bmatrix} y^b \\ -x^b \end{bmatrix}, z^b \neq 0 \tag{2}$$

where f is the focal length of the camera. As t^b in itself is not available, the relationship in (1) is used to express t^b in (2) as

$$\begin{bmatrix} x^b \\ y^b \\ z^b \end{bmatrix} = \begin{bmatrix} \frac{sc_z^n}{s\sin(\theta)+\cos(\theta)(f\cos(\phi)+r\sin(\phi))} \\ -\frac{rc_z^n}{s\sin(\theta)+\cos(\theta)(f\cos(\phi)+r\sin(\phi))} \\ -\frac{fc_z^n}{s\sin(\theta)+\cos(\theta)(f\cos(\phi)+r\sin(\phi))} \end{bmatrix} \tag{3}$$

All features tracked by the camera are assumed to be stationary with respect to {N}. Hence the UAV linear and angular velocity relative to a feature tracked by the OF algorithm, v_F^b and ω_F^b, will be equal for every tracked feature. Furthermore it is assumed that the terrain is flat, such that every feature is located at the same altitude. For every feature j, the relationship between OF and body-fixed linear/angular velocity is given as

$$\begin{gathered} \begin{bmatrix} \dot{r}_j \\ \dot{s}_j \end{bmatrix} = -M_j(f, r_j, s_j, \phi, \theta, c_z^n) \begin{bmatrix} v_F^b \\ \omega_F^b \end{bmatrix} \\ M_j = \frac{f}{z_j^b} \begin{bmatrix} 0 & 1 & -\frac{y_j^b}{z_j^b} & -\frac{{y_j^b}^2}{z_j^b} - z_j^b & \frac{y_j^b x_j^b}{z_j^b} & x_j^b \\ -1 & 0 & \frac{x_j^b}{z_j^b} & \frac{x_j^b y_j^b}{z_j^b} & -\frac{{x_j^b}^2}{z_j^b} - z_j^b & y_j^b \end{bmatrix} \end{gathered} \tag{4}$$

where $M_j \in \mathcal{R}^{2\times 6}$ in (4) is motivated by [41]. If the number of features being tracked is k, then the OF vector has dimension $2k$. A matrix $M \in \mathcal{R}^{2k\times 6}$ might be created by concatenating the matrices $M_j, j = 1 \ldots k$, and the following relationship is obtained

$$\begin{bmatrix} \dot{r_1} \\ \dot{s_1} \\ \vdots \\ \dot{r_k} \\ \dot{s_k} \end{bmatrix} = -M \begin{bmatrix} v_F^b \\ \omega_F^b \end{bmatrix}, M = \begin{bmatrix} M_1 \\ \vdots \\ M_k \end{bmatrix} \tag{5}$$

By calculating the pseudoinverse of M in (5) the angular and linear velocity can be computed as

$$\begin{bmatrix} v_F^b \\ \omega_F^b \end{bmatrix} = -M^+ \begin{bmatrix} \dot{r_1} \\ \dot{s_1} \\ \vdots \\ \dot{r_k} \\ \dot{s_k} \end{bmatrix} \tag{6}$$

M^+ exists only if M^TM has full rank. This can only happen if the number of flow vectors are greater or equal to three. This is always the case in the experiment.

IV. Observer Design

A. Dynamic System

The dynamics of attitude, position, and velocity is described by

$$\dot{R} = RS(\omega^b) \tag{7a}$$

$$\dot{p^n} = v^n \tag{7b}$$

$$\dot{v}^n = a^n + g^n \tag{7c}$$

The objective is to estimate the attitude R, the position p^n, and the velocity v^n with exponential convergence rate. In addition to this, an estimator for the gyro bias b^b is also provided.

B. Assumptions

The observer design is based on the following assumptions:

Assumption 1: the OF algorithm uses a sufficient number of image features, such that M has full rank and Eq. (6) can be used.

Assumption 2: the gyro bias b^b is constant.

Assumption 3: there exists a constant $c_{obs} > 0$ such that, $\forall t \geq 0,\ \|v_F^b \times a^b\| \geq c_{obs}$.

Assumption 3 is a condition of non-collinearity for the vectors v_F^b and a^b, i.e. the angle between them is non-zero and none of them can be identically zero (see, e.g., [12], [6]). For a fixed-wing UAV this means that the observer cannot work while the vehicle stands still on the ground, but presents no issues during flight.

C. Observer Equations

The full observer was introduced in [41] as

$$\Sigma_1 \begin{cases} \dot{\hat{R}} = \hat{R}S(\omega_m^b - \hat{b}^b) + \sigma K_P \hat{J} \\ \dot{\hat{b}}^b = \text{Proj}(\hat{b}^b, -k_I \text{vex}(\mathbb{P}_a(\hat{R}_s^T K_P \hat{J}))) \end{cases} \tag{8}$$

$$\Sigma_2 \begin{cases} \dot{\hat{p}}^n = \hat{v}^n + K_{pp}(p^n - \hat{p}^n) + K_{pv}(v^n - \hat{v}^n) \\ \dot{\hat{v}}^n = \hat{a}^n + g^n + K_{vp}(p^n - \hat{p}^n) + K_{vv}(v^n - \hat{v}^n) \\ \dot{\xi} = -\sigma K_P \hat{J} a^b + K_{\xi p}(p^n - \hat{p}^n) + K_{\xi v}(v^n - \hat{v}^n) \\ \hat{a}^n = \hat{R}a^b + \xi \end{cases} \tag{9}$$

$$\text{OF} \left\{ \begin{bmatrix} \hat{v}_F^b \\ \hat{\omega}_F^b \end{bmatrix} = -\hat{M}^+ \begin{bmatrix} \dot{r} \\ \dot{s} \end{bmatrix} \right. \tag{10}$$

The subsystem Σ_1 represents the attitude observer, whereas Σ_2 represents the translational motion observer. In addition, (10) is given by machine vision. $\sigma \geq 1$ is a scaling factor tuned to achieve stability, k_I is a positive scalar gain, $\text{Proj}(\cdot,\cdot)$ represents a parameter projection [47] that ensures that $\|\hat{b}^b\|$ does not exceed a design constant $L_{\hat{b}^b} > L_{b^b}$ (see Appendix), and $\hat{R}_s = \text{sat}(\hat{R})$. The matrix $\hat{J}$ is the output injection term, whose design is inspired by the TRIAD algorithm [48] and defined

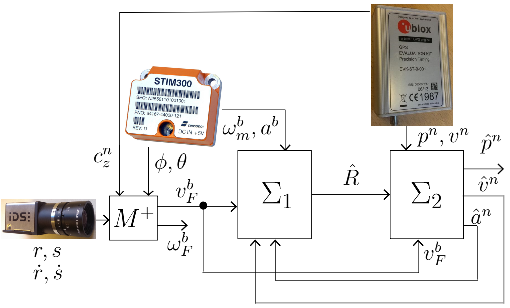


Fig. 3. Block diagram of the observer. Σ_1 represents the attitude observer, and Σ_2 the translational motion observer.

as

$$\hat{J}(v_F^b,\ \hat{v}^n, a^b, \hat{a}^n, \hat{R}) := \hat{A}_n A_b^T - \hat{R} A_b A_b^T \quad (11a)$$

$$A_b := [v_F^b,\ v_F^b \times a^b,\ v_F^b \times (v_F^b \times a^b)] \quad (11b)$$

$$\hat{A}_n := [\hat{v}^n,\ \hat{v}^n \times \hat{a}^n,\ \hat{v}^n \times (\hat{v}^n \times \hat{a}^n)] \quad (11c)$$

The subsystem Σ_2 represents the translational motion observer, where $K_{pp}, K_{pv}, K_{vp}, K_{vv}, K_{\xi p}$, and $K_{\xi v}$ are observers gains yet to be defined, and $g^n = [0, 0, 9.81]^T$ is the gravity vector in NED.

The system Σ_1–Σ_2 is a feedback interconnection, as illustrated by Fig. 3.

D. Stability Proof

The error dynamics of the non-linear observer can be written in a compact form as

$$\Sigma_1 \begin{cases} \dot{\tilde{R}} = RS(\omega^b) - \hat{R}S(\omega_m^b - \hat{b}^b) - \sigma K_P \hat{J} \\ \dot{\tilde{b}}^b = -\text{Proj}(\hat{b}^b, \tau(\hat{J})) \end{cases} \quad (12a)$$

$$\Sigma_2 \left\{ \dot{\tilde{w}} = (A - KC)\tilde{w} + B\tilde{d} \right. \quad (12b)$$

where $\tilde{w} = [(\tilde{p}^n)^T, (\tilde{v}^n)^T, (\tilde{a}^n)^T]^T$ collects the estimated position, velocity and acceleration vectors, $\tilde{d} = (RS(\omega^b) - \hat{R}S(\omega_m^b - \hat{b}^b))a^b + (R - \hat{R})\dot{a}^b$, and the four matrices in (12b) are defined as

$$A = \begin{bmatrix} 0_{6\times3} & I_6 \\ 0_{3\times3} & 0_{3\times6} \end{bmatrix}, \qquad B = \begin{bmatrix} 0_{6\times3} \\ I_3 \end{bmatrix},$$

$$C = \begin{bmatrix} I_6 & 0_{6\times3} \end{bmatrix}, \qquad K = \begin{bmatrix} K_{pp} & K_{pv} \\ K_{vp} & K_{vv} \\ K_{\xi p} & K_{\xi v} \end{bmatrix}.$$

Theorem 1 provides conditions that ensure USGES of the origin of the error dynamics (12).

Theorem 1: Let σ be chosen to ensure stability according to Lemma 1 in [13] and define $H_K(s) = (Is - A + KC)^{-1}B$. There exists a set $(0, c)$ such that, if K is chosen such that $A - KC$ is Hurwitz and $\|H_K(s)\|_\infty < \gamma$, for $\gamma \in (0, c)$, then the origin of the error dynamics (12) is USGES as defined by [49] when the initial conditions satisfy $\|\hat{b}^b(0)\| \leq L_{\hat{b}^b}$.

Proof: For the proof, see [41]. ■

V. Experimental Results

This section describes the experiment carried out to gather the necessary data and the results obtained with the non-linear observer and OF.

A. Setup

The UAV employed is a UAV Factory Penguin-B, equipped with a custom-made payload that includes all the necessary sensors. The IMU is a Sensonor STIM300, a low-weight, tactical grade, high-performance sensor that includes gyroscopes, accelerometers, and inclinometers, all recorded at a frequency of 300 Hz. The chosen GPS receiver is a uBlox LEA-6T, which gives measurements at 5 Hz. The video camera is an IDS GigE uEye 5250CP provided with an 8mm lens. The camera is configured for a hardware-triggered capture at 10 Hz: the uBlox sends a digital pulse-per-second signal whose rising edge is accurately synchronized with the time of validity of the recorded GPS position, which guarantees that the image capture is synchronized with the position measurements. The experiment has been carried out on 6 February 2015 at the Eggemoen Aviation and Technology Park, Norway, in a sunny day with good visibility, very little wind, an air temperature of about -8°C. The terrain is covered with snow and flat enough to let all features be considered as lying at zero altitude.

The observer is implemented using forward Euler discretization with a time-varying step depending on the interval of data acquisition of the fastest sensor, namely the STIM300, and it is typically around 0.003 seconds. The various parameters and gains are chosen as $L_{b^b} = 2^\circ/\text{s}$, $L_{\hat{b}^b} = 2.1^\circ/\text{s}$, $\sigma = 1$, $K_P = \text{diag}[0.1, 0.1, 0.5]$, $k_I = 0.006$, $K_{pp} = 30I_3$, $K_{pv} = 2I_3$, $K_{vp} = I_3$, $K_{vv} = 100I_3$, $K_{\xi p} = I_3$, and $K_{\xi v} = 50I_3$. All the gains are obtained by running the observer several times and correcting the gains until a satisfactory performance was achieved. Concerning the gains of the translational motion observer, it is also possible to tune them with the help of a linear matrix inequality formulation that allows $\|H_K(s)\|_\infty$ to satisfy the conditions of Theorem 1 (see [13], [50] for details).

The reference provided for the attitude, position, and velocity is the output of the EKF of the autopilot mounted on the Penguin-B. An exact reference for the gyro bias is not available, but an approximation of the real value is calculated by averaging the gyro measurements at standstill before and after the flight. The accelerometer bias is not estimated, but it is computed the same way as the gyro bias and subtracted from the accelerometers measurements before being used in the observer.

All the images are processed with a resolution of 1600×1200 (width×height) pixels and in their original state, without any filtering. The lens distortion of the camera is not accounted for, and no correction is applied to the images. SIFT is implemented with the open source computer vision library (OpenCV) [51] with default settings. Each match is tagged with a value indicating the accuracy of the match, and the smallest of these values is considered to be the best match. To increase the reliability of the OF vectors, each match is compared to the best one. Every match with an uncertainty more than double the uncertainty of the best match is removed. Also the template matching algorithm is implemented with OpenCV. The size of the templates is chosen to be 120×90 pixels and a correlation of 99% is required in order for a template match to be considered reliable and not removed.

B. Results

The results here presented refer to a complete flight of the Penguin-B, from take-off to landing, which correspond to a travelled distance of approximately 9 km in around 5 min. The time on the x-axes is the elapsed time since the data logging begins, and only the significant part involving the flight is represented. The manoeuvres performed include flights on a straight line and turns with a large and small radius of curvature, namely approximately 200 m and 100 m, as it can be noticed from Fig. 6. The body velocity calculated via the OF is represented in Fig. 4 and is the result of (6). The estimated attitude, position, and velocity are illustrated in Fig. (5)–(7): it is clear that the observer (blue, solid line) performs well when compared to the EKF (red, dashed line). The pitch estimate presents some deviation, probably due to a misalignment of the sensor on the payload. The estimated North and East velocities show some small peaks at around 1130–1150 s and 1240 s, which are due to the presence of outliers in the GPS data. The estimated gyro bias is presented in Fig. 8: the estimates do not converge as well as the other states, but they remain within 0.2° of their initial estimate.

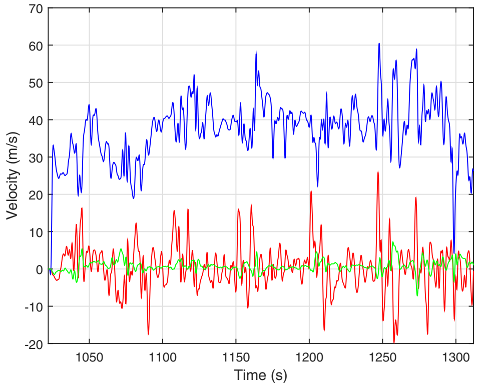


Fig. 4. Body-fixed velocity in the x, y, and z axis (blue, red, and green, respectively) calculated via machine vision.

VI. Conclusions

In this paper a USGES non-linear observer has been tested on experimental data obtained by flying a fixed-wing UAV with a custom-made payload of sensors. An OF algorithm has been employed to calculate the body-referenced velocity of the vehicle by means of IMU, camera

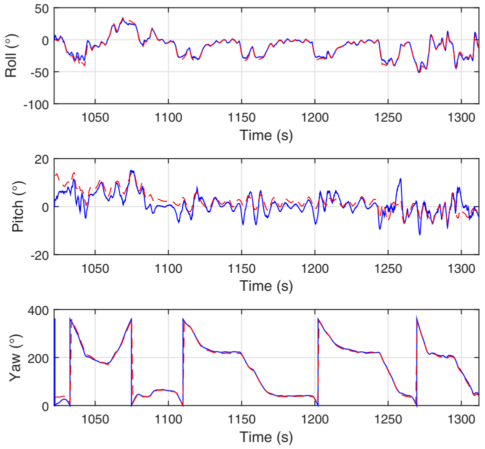


Fig. 5. Estimated (blue, solid) and EKF (red, dashed) attitude.

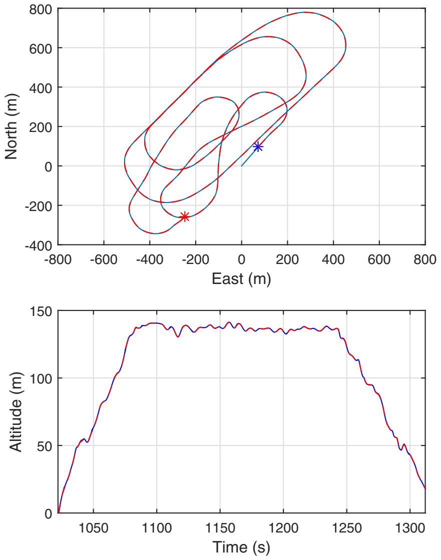


Fig. 6. Estimated (blue, solid) and EKF (red, dashed) position on the N-E plane and altitude. The blue and red star indicate the start and end of the data set, respectively.

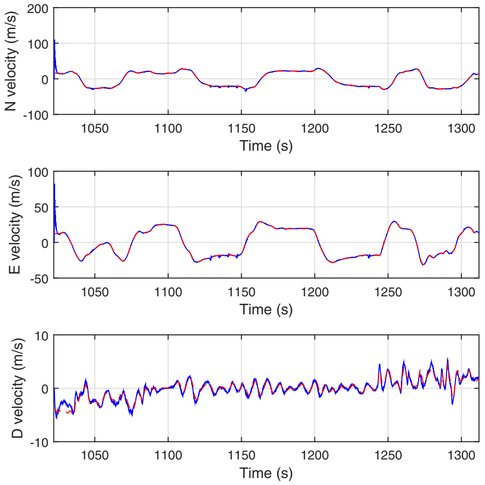


Fig. 7. Estimated (blue, solid) and EKF (red, dashed) NED velocities.

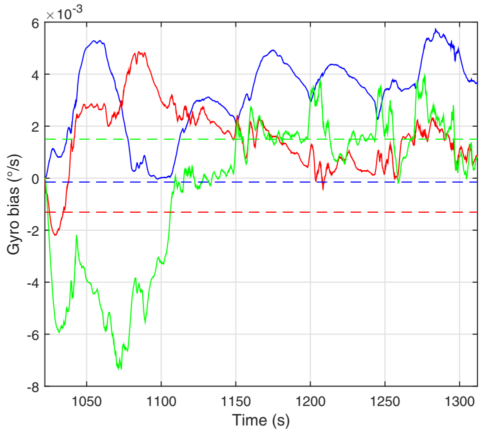


Fig. 8. Gyro bias calculated at standstill (dashed) and estimated by the observer (solid). The x, y, and z components are blue, red, and green, respectively.

images and GPS measurements. Such velocity, accelerometers measurements, estimated NED velocity, and estimated NED acceleration have been used as reference vectors in the injection term of the observer in order to provide the desired estimates. The results presented for different types of manoeuvres confirm the validity of the analysis and design.

Acknowledgements

This work has been carried out at the Centre for Autonomous Marine Operations and Systems (AMOS), supported by the Research Council of Norway through the Centres of Excellence funding scheme, grant no. 223254, and the FRINATEK grant 221666. The Norwegian Research Council is acknowledged as the main sponsor of AMOS. The authors are grateful for the assistance provided by the UAV engineers at NTNU and Maritime Robotics AS, in particular Lars Semb and Carl Erik Stephansen. Significant contributions to the construction of the UAV payload was made by the rest of the navigation team at NTNU, in particular Sigurd M. Albrektsen, Jakob M. Hansen and Kasper T. Borup.

REFERENCES

[1] P. Batista, C. Silvestre, and P. Oliveira, "GES attitude observers - Part I: Multiple general vector observations," *Proceedings of the 18th IFAC World Congress*, vol. 18, pp. 2985–2990, 2011.

[2] ——, "GES attitude observers - Part II: Single vector observations," *Proceedings of the 18th IFAC World Congress*, vol. 18, pp. 2991–2996, 2011.

[3] H. Grip, T. Fossen, T. Johansen, and A. Saberi, "Attitude estimation using biased gyro and vector measurements with time-varying reference vectors," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1332–1338, 2012.

[4] J. Guerrero-Castellanos, H. Madrigal-Sastre, S. Durand, L. Torres, and G. Muñoz-Hernández, "A robust nonlinear observer for real-time attitude estimation using low-cost MEMS inertial sensors," *Sensors*, vol. 13, no. 11, pp. 15 138–15 158, 2013.

[5] M. Hua, G. Ducard, T. Hamel, R. Mahony, and K. Rudin, "Implementation of a nonlinear attitude estimator for aerial robotic vehicles," *IEEE Transactions on Control System Technology*, vol. 22, no. 1, pp. 201–213, 2014.

[6] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[7] R. Mahony, T. Hamel, J. Trumpf, and C. Lageman, "Nonlinear attitude observers on SO(3) for complementary and compatible measurements: A theoretical study," *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 6407–6412, 2009.

[8] R. Mahony, M. Euston, J. Kim, P. Coote1, and T. Hamel, "A non-linear observer for attitude estimation of a fixed-wing unmanned aerial vehicle without GPS measurements," *Transactions of the Institute of Measurement and Control*, vol. 33, no. 6, pp. 699–717, 2011.

[9] J. Trumpf, R. Mahony, T. Hamel, and C. Lageman, "Analysis of non-linear attitude observers for time-varying reference measurements," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2789–2800, 2012.

[10] S. Salcudean, "A globally convergent angular velocity observer for rigid body motion," *IEEE Transactions on Automatic Control*, vol. 36, no. 12, pp. 1493–1497, 1991.

[11] J. Thienel and R. Sanner, "A coupled nonlinear spacecraft attitude controller and observer with an unknown constant gyro bias and gyro noise," *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 2011–2015, 2003.

[12] M. Hua, "Attitude estimation for accelerated vehicles using GPS/INS measurements," *Control Engineering Practice*, vol. 18, no. 7, pp. 723–732, 2010.

[13] H. Grip, T. Fossen, T. Johansen, and A. Saberi, "A nonlinear observer for integration of GNSS and IMU measurements with gyro bias estimation," *American Control Conference (ACC)*, pp. 4607–4612, 2012.

[14] ——, "Globally exponentially stable attitude and gyro bias estimation with application to GNSS/INS integration," *Automatica*, vol. 51, pp. 158–166, 2015.

[15] ——, "Nonlinear observer for GNSS-aided inertial navigation with quaternion-based attitude estimation," *American Control Conference (ACC)*, pp. 272–279, 2013.

[16] B.Lucas and T.Kanade, "An iterative image restoration technique with an application to stereo vision," *Proc. DARPA Image Underst, Workshop*, pp. 121–130, 1981.

[17] B.K.P.Horn and B.G.Schunk, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–204, 1981.

[18] D. Lowe, "Object recognition from local scale-invariant features," *Proc. Int. Conf. Computer Vision*, pp. 1150–1157, 1999.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[20] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "Mav navigation through indoor corridors using optical flow," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3361–3368, 2010.

[21] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 20–25, 2011.

[22] D. Dusha, W. Boles, and R. Walker, "Attitude estimation for a fixed-wing aircraft using horizon detection and optical flow," *Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, pp. 485–492, 2007.

[23] M. Mammarella, G. Campa, M. Fravolini, and M. Napolitano, "Comparing optical flow algorithms using 6-dof motion of real-world rigid objects," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1752 – 1762, 2012.

[24] S. Weiss, R. Brockers, and L. Matthies, "4DoF drift free navigation using inertial cues and optical flow," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4180–4186, 2013.

[25] J. C. Zufferey and D. Floreano, "Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, pp. 2594–2599, 2005.

[26] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 302–309, 2005.

[27] P. C. Merrell, D.-J. Lee, and R. W. Beard, "Obstacle avoidance for unmanned air vehicles using optical flow probability distributions," *Society of Photo-Optical Instrumentation*

Engineers (SPIE) Conference Series, vol. 5609, no. 1, pp. 13–22, 2004.

[28] J. Conroy, G. Gremillion, B. Ranganathan, and J. Humbert, “Implementation of wide-field integration of optic flow for autonomous quadrotor navigation,” *Autonomous Robots*, vol. 27, no. 3, pp. 189–198, 2009.

[29] F. Ruffier and N. Franceschini, “Visually guided micro-aerial vehicle: Automatic take off, terrain following, landing and wind reaction,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 3, pp. 2339–2346, 2004.

[30] P. C. Merrell, D.-J. Lee, and R. W. Beard, “Statistical analysis of multiple optical flow values for estimation of unmanned aerial vehicle height above ground,” *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 5608, pp. 298–305, 2004.

[31] R. Brockers, S. Susca, D. Zhu, and L. Matthies, “Fully self-contained vision-aided navigation and landing of a micro air vehicle independent from external sensor inputs,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8387, 2012.

[32] B. Herisse, F.-X. Russotto, T. Hamel, and R. Mahony, “Hovering flight and vertical landing control of a vtol unmanned aerial vehicle using optical flow,” *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 801–806, 2008.

[33] J. Kehoe, A. Watkins, R. Causey, and R. Lind, “State estimation using optical flow from parallax-weighted feature tracking,” *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, vol. 8, pp. 5030–5045, 2006.

[34] R. J. D. Moore, S. Thurrowgood, and M. V. Srinivasan, “Vision-only estimation of wind field strength and direction from an aerial platform,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 4544–4549, 2012.

[35] “Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 957–964, 2012.

[36] D. Mercado, G. Flores, P. Castillo, J. Escareno, and R. Lozano, “Gps/ins/optic flow data fusion for position and velocity estimation,” *International Conference on Unmanned Aircraft Systems, ICUAS - Conference Proceedings*, pp. 486–491, 2013.

[37] M. Bibuli, M. Caccia, and L. Lapierre, “Path-following algorithms and experiments for an autonomous surface vehicle,” *Control Applications in Marine Systems*, vol. 7, no. 1, pp. 81–86, 2007.

[38] S. Ahrens, D. Levine, G. Andrews, and J. P. How, “Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2643–2648, 2009.

[39] A. Shabayek, C. Demonceaux, O. Morel, and D. Fofi, “Vision based uav attitude estimation: Progress and insights,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 65, no. 1-4, pp. 295–308, 2012.

[40] H. Chao, Y. Gu, and M. Napolitano, “A survey of optical flow techniques for robotics navigation applications,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 73, no. 1-4, pp. 361–372, 2014.

[41] L. Fusini, T. Fossen, and T. Johansen, “A uniformly semiglobally exponentially stable nonlinear observer for GNSS- and camera-aided inertial navigation,” *22nd Mediterranean Conference of Control and Automation (MED)*, pp. 1031–1036, 2014.

[42] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, UK: John Wiley & Sons, Ltd, 2011.

[43] C. S. Fuh and P. Maragos, “Region-based optical flow estimation,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Proceedings CVPR.*, pp. 130–135, 1989.

[44] J. Sarvaiya, S. Patnaik, and S. Bombaywala, “Image registration by template matching using normalized cross-correlation,” *International Conference on Advances in Computing, Control and Telecommunication Technologies*, pp. 819–822, 2009.

[45] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[46] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Stamford, CT: Cengage Learning, 2008.

[47] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: Wiley, 1995.

[48] M. Shuster and S. Oh, “Three-axis attitude determination from vector observations,” *Journal of Guidance, Control and Dynamics*, vol. 4, no. 1, pp. 70–77, 1981.

[49] A. Loria and E. Panteley, “Cascaded nonlinear time-varying systems: analysis and design,” in *Advanced Topics in Control Systems Theory*. London: Springer-Verlag (F. Lamnabhi-Lagarrigue, A. Loria and E. Panteley Eds.), 2004, ch. 2, pp. 23–64.

[50] H. Grip, A. Saberi, and T. Johansen, “Observers for interconnected nonlinear and linear systems,” *Automatica*, vol. 48, no. 7, pp. 1339–1346, 2012.

[51] G. Bradski, “The opencv library,” *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120–126, 2000.

Appendix

The parameter projection Proj$(\cdot,\cdot)$ is defined as:

$$\text{Proj}(\hat{b}^b,\tau)=\begin{cases}\left(I-\frac{c(\hat{b}^b)}{\|\hat{b}^b\|^2}\hat{b}^b\hat{b}^{bT}\right)\tau, & \|\hat{b}^b\|\geq L_b,\ \hat{b}^{bT}\tau>0\\ \tau, & \text{otherwise}\end{cases}$$

where $c(\hat{b}^b)=\min\{1,(\|\hat{b}^b\|^2-L_b^2)/(L_{\hat{b}^b}^2-L_{b^b}^2)\}$. This operator is a special case of that from Appendix E of [47].

Appendix C

Derivation of the M-matrix

This appendix derives the M-matrix from Section 3.3.1, assuming that measurements of optical flow (OF), altitude and attitude are available. Assuming that any feature in the image, corresponding to an OF vector, are stationary w.r.t the terrain, the pinhole camera model from equation (3.2) can be differentiated with respect to time to get a relationship between OF and egomotion of the camera.

$$\begin{bmatrix} \dot{r} \\ \dot{s} \end{bmatrix} = \frac{f}{z^c} \begin{bmatrix} 0 & 1 & -\frac{y^c}{z^c} \\ -1 & 0 & \frac{x^c}{z^c} \end{bmatrix} \begin{bmatrix} \dot{x^c} \\ \dot{y^c} \\ \dot{z^c} \end{bmatrix} \tag{C.1}$$

The vector $[\dot{r}, \dot{s}]^T$ is the OF, and is a measure of how a feature in the image plane moves in horizontal (r) and vertical (s) direction in time. The vector $[\dot{x^c}, \dot{y^c}, \dot{z^c}]^T$ of the right hand side can be recognized as

$$\begin{aligned} \dot{p^c} &= \begin{bmatrix} \dot{x^c} \\ \dot{y^c} \\ \dot{z^c} \end{bmatrix} \\ &= \boldsymbol{v}^c_{T/c} + \left[\omega^c_{T/c}\right]_\times (p^c - o^c_T) \end{aligned} \tag{C.2}$$

where the following definitions are necessary:

Definition 9. $\boldsymbol{v}^c_{T/c}$ *is the linear velocity of the terrain with respect to the camera represented in the camera-fixed coordinate system.*

Definition 10. *$\omega^c_{T/c}$ is the angular velocity of the terrain with respect to the camera represented in the camera-fixed coordinate system.*

Definition 11. *o^c_T is the terrain origin/point of rotation in camera coordinates. All rotations of the terrain seen in the image will be rotations about the camera, hence the rotation point o^c_T coincides with the origin of the camera frame.*

(C.1) might be rewritten as

$$\begin{aligned}
\begin{bmatrix} \dot{r} \\ \dot{s} \end{bmatrix} &= \frac{f}{z^c} \begin{bmatrix} 0 & 1 & -\frac{y^c}{z^c} \\ -1 & 0 & \frac{x^c}{z^c} \end{bmatrix} \left(\boldsymbol{v}^c_{T/c} + \left[\omega^c_{T/c} \right]_\times (p^c - o^c_T) \right) && \text{(C.3)} \\
&= \frac{f}{z^c} \left[\begin{matrix} 0 & 1 & -\frac{y^c}{z^c} \\ -1 & 0 & \frac{x^c}{z^c} \end{matrix} \;\middle|\; \begin{matrix} 0 & 1 & -\frac{y^c}{z^c} \\ -1 & 0 & \frac{x^c}{z^c} \end{matrix} \right] \begin{bmatrix} \boldsymbol{v}^c_{T/c} \\ \left[\omega^c_{T/c} \right]_\times (p^c - o^c_T) \end{bmatrix} && \text{(C.4)} \\
&= \frac{f}{z^c} \left[A \;\middle|\; A \right] \begin{bmatrix} \boldsymbol{v}^c_{T/c} \\ \left[\omega^c_{T/c} \right]_\times (p^c - o^c_T) \end{bmatrix} && \text{(C.5)} \\
\boldsymbol{A} &= \begin{bmatrix} 0 & 1 & -\frac{y^c}{z^c} \\ -1 & 0 & \frac{x^c}{z^c} \end{bmatrix}
\end{aligned}$$

Using the properties of the crossproduct

$$\left[\omega^c_{T/c} \right]_\times (p^c - o^c_T) = -\left[(p^c - o^c_T) \right]_\times \omega^c_{T/c} \tag{C.6}$$

With $[\boldsymbol{v}]_\times$ being the skew-symmetric matrix

$$[\boldsymbol{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{C.7}$$

one may rewrite (C.5) as

$$\begin{aligned}
\begin{bmatrix} \dot{r} \\ \dot{s} \end{bmatrix} &= \frac{f}{z^c} \left[A \;\middle|\; A \right] \begin{bmatrix} \boldsymbol{v}^c_{T/c} \\ -\left[(p^c - o^c_T) \right]_\times \omega^c_{T/c} \end{bmatrix} \\
&= \frac{f}{z^c} \left[A \;\middle|\; -A \cdot \left[(p^c - o^c_T) \right]_\times \right] \begin{bmatrix} \boldsymbol{v}^c_{T/c} \\ \omega^c_{T/c} \end{bmatrix} && \text{(C.8)} \\
&= M'(f, p^c, o^c_T) \begin{bmatrix} \boldsymbol{v}^c_{T/c} \\ \omega^c_{T/c} \end{bmatrix}
\end{aligned}$$

$$M'(f, p^c, o_T^c) = \frac{f}{z^c} \begin{bmatrix} 0 & 1 & -\frac{y^c}{z^c} & -\frac{y^c(y^c - y_0^c)}{z^c} - (z^c - z_0^c) & \frac{y^c(x^c - x_0^c)}{z^c} & x^c - x_0^c \\ -1 & 0 & \frac{x^c}{z^c} & \frac{x^c(y^c - y_0^c)}{z^c} & -\frac{x^c(x^c - x_0^c)}{z^c} - (z^c - z_0^c) & y^c - y_0^c \end{bmatrix} \tag{C.9}$$

Now that a relationship between linear and angular velocity and OF is established, it is time to argue for the centre of rotation o_T^c. All rotations of objects seen in the image will be rotations about the UAV, hence the rotation point o_T^c coincides with the origin of the body frame. As the camera is placed close to the centre of Gravity of the UAV, the following assumption is a good approximation

$$o_T^c = [0, 0, 0]^T \tag{C.10}$$

Inserting (C.10) into (C.8) one finally arrive at an expression for the linear and angular velocity of the terrain with respect to the camera in the camera-fixed frame

$$\begin{aligned} \begin{bmatrix} \dot{r} \\ \dot{s} \end{bmatrix} &= \frac{f}{z^c} \begin{bmatrix} 0 & 1 & -\frac{y^c}{z^c} & -\frac{y^c(y^c)}{z^c} - z^c & \frac{y^c x^c}{z^c} & x^c \\ -1 & 0 & \frac{x^c}{z^c} & \frac{x^c y^c}{z^c} & -\frac{x^c(x^c)}{z^c} - z^c & y^c \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{T/c}^c \\ \omega_{T/c}^c \end{bmatrix} \\ &= M(f, p^c) \begin{bmatrix} \boldsymbol{v}_{T/c}^c \\ \omega_{T/c}^c \end{bmatrix} \end{aligned} \tag{C.11}$$

This in turn might then be extended to concern N points $(r_1, s_1) \ldots (r_N, s_N)$

$$\begin{bmatrix} \dot{r_1} \\ \dot{s_1} \\ \vdots \\ \dot{r_N} \\ \dot{s_N} \end{bmatrix} = \begin{bmatrix} M(f, p_1^C) \\ \vdots \\ M(f, p_N^C) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{T/c}^c \\ \omega_{T/c}^c \end{bmatrix} \tag{C.12}$$

By calculating the pseudoinverse of $\boldsymbol{M}$ in (C.12) the angular and linear velocity can be computed as

$$\begin{bmatrix} \boldsymbol{v}_{T/c}^c \\ \omega_{T/c}^c \end{bmatrix} = \begin{bmatrix} M(f, p_1^c) \\ \vdots \\ M(f, p_N^c) \end{bmatrix}^+ \begin{bmatrix} \dot{r_1} \\ \dot{s_1} \\ \vdots \\ \dot{r_N} \\ \dot{s_N} \end{bmatrix} \tag{C.13}$$

$\boldsymbol{M}^+$ being the pseudoinverse of $\boldsymbol{M}$. The pseudoinverse can be calculated as $(M^T M)^{-1} M^T$ if $M^T M$ has full rank for all states. This can only happen if the number of flow vectors are greater or equal to three. In addition the three vectors cannot be located on the same line. This is because the system of equations will be linearly dependent in this case and the system undetermined. A minimum demand is therefore that at least three OF vectors must be present when calculating velocities, and the vectors cannot be located on the same line.

Velocity in body coordinates

Now an expression for the linear and angular velocity of the terrain with respect to the camera given in the camera-fixed coordinate frame is present($\boldsymbol{v}^c_{T/c}$ and $\omega^c_{T/c}$). However the interesting velocity is the linear and angular velocity of the camera with respect to the terrain given in the camera-fixed coordinate frame, namely $\boldsymbol{v}^c_{c/T}$ and $\omega^c_{c/T}$.

$$\begin{bmatrix} \boldsymbol{v}^c_{c/T} \\ \omega^c_{c/T} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{v}^c_{T/c} \\ \omega^c_{T/c} \end{bmatrix} \tag{C.14}$$

As the camera is placed very near the center of gravity of the UAV, it is assumed that the camera frame and the body frame coincides. Hence $\{c\} = \{b\}$. Furthermore assume that the terrain being filmed coincides with NED (terrain not moving) $\{T\} = \{n\}$. This results in the following formula for body-fixed velocity in 6 DOF based on OF:

$$\begin{bmatrix} \boldsymbol{v}^b_{b/n} \\ \omega^b_{b/n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}^c_{c/T} \\ \omega^c_{c/T} \end{bmatrix} = - \begin{bmatrix} M(f, p^c_1) \\ \vdots \\ M(f, p^c_N) \end{bmatrix}^+ \begin{bmatrix} \dot{r_1} \\ \dot{s_1} \\ \vdots \\ \dot{r_N} \\ \dot{s_N} \end{bmatrix} \tag{C.15}$$

For clarity the $\boldsymbol{M}$-matrix is now defined:

Definition 12. *The matrix $M(f, p^c)$ relates optical flow to angular and linear velocity*

$$M(f, p^c) = \frac{f}{z^c} \begin{bmatrix} 0 & 1 & -\frac{y^c}{z^c} & -\frac{y^c(y^c)}{z^c} - z^c & \frac{y^c x^c}{z^c} & x^c \\ -1 & 0 & \frac{x^c}{z^c} & \frac{x^c y^c}{z^c} & -\frac{x^c(x^c)}{z^c} - z^c & y^c \end{bmatrix} \tag{C.16}$$

Appendix D

Software Implementation

As great focus in this thesis has been to provide real world experimental results, a lot of implementation have been conducted. This appendix seeks address some important implementation considerations.

The OF algorithms used in thesis has been implemented in C++ with the open source computer vision framework, openCV (Bradski, 2000). The Multiplicative Extended Kalman Filter and the Nonlinear Observers are implemented in Matlab. A simulator is implemented in Matlab. The simulator provide realistic IMU sensor data, GNSS and OF vectors from a simulated flight. Implemented software is illustrated in Figure D.1.

This appendix is organized as follows.

Optical flow algorithms in Appendix D.1. Implementation specifics of the OF algorithms used in this thesis are presented.

MEKF in Appendix D.2. General implementation considerations for the MEKF is highlighted.

NO in Appendix D.3. General implementation considerations for the NO is presented.

Camera simulator in Appendix D.4. Explanation of how a simulated OF sensor is implemented.

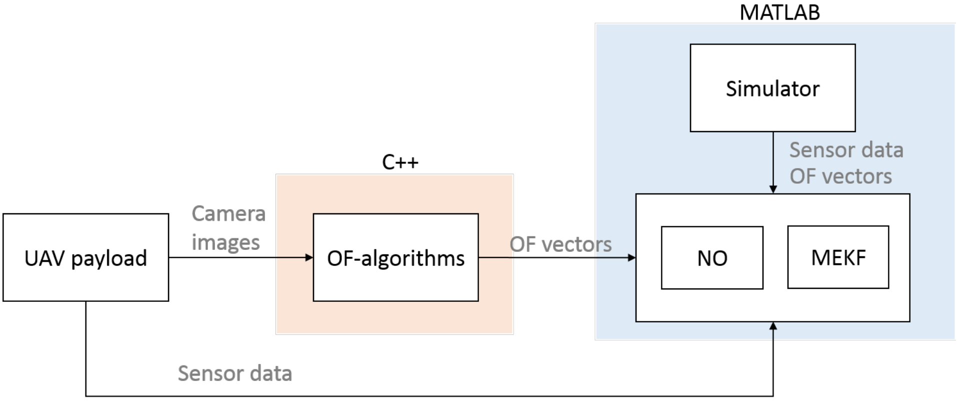


Figure D.1: Implemented software is indicated in the red and blue boxes.

D.1 Optical Flow

Images from a UAV test flight is stored on an onboard payload module, and must be post-processed to get measurement of the optical flow (OF). Different OF algorithms are implemented in C++. The openCV library (Bradski, 2000) has been used as framework. The algorithms SIFT (Lowe, 1999), SURF (Bay, Tuytelaars and Van Gool, 2008), Lucas-Kanade (Lucas and Kanade, 1981) , Farneback (Farnebäck, 2003) and Template Matching (Sonka et al., 2014) are all implemented and employed on images from the UAV flight described in Section 6.1. SIFT and SURF are both feature-based algorithms. The SIFT algorithm is found to perform better than SURF in the application of finding OF vectors from a UAV flying over snowy environments. Lucas-Kanade showed to be very inaccurate in the presence of large displacements in the image (due to the large velocity of the UAV), and was therefore not appropriate for this application. The Farneback algorithm produced a dense vector field, resulting in unwieldy amount of data. The SIFT and Template-Matching algorithms were found to be a good combination for generating reliable OF vectors that could be used by the observers. These algorithms will therefore be described more in detail.

This section will give an explanation about the implementation, and some tuning specifics done to achieve good OF results. Sometimes even worlds best tuned OF algorithm may output erroneous OF vector, hence a simple outlier detector is implemented and presented in this section.

SIFT

The SIFT algorithm (Lowe, 1999) is implemented with OpenCV (Bradski, 2000), where it exist a class for the SIFT algorithm. A pseudo code of the implementation is seen in listing D.1. This SIFT class is used to find features and calculate the descriptors. Then a Flann (Fast Approximate Nearest Neighbour Search Library) based feature matcher (Muja and Lowe, 2009) is used to find common features in consecutive images. Each match from the Flann based matcher provides a value that expresses the least squares distance between the descriptors. The smaller this distance is, the better the match is. After doing the matching process, each match is assessed at the end with the best match in mind. This is done in order to ensure that every match is good. A match that has a distance that is greater than the double of the best match, is said to be a bad match. If however, the best match has distance less than 0.01, then all matches are compared to a threshold of 0.02. This makes sure that only the features corresponding to the best matches are used to calculate the discrete OF. The discrete OF is then calculated as the displacement of the features corresponding to the best matches.

Listing D.1: Implementation of the SIFT OF algorithm.

```
%\caption{Implementation of the SIFT OF algorithm.}
%\label{listing:siftimpl}
// Locates features in the image and assigns descriptors
sift(image[k], keypoints[k], descriptors[k]);

// Match descriptors with Flann Matcher and store results in matches
cv::FlannBasedMatcher matcher;
std::vector<cv::DMatch> matches;
if(descriptors[k].rows != 0 && descriptors[k-1].rows != 0){
    // Match descriptors from the new image and the previous image
    matcher.match((cv::Mat)descriptors[k], ...
         (cv::Mat)descriptors[k-1], matches);

    // Quick calculation of max and min distances between keypoints
    double max_dist = 0;
    double min_dist = 100;
    for( int i = 0; i < matches.size(); i++ ){
        double dist = matches[i].distance;
        if( dist < min_dist )
            min_dist = dist;
        if( dist > max_dist )
            max_dist = dist;
    }

    // Collect only "good matches"
    std::vector<cv::DMatch> good_matches;
    for( int i = 0; i < matches.size(); i++ ){
        if( matches[i].distance ≤ std::max(2*min_dist, 0.02) ) {
            good_matches.push_back( matches[i]);
        }
    }

    // Calculate optical flow
    for (int i = 0; i<good_matches.size(); i ++){
        cv::DMatch match = good_matches[i];
        // Create OF-vector from corresponding keypoints.
        OF of_vector[i] = (keypoints[k][match.queryIdx].pt - ...
            keypoints[k-1][match.trainIdx].pt);
    }
}
```

Template Matching

Fixed point template matching is implemented with functions from OpenCV. Two consecutive images from the camera on the UAV are used to calculate the discrete OF. In this thesis the images are divided in twelve regions of equal size placed symmetrical around in the whole image. A template is extracted from the centre of each region, implying that the template should be smaller than the size of the regions. The templates are then matched with the next image to find the displacement. The flow is only calculated if the match for each template is better than 99%. This value might seem high, but there is still a chance for mismatches, even with this value.

Feature-based Template Matching

A feature-based template matching algorithm is implemented. A maximum of 40 features are extracted from the image using the openCV-function `goodFeaturesToTrack` (Shi and Tomasi, 1994). A template is then extracted with dimensions 120 × 90 pixels. The templates are matched with the second image in order to get measurements of the OF. The chance of getting regions with much contrast, and hence being easy to match, increases when creating using features as interest points. However there is no lower bound for the number of OF vectors extracted by the algorithm, and this method is not used in the experimental results.

Outlier Detector

A simple outlier detector is implemented in Matlab. An incorrect optical flow vector may arise if the match of one feature from one image to the next fails. This could be that the optical flow algorithm outputs a erroneous vector by a mistake, which may happen if it is hard to distinguish between different features in the image. If flying over uniform terrain like sand, forest and ocean, it might be hard to get correct matches between features from consecutive images, hence hard to calculate correct optical flow vectors. In order to suppress the errors that may arise from wrong optical flow vectors, a two-dimensional vector outlier detector

Table D.1: Outlier Detection Algorithm

#	**Description**
1.	Generate a two-dimensional histogram of a set of vectors $V = \{v_i\}, i \in \{1, ..., N\}, v_i = [x, y]^T$. The vectors are sorted on their x- and y-value in the histogram. The height of a bin in the histogram corresponds to the number of vectors mapped inside the bin. When dealing with optical flow vectors the x-value is corresponding to $\dot{r}$, while y is corresponding to $\dot{s}$. The histogram is divided in 10×10 equally spaced bins, with a minimum bin width and height of 20 pixels in x and y direction, meaning that there is a room of at least 20 pixels margin in each direction (recall that if rolling, pitching or yawing there will be differences in the vectors across the image).
2.	Find the bin with the largest number of vectors.
2.1	If one bin is larger than all others, set the acceptance region to this bin plus half the width of the bin in both directions. Thus the vectors closest to the best bin might be accepted. The algorithm is finished.
2.2	If there are multiple bins with the largest number of vectors inside, check if all of them are neighbours. These bins are called the maxima. If they are neighbours the algorithm accepts all bins. If not, jump to 3).
3.	Check the neighbours of the maxima. Does one of the maxima has a more populated neighbourhood than the other ones? If yes, this particular bin plus half of the bin width in both directions is the acceptance region. If not, let the bins with the most populated neighbourhoods be in the maxima. Now jump to 4)
4.	Check whether the remaining bins in the maxima are neighbours. If they are, this is the acceptance region. If there are multiple neighbourhoods with the same population, one are unable to determine which vectors are outliers, and which are correct. The region of acceptance is set as empty, meaning there are no inliers.

is designed. The outlier detector is based on a two-dimensional histogram. By placing the optical flow vectors in a two-dimensional histogram based on their $\dot{r}$- and $\dot{s}$-value it is possible to determine the areas with the highest density of vectors, namely a neighbourhood of the most common vectors. It is assumed that the neighbourhood with the most vectors are the inliers, while the vectors outside of this neighbourhood are indeed outliers. The algorithm is described in detail in Table D.1.

D.2 Multiplicative Extended Kalman Filter

The multiplicative extended Kalman filter (MEKF) is implemented in Matlab. The MEKF is implemented to run off-line with real world sensor data collected from a UAV experiment. This section contains implementation details about the MEKF.

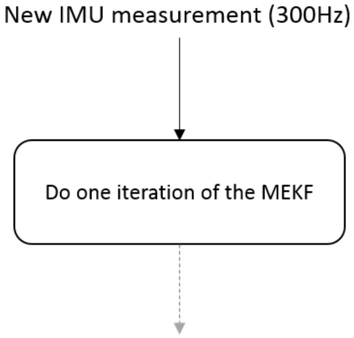


Figure D.2: The MEKF is triggered by the IMU and hence run on the same frequency as the IMU, that is 300 Hz.

How different measurement rates are handled and the quaternion ambiguity are addressed. The MEKF algorithm is presented in Chapter 4.

Handle Different Measurement Rates

One of the main challenges is how to handle that different sensor data arrive with different frequency. In this thesis this is handled in a simple manner. The MEKF is configured to run on same the frequency as the highest measurement rate. The sensor with the highest measurement rate is the IMU. The IMU outputs data with a frequency of 300Hz. From the IMU we get acceleration, angular velocity and inclinometer measurements. One iteration of the MEKF is triggered by the presence of a new IMU measurement as shown in Figure D.2.

The different measurement rates are handled in the following way: First calulate the Kalman gain matrix K, assuming that all measurements are available. If one of the sensors have not provided a new measurement set the column corresponding to this sensor to zero. Note that this is essentially the same as saying that the covariance of an absent measurements is infinity (Gelb et al., 1974). Setting the covariance to infinity would lead to numerical issues, hence K is set to zero in this thesis.

In practice this is handled by the code snippet in listing D.2. Note that the three first elements in K , corresponding to the IMU, are always present. 'valid_cv' indicates whether the measurement from computer vision is new. 'valid_gps' indicates the validity of the GPS position and velocity measurements. A mask matrix is constructed based on the validity of the sensor data, and a masked Kalman gain matrix is obtained by matrix multiplication.

Listing D.2: Measurements arriving at different rates are handled by masking the Kalman gain matrix K

```
K_        = P_bar*H'/(H*P_bar*H'+R);
mask      = blkdiag(eye(3), valid_cv*eye(3), valid_gps*eye(6));
K         = K_*mask;
```

Quaternion Ambiguity

As the attitude is represented by the unit quaternion in the MEKF, the quaternion ambiguity must be considered. The quaternion ambiguity is that a quaternion q represents the same rotation as $-q$. In order to handle this ambiguity, a check is done after constructing the unit quaternion from $\delta\hat{\boldsymbol{\epsilon}}$ as shown in listing D.3.

Listing D.3: Handling the quaternion ambiguity

```
dq_bar        = qbuild(deps_hat); % construct quaternion from vector
q_hat_new     = qmult(q_hat_prev, dq_bar); % Update quaternion estimate
q_hat_new     = q_hat_new/norm(q_hat_new);
if(norm(q_hat_new+q_hat_prev)<norm(q_hat_new-q_hat_prev))
    q_hat = -q_hat_new;
else
    q_hat = q_hat_new;
end
return q_hat
```

By the code snippet above it is ensured that the new estimated quaternion is the quaternion closest to the previous estimate.

D.3 Nonlinear Observer

The nonlinear observer is implemented in Matlab. This section highlights how different measurement rates are handled as well as some general implementation considerations.

First we show how the estimated states are updated in the nonlinear observer. This is done by first order Euler integration, and takes on the following form:

$$\hat{R}(k+1) = \hat{R}(k) + \delta t \dot{\hat{R}} \tag{D.1}$$

$$\hat{\boldsymbol{b}}^b_{\text{gyro}}(k+1) = \hat{\boldsymbol{b}}^b_{\text{gyro}}(k) + \delta t \dot{\hat{\boldsymbol{b}}}^b_{\text{gyro}} \tag{D.2}$$

$$\hat{\boldsymbol{p}}^n_{b/n}(k+1) = \hat{\boldsymbol{p}}^n_{b/n}(k) + \delta t \dot{\hat{\boldsymbol{p}}}^n_{b/n} \tag{D.3}$$

$$\hat{\boldsymbol{v}}^n_{b/n}(k+1) = \hat{\boldsymbol{v}}^n_{b/n}(k) + \delta t \dot{\hat{\boldsymbol{v}}}^n_{b/n} \tag{D.4}$$

$$\xi(k+1) = \xi(k) + \delta t \dot{\xi} \tag{D.5}$$

δt being the time between two iterations of the observer. The observer is run on 300Hz, meaning that $\delta t = (1/300)$sec.

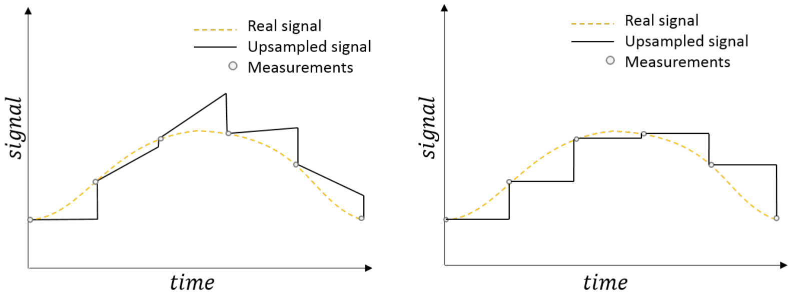
Handle Different Measurement Rates

As the injection term of the NO, $\hat{J}$ from equation (5.4) depends on the body-fixed velocity and acceleration measurements as reference vectors, the filter is required to have body-fixed measurements at every iteration. As it is desirable to take advantage of all available information, the observer is implemented to run on the same rate as the most frequent sensor. The IMU outputs data with 300Hz, the computer vision 10Hz, and the GPS 5Hz. The observer is implemented to run with 300 iterations every second. Two look up tables are generated to keep track of when a measurement is available before the observer is run. One for the GPS and one for the OF measurements. The objective of the look up table is to notify the observer when a new measurement is available.

As the CV only supplies ten new measurements of body-fixed velocity every second, this means that reference vectors must be generated when not having new data. This could be done in multiple ways, and the most obvious would be: 1) Linear extrapolation as illustrated in Figure D.3 (a). Linear extrapolation can be achieved by comparing previous and current measurement, create a slope, and extrapolate until a new measurement arrives.

```
dt = (time - time_of_previous);
slope = (vel_b - vel_b_previous) / dt;
if( not new measurement)
    vel_b = vel_b + slope*dt;
else
    vel_b = new measurement;
end
```

or 2) zero-order hold which is illustrated in Figure D.3 (b). That is to keep the reference vector the same as the last measurement until a new measurement arrives.



(a)

(b)

Figure D.3: (a) Linear extrapolation (b) Zero-order-hold

In this thesis this upsampling is done by zero-order hold. That is to keep the reference vector the same as the last measurement until a new measurement arrives. This is illustrated in Figure D.4.

The GPS measurements arrives at 5Hz. When new GPS measurements are available and the complete translational observer is run. When no GPS measurement is available, then the observer corrector terms $K_{pp}(\boldsymbol{p}^n_{GPS} - \hat{\boldsymbol{p}}^n_{b/n})$, $K_{pv}(\boldsymbol{v}^n_{GPS} - \hat{\boldsymbol{v}}^n_{b/n})$, $K_{vp}(\boldsymbol{p}^n_{GPS} - \hat{\boldsymbol{p}}^n_{b/n})$, $K_{vv}(\boldsymbol{v}^n_{GPS} - \hat{\boldsymbol{v}}^n_{b/n})$, $K_{\xi p}(\boldsymbol{p}^n_{GPS} - \hat{\boldsymbol{p}}^n_{b/n})$ and $K_{\xi v}(\boldsymbol{v}^n_{GPS} - \hat{\boldsymbol{v}}^n_{b/n})$ are omitted in the state update. The GPS look up table is illustrated in Figure D.5.

D.4 Camera Simulator

A camera/OF sensor is simulated in order to get measurements of the OF. The camera simulator is a part of the simulator presented in Chapter 9. Here the objective is to describe how a camera and OF algorithm can be simulated without having access to real images. The objective is to find the displacement of a projected point in the image plane between time t_{k-1} to t_k, that is $\Delta r = r(t_k) - r(t_{k-1})$ and $\Delta s = s(t_k) - s(t_{k-1})$. Lets first consider how one can choose features to project given the UAVs attitude, position and a elevation profile of the terrain. These features are the one that we wish to find the OF of.

At a time t_k a ray is drawn in the camera z-axis as shown in Figure D.6. The ray intersects the ground plane at a point $\boldsymbol{t}^n_{\text{centre}} = [x^n_{\text{centre}}, y^n_{\text{centre}}, 0]^T$ or expressed in {c} $\underline{\boldsymbol{t}}^c_{\text{centre}} = (T^n_c)^{-1}\underline{\boldsymbol{t}}^n_{\text{centre}}$, T^n_c being the homogeneous transformation matrix relating {c} and {n} . The point $\boldsymbol{t}^c_{\text{centre}}$ is named the "centre ground point".

Points are chosen deterministically around the centre ground point, $\boldsymbol{t}^c_{\text{centre}}$, distributed on a plane perpendicular to the ray from {c} to the centre ground point. This plane is named the field of view (FOV) plane. The points are distributed

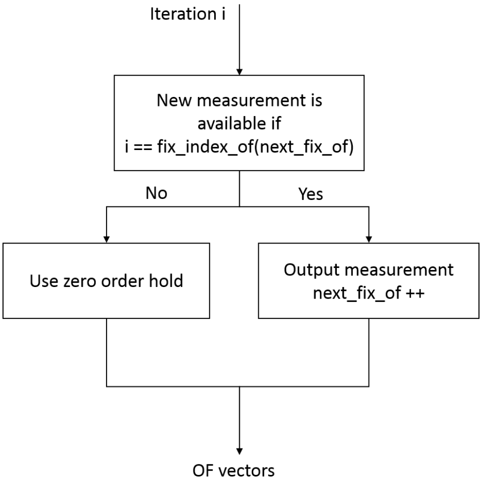

Iteration i
New measurement is available if i == fix_index_of(next_fix_of)
No
Yes
Use zero order hold
Output measurement next_fix_of ++
OF vectors

Figure D.4: The observer ask the look up table, fix_index_of, whether a new OF measurement is available. If a new measurement is available then a new v_{cv}^b is calculated based on the new OF vectors. If not, then zero order hold is used on v_{cv}^b.

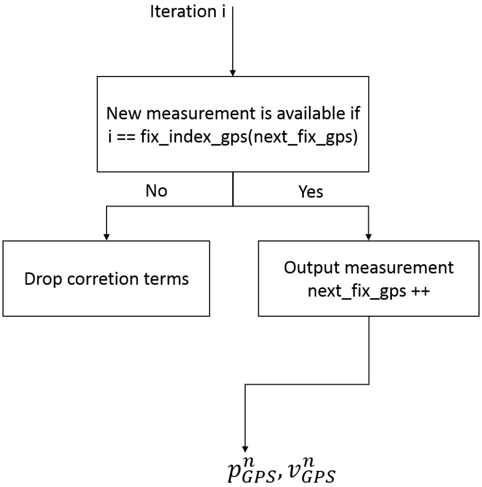

Iteration i
New measurement is available if i == fix_index_gps(next_fix_gps)
No
Yes
Drop corretion terms
Output measurement next_fix_gps ++
p_{GPS}^n, v_{GPS}^n

Figure D.5: The observer ask the look up table, fix_index_gps, whether a new GPS measurement is available. If a new measurement is available then $\boldsymbol{p}_{GPS}^n$ and $\boldsymbol{v}_{GPS}^b$ is output. If not, then the observer corrector terms are omitted in the state update.

on the FOV plane, ranging from the centre ground point -30 to 30 meters in camera x-direction -40 to 40 meters in camera y-direction, separated with 10 meters in both dimensions. Lets call these points ”FOV features” and denote them by $\boldsymbol{p}^c_{\text{FOV}}$. The FOV features in camera coordinates is then defined as $\boldsymbol{p}^c_{\text{FOV}} \in \boldsymbol{t}^c_{\text{centre}} + [x, y, 0]^T, x \in [-30, -20, \ldots, 20, 30], y \in [-40, -30, \ldots, 30, 40]$. Lets now consider only one of the FOV features, and denote this FOV feature $\boldsymbol{p}^c_{\text{FOV}}$.

The FOV feature $\boldsymbol{p}^c_{\text{FOV}}$ is transformed to {n} by $\underline{\boldsymbol{p}}^n_{\text{FOV}} = T^n_c \underline{\boldsymbol{p}}^c_{\text{FOV}}$. Let the FOV feature be defined as $\boldsymbol{p}^n_{\text{FOV}} = [x^n_{\text{FOV}}, y^n_{\text{FOV}}, z^n_{\text{FOV}}]$. The FOV feature is then projected onto the terrain by using $x^n_{\text{FOV}}, y^n_{\text{FOV}}$ and the elevation h at the $x^n_{\text{FOV}}, y^n_{\text{FOV}}$ coordinate of the elevation profile. The projected point is then $\boldsymbol{p}^n = [x^n_{\text{FOV}}, y^n_{\text{FOV}}, h]^T$, which is called a ”feature”.

Now that the feature location in {n} is found, it is in our interest to find the projection of this feature at time t_k and t_{k-1}. The camera moves between t_{k-1} and t_k, meaning the homogeneous transformation matrix T^n_c is time variant. The feature can then be transformed to {c} by $\underline{\boldsymbol{p}}^c(t_k) = (T^n_c)^{-1}(t_k)\underline{p}^n$ and $\underline{\boldsymbol{p}}^c(t_{k-1}) = (T^n_c)^{-1}(t_{k-1})\underline{p}^n$. The points $p^c(t_k), p^c(t_{k-1})$ represents the feature on the surface of the terrain given in camera coordinates at time t_k and t_{k-1} respectively.

The feature at time t_{k-1} and t_k can then be projected onto the image plane by the pinhole camera model from (3.3), yielding $\boldsymbol{x}^m(t_k) = [r(t_k), s(t_k)]^T$ and $\boldsymbol{x}^m(t_{k-1}) = [r(t_{k-1}), s(t_{k-1})]^T$. The discrete OF can then be found as $\Delta r = r(t_k) - r(t_{k-1})$ and $\Delta s = s(t_k) - s(t_{k-1})$.

Fig. D.6 illustrates the relationship between the centre ground point $\boldsymbol{t}_{\text{centre}}$, FOV features $\boldsymbol{p}_{\text{FOV}}$, and features $\boldsymbol{p}$.

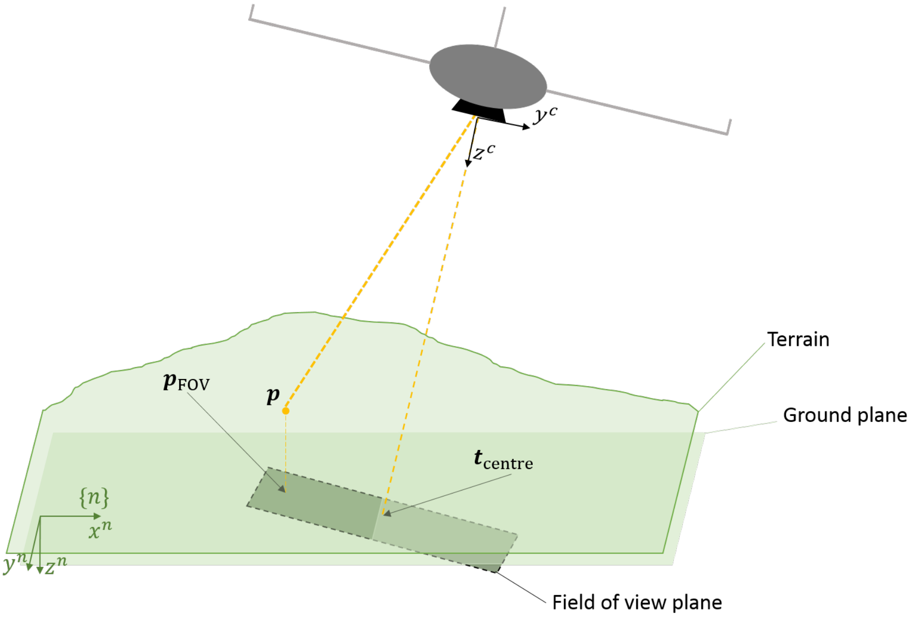


Figure D.6: Features on the surface of the terrain are chosen based on the attitude and position of the UAV. A ray along the camera z-axis intersects the ground plane at $\boldsymbol{t}_{\text{centre}}$. A plane denoted field of view (FOV) is constructed perpendicular to the ray. FOV features are distributed along the FOV plane. Features are constructed with z^n-component from the elevation profile and x^n, y^n coordinate from the corresponding FOV feature. Features $\boldsymbol{p}$ are projected onto the image plane by the pinhole camera model to find the image plane coordinate $x^m = [r, s]^T$. This is done at time t_k and t_{k+1} with the same features, $\boldsymbol{p}$, to get the discrete OF dr and ds.

Appendix E

Payload Setup

A Penguin B UAV produced by the UAV factory is equipped with a custom payload made by the navigation team at NTNU, in particular Sigurd M. Albrektsen, Jakob M. Hansen, Kasper T. Borup and Lorenzo Fusini.

The UAV payload has different sensors. Accelerometers, gyroscopes, inclinometer, GPS, altimeter and a video camera are present in the payload. The payload is shown in Figure E.1. A complete list of the sensors used in this thesis is given in Table E.1. In addition to these components a second IMU, an ADIS 16488, produced by Analog Devices is mounted in the payload. The data from the ADIS is not used in this thesis. A laser altimeter is mounted in the payload, but the measurements from the altimeter was not extracted. A custom synchronization board is designed. The synchronization board timestamps all measurements, such that all measurements are aligned on the same time axis. The camera is attached to the payload, and the lens is directed straight down in the body frame as displayed in Figure Figure E.2.

Table E.1: Components used in the flight experiment

Component	**Manufacturer**	**Device name**	**Output**	**Rate**
UAV	UAV Factory	Penguin B		
Camera	IDS	UI-5250CP-C-HQ	Images	10 fps
GPS	uBlox	LEA-6T	Position	5 Hz
IMU	Sensonor AS	STIM300	Acc., gyro., inclinometer	300 Hz

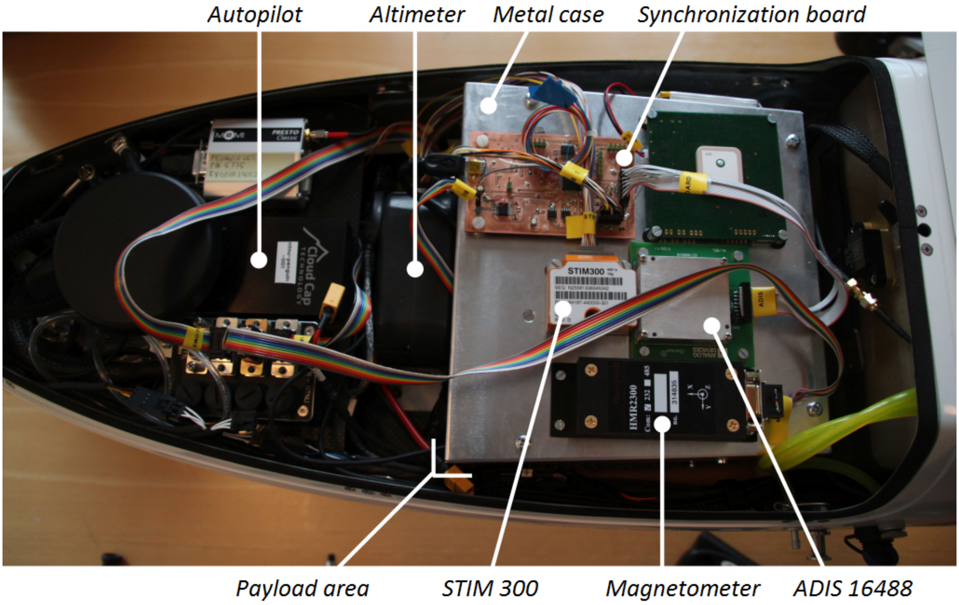


Figure E.1: Payload setup

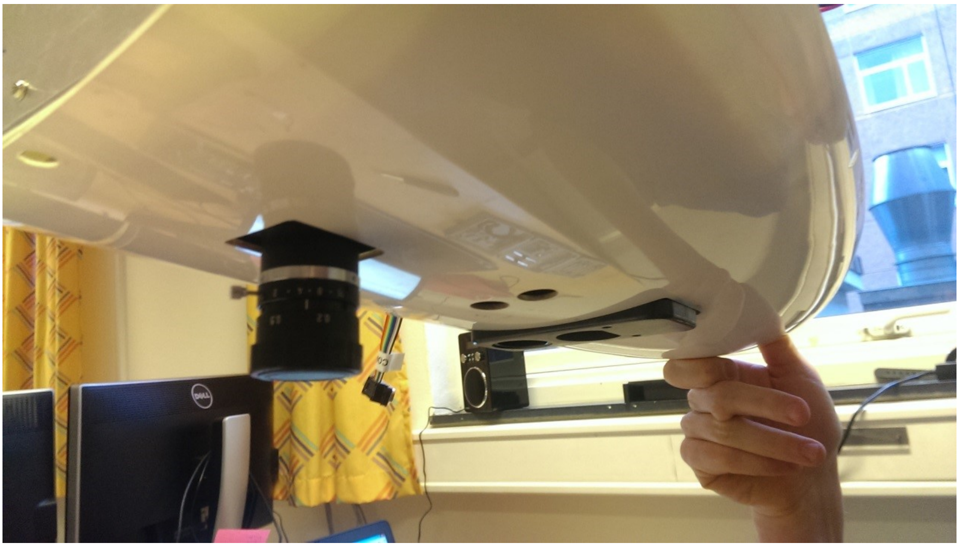


Figure E.2: Payload module is placed such that camera is pointing downwards

In order to calculate OF, the choice of camera and lens can affect the performance of the algorithms significantly. In the payload a commercial camera manufactured by IDS is used. The lens is produced by Tamron. The camera and lens specification are given in Table E.2. Some basic features of the camera like field of view varies with the focal length of the lens. In this experiment a lens with constant focal length (no optical zoom) is preferred. Thus the field of view in both horizontal and vertical direction is constant. The camera is configured to have global shutter.

Table E.2: Camera and lens specification

Camera	**UI-5250CP-C-HQ**
Sensor technology	CMOS Color
Resolution(h × v)	1600 × 1200
Sensor Size(h × v)	$7.2mm \times 5.4mm$
Pixel size	$4.5\mu m$
Maximal FPS	35.6
Shutter	Rolling shutter/**Global shutter**/Global start shutter
Color depth	12 bit
Data transmission	Gigabit ethernet
Input Voltage	12-24 VDC
Lens	**Tamron**
Focal length	$8mm$
Horizontal field of view α_h	48.46°
Vertical field of view α_v	37.3°

Appendix F

Reference frames

Some common reference frames will now be described.

F.1 ECI

The Earth Centered Inertial frame has its origin in the center of the earth, with Z-axis oriented along the rotational axis of the earth. The X-axis is pointing towards the vernal equinox. The Y-axis completes the right hand system. This frame can be assumed inertial for terrestrial navigation and hence be denoted by {i} .

F.2 ECEF

The Earth-Centered Earth-Fixed coordinate system is the cartesian coordinate system rotating with the earth, and is denoted by {e} . The Z-axis is pointing along the earth rotation axis. The X-axis is pointing towards the intersection of 0° longitude (Greenwich meridian) and 0° latitude (Equator). The Y-axis is oriented to complete the right hand system. ECEF rotates relative to ECI with the earth rotation rate ω_e, and is not an inertial frame. The ECEF coordinate system can be represented by cartesian coordinates as well as ellipsoidal coordinates. The GPS satellite positions are given in cartesian coordinates, while the output to the user is normally given in ellipsoidal coordinates, i.e. latitude, longitude and height. The GPS is currently using the WGS84-reference ellipsoid, and the latitude, longitude and height are given in means of this reference ellipsoid.

F.3 NED

The North East Down frame is defined relative to the Earths reference ellipsoid, and is denoted {n} . The Z-axis is pointing downwards, perpendicular to the tangent plane of the ellipsoid. The X-axis is pointing towards the true north, while the Y-axis is directed East. The NED frame, {n} , is not inertial, but when dealing with local navigation one can assume it is inertial. In this thesis {n} is assumed inertial.

F.4 Body Frame

The Body frame is a coordinate frame fixed to the body, and is denoted {b} . X-axis is pointing forwards of the body, Z-axis is pointing downwards, while Y-axis is pointing to the right in order to fulfill the right hand system. The body frame {b} is related to the NED frame {n} by the Euler angles roll, pitch and yaw, or by the attitude quaternion q, or by the direction cosine matrix (DCM) R_b^n.

F.5 Image plane

The image plane is a the plane located on the imaging chip of a camera. A point in the image plane has coordinates $[r, s]$ where the origin is in the center of the image, with r-axis oriented towards the right, and s-axis is oriented downwards. The image plane is denoted by {m} .

F.6 Camera Frame

The camera frame is fixed to the camera and is denoted by {c} . The Z-axis is oriented along the focal line, hence out of the image plane, X-axis is oriented upwards in the image plane (oposite direction of s-axis), while the Y-axis is oriented to the right of the image plane (same direction as r-axis).

Appendix G

Attitude representation

In this section different attitude representations are listed. In specific the different attitude representations presented here is used to describe the orientation of {b} relative to {n} . In this thesis you will find that the quaternion, the direction cosine matrix (DCM) matrix and the Euler angles are used. Different attitude representations have different advantages and drawbacks.

G.1 Euler angles

Euler angles are perhaps the simplest and most intuitive representation of the attitude. From three angles, $\mathbf{\Theta}_{b/n} = [\phi, \theta, \psi]^T$ (roll, pitch and yaw), it is possible to determine the orientation of {b} relative to {n} . The main advantages of this representation are that it has a minimum number of parameters to describe the orientation (only three parameters), and it is intuitive. The main drawback is the singularity arising at $\theta = 90^\circ$, often referred to as a gimbal lock.

The Euler angles are related to the rotation matrix by the following

$$R_b^n = R_{z,\psi} R_{y,\theta} R_{x,\phi} \tag{G.1}$$

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \tag{G.2}$$

$$R_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \tag{G.3}$$

$$R_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{G.4}$$

The relation between derivative of the Euler angles and the angular velocity are defined by

$$\dot{\boldsymbol{\Theta}}_{b/n} = T_\Theta(\boldsymbol{\Theta}_{b/n})\boldsymbol{\omega}^b_{b/n} \tag{G.5}$$

$$T_\Theta^{-1}(\boldsymbol{\Theta}_{b/n}) = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \Rightarrow T_\Theta(\boldsymbol{\Theta}_{b/n}) \quad = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{G.6}$$

G.2 Direction Cosine Matrix

The Direction Cosine Matrix (DCM) is a singularity free representation of the attitude. An example of a DCM is R_b^n which transforms vectors from {b} to {n} . A DCM is a orthogonal 3×3 matrix in SO(3). The DCM has nine parameters to define a orientation, and is the DCM relating {b} to {n} is defined as

$$R_b^n = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{G.7}$$

As $R \in SO(3)$ it satisfies

$$RR^T = R^T R = I_{3\times3}, \quad det(R) = 1 \tag{G.8}$$

The dynamics of the DCM is given by

$$\dot{R}_b^n = R_b^n \left[\boldsymbol{\omega}^b_{b/n}\right]_\times \tag{G.9}$$

G.3 Unit Quaternions

The unit quaternion is a four parameter representation of the attitude, and is a singularity free alternative to the Euler angles and the DCM (Egeland and Gravdahl, 2002; Fossen, 2011). A quaternion is defined as a complex number with a real part η and with a vector of three imaginary components $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \epsilon_3]^T$. The quaternion is defined as

$$q = \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \tag{G.10}$$

If a rotation is defined by a angle β and an unit vector $\boldsymbol{k}$ (angle-axis-representation), then the corresponding quaternion is defined as (Chou, 1992)

$$\eta = cos\left(\frac{\beta}{2}\right) \tag{G.11}$$

$$\boldsymbol{\epsilon} = \boldsymbol{k} sin\left(\frac{\beta}{2}\right) \tag{G.12}$$

Moreover a unit quaternion satisfies

$$q^T q = 1 \tag{G.13}$$

The inverse of a unit quaternion is

$$q^{-1} = \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix} \tag{G.14}$$

The time derivative of an inverse quaternion:

$$\dot{q}^{-1} = -q^{-1}\dot{q}q^{-1} \tag{G.15}$$

The quaternion product denoted $\otimes$ is defined as

$$q_1 \otimes q_2 = \begin{bmatrix} \eta_1\eta_2 - \boldsymbol{\epsilon}_1^T\boldsymbol{\epsilon}_2 \\ \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + [\boldsymbol{\epsilon}_1]_\times \boldsymbol{\epsilon}_2 \end{bmatrix} \tag{G.16}$$

Quaternion product can also be applied as a matrix multiplication

$$q_1 \otimes q_2 = F(q_1)q_2 = E(q_2)q_1 \tag{G.17}$$

$$F(q) = \begin{bmatrix} \eta & -\boldsymbol{\epsilon}^T \\ \boldsymbol{\epsilon} & \eta I_{3\times3} + [\boldsymbol{\epsilon}]_\times \end{bmatrix} \tag{G.18}$$

$$E(q) = \begin{bmatrix} \eta & -\boldsymbol{\epsilon}^T \\ \boldsymbol{\epsilon} & \eta I_{3\times3} - [\boldsymbol{\epsilon}]_\times \end{bmatrix} \tag{G.19}$$

$$\tag{G.20}$$

The inverse of a quaternion product is defined as

$$(q_1 \otimes q_2)^{-1} = q_2^{-1} \otimes q_1^{-1} \tag{G.21}$$

The quaternion product is applicable for vectors in $\mathcal{R}^3$ by treating the vectors as quaternions with first element, η, equal to zero. The corresponding quaternion, p, to a vector $\boldsymbol{p} \in \mathcal{R}^3$ is

$$p = \begin{bmatrix} 0 \\ \boldsymbol{p} \end{bmatrix} \in \mathcal{R}^4 \tag{G.22}$$

The quaternion product $\otimes$ of a vector and a quaternion is hence

$$\boldsymbol{p} \otimes q = \begin{bmatrix} -\boldsymbol{p}^T \boldsymbol{\epsilon} \\ \eta \boldsymbol{p} + [\boldsymbol{p}]_\times \boldsymbol{\epsilon} \end{bmatrix} \tag{G.23}$$

The identity quaternion $q_{id}$is defined as

$$q_{id} = q^{-1} \otimes q = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{G.24}$$

and holds the property

$$q_{id} \otimes p = p \otimes q_{id} = p \tag{G.25}$$

The relationship between a quaternion and a rotation matrix is defined as below

$$R_b^n(q) = I_{3\times3} + 2\eta[\boldsymbol{\epsilon}]_\times + 2[\boldsymbol{\epsilon}]_\times^2 \tag{G.26}$$

$$q \otimes p \otimes q^{-1} = R_b^n(q)\boldsymbol{p}, \quad p = \begin{bmatrix} 0 & \boldsymbol{p}^T \end{bmatrix}^T, \quad \boldsymbol{p} \in \mathcal{R}^3 \tag{G.27}$$

The dynamics of a quaternion is related to the angular velocity by the following relationship:

$$\dot{q} = \frac{1}{2} q \otimes \boldsymbol{\omega}_{b/n}^b = \frac{1}{2} T(q) \boldsymbol{\omega}_{b/n}^b \tag{G.28}$$

$$T(q) = \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\epsilon}^T \\ \eta I_{3\times3} + [\boldsymbol{\epsilon}]_\times \end{bmatrix} \tag{G.29}$$

Appendix H

Sensor Models

This section presents the sensors and the measurement equations used in the design of the observers. In the following {n} is assumed to be inertial.

H.1 Accelerometer

The accelerometer is a sensor that senses the acceleration of the rigid body of which the accelerometer is attached to. The accelerometer measures the specific force $\boldsymbol{f}^b_{\text{imu}}$. Assuming small misalignment and scaling error, hence neglecting them, the accelerometer measurement might be modelled as (Fossen, 2011):

$$\begin{aligned} \boldsymbol{f}^b_{\text{imu}} &= a^b_{b/n} - R^b_n \boldsymbol{g}^n + \boldsymbol{b}^b_{\text{acc}} + \boldsymbol{w}_{\text{acc}} && \text{(H.1)} \\ &= R^b_n \left(\dot{\boldsymbol{v}}^n_{b/n} - \boldsymbol{g}^n \right) + \boldsymbol{b}^b_{\text{acc}} + \boldsymbol{w}_{\text{acc}} && \text{(H.2)} \\ & && \text{(H.3)} \end{aligned}$$

Different accelerometer error models have been proposed by Vik (2014); Fossen (2011). In this thesis the accelerometer error is decomposed into a bias and a white noise component. A simple bias model is used, modelling the bias as a slowly varying process, namely the Wiener Process:

$$\dot{\boldsymbol{b}}^b_{\text{acc}} = \boldsymbol{w}_{\text{bacc}} \tag{H.4}$$

$\boldsymbol{w}_{\text{bacc}} \in \mathcal{R}^3$ being Gaussian white noise.

H.2 Gyroscope

The gyroscope measures the angular velocity of {b} relative to the inertial frame.

$$\boldsymbol{\omega}^b_{\text{imu}} = \boldsymbol{\omega}^b_{b/n} + \boldsymbol{\omega}^b_{n/i} + \boldsymbol{b}^b_{\text{gyro}} + \boldsymbol{w}_{\text{gyro}} \tag{H.5}$$

By assuming that {n} is inertial, it is possible to neglect the earth rotation, hence the measurement model might be rewritten as:

$$\boldsymbol{\omega}^b_{\text{imu}} = \boldsymbol{\omega}^b_{b/n} + \boldsymbol{b}^b_{\text{gyro}} + \boldsymbol{w}_{\text{gyro}} \tag{H.6}$$

As for the accelerometer, the gyroscope measurement is assumed to be contamined by white noise and a bias component. The gyroscopic bias is modelled as a Wiener process:

$$\dot{\boldsymbol{b}}^b_{\text{gyro}} = \boldsymbol{w}^b_{\text{bgyro}} \tag{H.7}$$

H.3 Computer Vision

The theory behind using a camera as a sensor is covered in Chapter 3. The output from the computer vision subsystem can be stated as:

$$\boldsymbol{v}^b_{cv} = \boldsymbol{v}^b_{b/n} + \boldsymbol{w}_{cv} = R^b_n \boldsymbol{v}^n_{b/n} + \boldsymbol{w}_{cv} \tag{H.8}$$

Or when dealing with normalized measurements

$$\boldsymbol{v}^b_{cv} = \frac{\boldsymbol{v}^b_{b/n}}{\|\boldsymbol{v}^b_{b/n}\|} + \boldsymbol{w}_{cv} = R^b_n \frac{\boldsymbol{v}^n_{b/n}}{\|\boldsymbol{v}^n_{b/n}\|} + \boldsymbol{w}_{cv} \tag{H.9}$$

H.4 Global Positioning System (GPS)

The GPS measures the position and the velocity of {b} given in {e} coordinates. When modelling the GPS measurements in the Kalman filter, the measurement are assumed to be contaminated by white noise. By assuming local navigation the GPS measurement can be transformed to {n} . The following measurement equations can be stated:

$$\boldsymbol{p}^n_{\text{GPS}} = \boldsymbol{p}^n_{b/n} + \boldsymbol{w}_{pos} \tag{H.10}$$

$$\boldsymbol{v}^n_{\text{GPS}} = \boldsymbol{v}^n_{b/n} + \boldsymbol{w}_{vel} \tag{H.11}$$

Note that a more realistic noise model for the GPS is the Gauss-Markov noise model. In the simulator in Section 7.1 a Gauss-Markov noise model is implemented to generate GPS measurement noise.

Bibliography

Ahrens, S., Levine, D., Andrews, G. and How, J. P. (2009), 'Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments', *Proceedings - IEEE International Conference on Robotics and Automation* pp. 2643–2648.

Batista, P., Silvestre, C. and Oliveira, P. (2011*a*), '{GES} Attitude Observers - {Part II}: Single Vector Observations', *Proceedings of the 18th IFAC World Congress* **18**, 2991–2996.

Batista, P., Silvestre, C. and Oliveira, P. (2011*b*), '{GES} Attitude Observers - {Part I}: Multiple General Vector Observations', *Proceedings of the 18th IFAC World Congress* **18**, 2985–2990.

Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. (2008), 'Speeded-Up Robust Features (SURF)', *Computer Vision and Image Understanding* **110**(3), 346–359.

Bay, H., Tuytelaars, T. and Van Gool, L. (2008), SURF: Speeded-Up Robust Features, *in* '9th European Conference on Computer Vision', Vol. 110, pp. 346–359.

Bazin, J. C., Demonceaux, C., Vasseur, P. and Kweon, I. S. (2010), 'Motion estimation by decoupling rotation and translation in catadioptric vision', *Computer Vision and Image Understanding* **114**(2), 254–273.

Bibuli, M., Caccia, M. and Lapierre, L. (2007), 'Path-following algorithms and experiments for an autonomous surface vehicle', *IFAC Proceedings Volumes (IFAC-PapersOnline)* **7**(6), 81–86.

B.K.P.Horn and B.G.Schunk (1981), 'Determining optical flow', *Artif. Intell.* **17**, 185–204.

Bradski, G. (2000), 'The OpenCV Library', *Dr Dobbs Journal of Software Tools* **25**, 120–125.

Brockers, R., Susca, S., Zhu, D. and Matthies, L. (2012), 'Fully self-contained vision-aided navigation and landing of a micro air vehicle independent from external sensor inputs', *Proceedings of SPIE - The International Society for Optical Engineering* **8387**.

Chao, H., Gu, Y. and Napolitano, M. (2014), 'A survey of optical flow techniques for robotics navigation applications', *Journal of Intelligent and Robotic Systems: Theory and Applications* **73**(1-4), 361–372.

Chou, J. C. K. (1992), 'Quaternion kinematic and dynamic differential equations', *IEEE Transactions on Robotics and Automation* **8**(1), 53–64.

Conroy, J., Gremillion, G., Ranganathan, B. and Humbert, J. S. (2009), 'Implementation of wide-field integration of optic flow for autonomous quadrotor navigation', *Autonomous Robots* **27**, 189–198.

Crassidis, J. L., Markley, F. L. and Cheng, Y. (2007), 'Survey of Nonlinear Attitude Estimation Methods', *Journal of Guidance, Control, and Dynamics* **30**(1), 12–28.

Davison, A. J., Reid, I. D., Molton, N. D. and Stasse, O. (2007), 'MonoSLAM: Real-time single camera SLAM', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6), 1052–1067.

Diel, D. D., DeBitetto, P. and Teller, S. (2005), 'Epipolar Constraints for Vision-Aided Inertial Navigation', *IEEE Workshop on Application of Computer Vision* pp. 221–228.

Dusha, D. (2007), 'Attitude estimation for a fixed-wing aircraft using horizon detection and optical flow', *Proceedings - Digital Image Computing Techniques and Applications: 9th Biennial Conference of the Australian Pattern Recognition Society, DICTA 2007* pp. 485–492.

Egeland, O. and Gravdahl, J. T. (2002), *Modeling and simulation for automatic control*, Marine Cybernetics Trondheim, Norway, Trondheim.

Euston, M., Coote, P., Mahony, R., Kim, J. and Hamel, T. (2008), 'A complementary filter for attitude estimation of a fixed-wing UAV', *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* pp. 340–345.

Farnebäck, G. (2003), 'Two-frame motion estimation based on polynomial expansion', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2749**, 363–370.

Fossen, T. I. (2011), *Handbook of Marine Craft Hydrodynamics and Motion Control*.

Fusini, L., Fossen, T. I. and Johansen, T. A. (2014), A Uniformly Semiglobally Exponentially Stable Nonlinear Observer for GNSS-and Camera-Aided Inertial Navigation, *in* 'Proceedings of the 22nd IEEE Mediterranean Conference on Control and Automation (MED'14)'.

Fusini, L., Hosen, J., Helgesen, H. H., Johansen, T. A. and Fossen, T. I. (2015), 'Experimental Validation of a Uniformly Semi-globally Exponentially Stable Nonlinear Observer for GNSS- and Camera-aided Inertial Navigation for Fixed-wing UAVs', *The 2015 International Conference on Unmanned Aircraft Systems* .

Gelb, A., Kasper, J. F. J., Nash, R. A. J., Price, C. F. and Sutherland, A. A. J. (1974), *Applied optimal estimation.*

Grabe, V., Bulthoff, H. H. and Giordano, P. R. (2012), 'On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow', *2012 IEEE International Conference on Robotics and Automation* pp. 491–497.

Grabe, V., Bulthoff, H. H. and Robuffo Giordano, P. (2012), 'Robust optical-flow based self-motion estimation for a quadrotor UAV', *IEEE International Conference on Intelligent Robots and Systems* pp. 2153–2159.

Grewal, M. S., Weill, L. R. and Andrews, A. P. (2001), *Global positioning systems, inertial navigation, and integration.*, Wiley.

Grip, H. F., Fossen, T. I., Johansen, T. A. and Saberi, A. (2012*a*), 'A nonlinear observer for integration of {GNSS and IMU} measurements with gyro bias estimation', *American Control Conference (ACC)* pp. 4607–4612.

Grip, H. F., Fossen, T. I., Johansen, T. A. and Saberi, A. (2012*b*), 'Attitude Estimation Using Biased Gyro and Vector Measurements With Time-Varying Reference Vectors', *IEEE Transactions on Automatic Control* **57**(5), 1332–1338.

Grip, H. F., Fossen, T. I., Johansen, T. A. and Saberi, A. (2013), 'Nonlinear Observer for {GNSS}-Aided Inertial Navigation with Quaternion-Based Attitude Estimation', *American Control Conference (ACC)* pp. 272–279.

Grip, H. F., Fossen, T. I., Johansen, T. A. and Saberi, A. (2015), 'Globally exponentially stable attitude and gyro bias estimation with application to {GNSS/INS} integration', *Automatica* **51**, 158–166.

Grotli, E. I., Chaillet, A. and Gravdahl, J. T. (2008), 'Output control of spacecraft in leader follower formation', *Proceedings of the 47th IEEE Conference in Decision and Control* pp. 1030–1035.

Groves, P. D. (2013), *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, second edn, Artech House, Norwood, MA, USA.

Guerrero-Castellanos, J., Madrigal-Sastre, H., Durand, S., Torres, L. and Muñoz Hernández, G. (2013), 'A Robust Nonlinear Observer for Real-Time Attitude Estimation Using Low-Cost {MEMS} Inertial Sensors', *Sensors* **13**(11), 15138–15158.

Hartley, R. (1997), 'In defense of the eight-point algorithm', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(6), 580–593.

Hartley, R. and Zisserman, A. (2003), *Multiple View Geometry in Computer Vision*, The Press Syndicate of the University of Cambridge.

Herisse, B., Russotto, F.-X., Hamel, T. and Mahony, R. (2008), 'Hovering flight and vertical landing control of a VTOL Unmanned Aerial Vehicle using optical flow', *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* pp. 801–806.

Hrabar, S., Sukhatme, G. S., Corke, P., Usher, K. and Roberts, J. (2005), 'Combined optic-flow and stereo-based navigation of urban canyons for a UAV', *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* pp. 302–309.

Hua, M. D. (2010), 'Attitude estimation for accelerated vehicles using {GPS/INS} measurements', *Control Engineering Practice* **18**(7), 723–732.

Hua, M. D., Ducard, G., Hamel, T., Mahony, R. and Rudin, K. (2014), 'Implementation of a Nonlinear Attitude Estimator for Aerial Robotic Vehicles', *IEEE Transactions on Control System Technology* **22**(1), 201–213.

Huang, T. S. and Netravali, A. N. (1994), 'Motion and structure from feature correspondences: a review', *Proceedings of the IEEE* **82**(2), 252–268.

Hutchinson, S., Hager, G. D. and Corke, P. I. (1996), 'A tutorial on visual servo control', *IEEE Transactions on Robotics and Automation* **12**(5), 651–670.

Kehoe, J. J., Watkins, A. S., Causey, R. S. and Lind, R. (2006), 'State estimation using optical flow from parallax-weighted feature tracking', *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference* **8**, 5030–5045.

Kreyszig, E. (2006), *Advanced Engineering Mathematics*, Vol. 53.

Krstic, M., Kanellakopoulos, I. and Kokotovic, P. V. (1995), *Nonlinear and Adaptive Control Design*, Wiley, New York.

Labrie, M. and Hebert, P. (2007), 'Efficient camera motion and 3D recovery using an inertial sensor', *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)* .

Lefferts, E., Markley, F. and Shuster, M. (1982), 'Kalman filtering for spacecraft attitude estimation', *Journal of Guidance, Control, and ...* **5**, 417–429.

Longuet-Higgins, H. C. (1981), 'A computer algorithm for reconstructing a scene from two projections'.

Longuet-Higgins, H. C. (1984), 'The reconstruction of a scene from two projections - configurations that defeat the 8-point algorithm'.

Lowe, D. (1999), 'Object recognition from local scale-invariant features', *Proceedings of the Seventh IEEE International Conference on Computer Vision* **2**.

Lucas, B. D. and Kanade, T. (1981), An Iterative Image Registration Technique with an Application to Stereo Vision, *in* 'Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 81)', pp. 674–679.

Ma, Y., Soatto, S., Košecká, J. and Sastry, S. (2001), 'An Invitation to 3D Vision', *Springer* .

Mahony, R., Euston, M., Kim, J., Coote1, P. and Hamel, T. (2011), 'A non-linear observer for attitude estimation of a fixed-wing unmanned aerial vehicle without {GPS} measurements', *Transactions of the Institute of Measurement and Control* **33**(6), 699–717.

Mahony, R., Hamel, T. and Pflimlin, J. M. (2008), 'Nonlinear complementary filters on the special orthogonal group', *IEEE Transactions on Automatic Control* **53**(Xx), 1203–1218.

Mahony, R., Hamel, T., Trumpf, J. and Lageman, C. (2009), 'Nonlinear attitude observers on {SO(3)} for complementary and compatible measurements: A theoretical study', *Proceedings of the 48th IEEE Conference on Decision and Control* pp. 6407–6412.

Mammarella, M., Campa, G., Fravolini, M. L. and Napolitano, M. R. (2012*a*), 'Comparing Optical Flow Algorithms Using 6-DOF Motion of Real-World Rigid Objects', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(6), 1752–1762.

Mammarella, M., Campa, G., Fravolini, M. L. and Napolitano, M. R. (2012*b*), 'Comparing Optical Flow Algorithms Using 6-DOF Motion of Real-World Rigid Objects', *IEEE Transactions on systems, Man, and Cybernetics* **42**(6), 1752–1762.

Markley, F. (2004), 'Attitude estimation or quaternion estimation?', *Journal of Astronautical Sciences* pp. 1–15.

Maybank, S. J. and Faugeras, O. D. (1992), 'A theory of self-calibration of a moving camera'.

Maybeck, P. S. (1979), 'Stochastic models, estimation, and control', *New York* **1**, 1–16.

Meingast, M., Geyer, C. and Sastry, S. (2004), 'Vision based terrain recovery for landing unmanned aerial vehicles', *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)* **2**.

Mercado, D. A., Flores, G., Castillo, P., Escareno, J. and Lozano, R. (2013), 'GP-S/INS/optic flow data fusion for position and Velocity estimation', *International Conference on Unmanned Aircraft Systems, ICUAS - Conference Proceedings* pp. 486–491.

Merrell, P. C., Lee, D.-J. and Beard, R. W. (2004*a*), 'Obstacle avoidance for unmanned air vehicles using optical flow probability distributions', *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **5609**(1), 13–22.

Merrell, P. C., Lee, D.-J. and Beard, R. W. (2004*b*), 'Statistical analysis of multiple optical flow values for estimation of unmanned aerial vehicle height above ground', *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **5608**, 298–305.

Moore, R. J. D., Thurrowgood, S. and Srinivasan, M. V. (2012), 'Vision-only estimation of wind field strength and direction from an aerial platform', *IEEE International Conference on Intelligent Robots and Systems* pp. 4544–4549.

Muja, M. and Lowe, D. G. (2009), Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, *in* 'International Conference on Computer Vision Theory and Applications (VISAPP '09)', pp. 1–10.

Nister, D. (2003), 'An efficient solution to the five point relative pose problem', pp. 195–202.

Okatani, T. and Deguchi, K. (2002), 'Robust estimation of camera translation between two images using a camera with a 3D orientation sensor', *Object recognition supported by user interaction for service robots* **1**, 275–278.

Phillips, R. and Schmidt, G. (1996), 'System Implications and Innovative Applications of Satellite Navigation'.

Ruffier, F. and Franceschini, N. (2004), 'Visually guided micro-aerial vehicle: Automatic take off, terrain following, landing and wind reaction', *Proceedings - IEEE International Conference on Robotics and Automation* **2004**(3), 2339–2346.

Salcudean, S. (1991), 'A globally convergent angular velocity observer for rigid body motion', *IEEE Transactions on Automatic Control* **36**(12), 1493–1497.

Sanfourche, M., Delaune, J. and Besnerais, G. (2012), 'Perception for UAV: Vision-based navigation and environment modeling', *Aerospace Lab Journal* pp. 1–19.

Schmidt, S. F. (1981), 'The Kalman filter - Its recognition and development for aerospace applications', *Journal of Guidance, Control, and Dynamics* **4**(1), 4–7.

Shakernia, O., Sharp, C. S., Vidal, R., Shim, D. H., Ma, Y. and Sastry, S. (2002), Multiple View Motion Estimation and Control for Landing an Unmanned Aerial Vehicle, *in* 'IEEE Internation Conference on Robotics and Automation'.

Shen, S., Michael, N. and Kumar, V. (2011), 'Autonomous multi-floor indoor navigation with a computationally constrained MAV', *Proceedings - IEEE International Conference on Robotics and Automation* pp. 20–25.

Shi, J. S. J. and Tomasi, C. (1994), 'Good features to track', *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on* .

Sonka, M., Hlavac, V. and Boyle, R. (2014), *Image processing, analysis, and machine vision*, Cengage Learning.

Thienel, J. and Sanner, R. M. (2003), 'A coupled nonlinear spacecraft attitude controller and observer with an unknown constant gyro bias and gyro noise', *IEEE Transactions on Automatic Control* **48**(11), 2011–2015.

Vik, B. (2014), *Integrated Satellite and Inertial Navigation Systems.*

Vik, B. and Fossen, T. I. (2001), 'A nonlinear observer for{ GPS and INS} integration', *Proceedings of the 40th IEEE Conference on Decision and Control* **3**, 2956–2961.

Weiss, S. (2012), 'Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments', *Proceedings - IEEE International Conference on Robotics and Automation* pp. 957–964.

Weiss, S., Brockers, R. and Matthies, L. (2013), '4DoF drift free navigation using inertial cues and optical flow', *IEEE International Conference on Intelligent Robots and Systems* pp. 4180–4186.

Wen, T., Deng, H. and Liu, J. (2011), Recovering ego-motion from optical flow for aerial navigation, *in* 'Proceedings of SPIE - The International Society for Optical Engineering', Vol. 8003.

Zhang, Z. (1998), 'Determining the Epipolar Geometry and its Uncertainty: A Review', *International Journal of Computer Vision* **27**, 161–195.

Zhuang, X. H., Huang, T. S. and Haralick, R. M. (1986), 'Two-view motion analysis: a unified algorithm.', *Journal of the Optical Society of America. A, Optics and image science* **3**(9), 1492–1500.

Zingg, S., Scaramuzza, D., Weiss, S. and Siegwart, R. (2010), MAV navigation through indoor corridors using optical flow, *in* 'Proceedings - IEEE International Conference on Robotics and Automation', pp. 3361–3368.

Zufferey, J. C. and Floreano, D. (2005), 'Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control', *Proceedings - IEEE International Conference on Robotics and Automation* **2005**, 2594–2599.