Institutionen för systemteknik
Department of Electrical Engineering

**Examensarbete**

# Vision and GPS based autonomous landing of an unmanned aerial vehicle

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan i Linköping
av

**Joel Hermansson**

LiTH-ISY-EX--10/4313--SE

Linköping 2010


**Linköpings universitet**
**TEKNISKA HÖGSKOLAN**

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

# Vision and GPS based autonomous landing of an unmanned aerial vehicle

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan i Linköping
av

**Joel Hermansson**

LiTH-ISY-EX--10/4313--SE

Handledare: **Martin Skoglund**
ISY, Linköpings universitet
**Andreas Gising**
CybAero AB

Examinator: **Thomas Schön**
ISY, Linköpings universitet

Linköping, 23 April, 2010

**Avdelning, Institution**
Division, Department

Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

**Datum**
Date

2010-04-23

**Språk**
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ _____

**Rapporttyp**
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ _____

**ISBN**
—

**ISRN**
LiTH-ISY-EX--10/4313--SE

**Serietitel och serienummer** **ISSN**
Title of series, numbering —

**URL för elektronisk version**
http://www.control.isy.liu.se
www.ep.liu.se

**Titel**
Title
Autonom landning av en modellhelikopter med hjälp av bildbehandling och GPS
Vision and GPS based autonomous landing of an unmanned aerial vehicle

**Författare**
Author
Joel Hermansson

**Sammanfattning**
Abstract

A control system for autonomous landing of an unmanned aerial vehicle (UAV) with high precision has been developed. The UAV is a medium sized model helicopter. Measurements from a GPS, a camera and a compass are fused with an extended Kalman filter for state estimation of the helicopter. Four PID-controllers, one for each control signal of the helicopter, are used for the helicopter control. During the final test flights fifteen landings were performed with an average landing accuracy of 35 cm.

A bias in the GPS measurements makes it impossible to land the helicopter with high precision using only the GPS. Therefore, a vision system using a camera and a pattern provided landing platform has been developed. The vision system gives accurate measurement of the 6-DOF pose of the helicopter relative the platform. These measurements are used to guide the helicopter to the landing target. In order to use the vision system in real time, fast image processing algorithms have been developed. The vision system can easily match up the with the camera frame rate of 30 Hz.

**Nyckelord**
Keywords
Vertical Take-Off and Landing, Unmanned Aerial Vehicle, helicopter, autonomous landing, vision, image processing, Extended Kalman filter, control

# Abstract

A control system for autonomous landing of an unmanned aerial vehicle (UAV) with high precision has been developed. The UAV is a medium sized model helicopter. Measurements from a GPS, a camera and a compass are fused with an extended Kalman filter for state estimation of the helicopter. Four PID-controllers, one for each control signal of the helicopter, are used for the helicopter control. During the final test flights fifteen landings were performed with an average landing accuracy of 35 cm.

A bias in the GPS measurements makes it impossible to land the helicopter with high precision using only the GPS. Therefore, a vision system using a camera and a pattern provided landing platform has been developed. The vision system gives accurate measurement of the 6-DOF pose of the helicopter relative the platform. These measurements are used to guide the helicopter to the landing target. In order to use the vision system in real time, fast image processing algorithms have been developed. The vision system can easily match up the with the camera frame rate of 30 Hz.

# Sammanfattning

Ett kontrolsystem för att autonomt landa en modellhelikopter har utvecklats. Mätdata från en GPS, en kamera samt en kompass fusioneras med ett Extended Kalman Filter för tillståndsestimering av helikoptern. Fyra PID-regulatorer, en för varje kontrolsignal på helikoptern, har används för regleringen. Under den sista provflygningen gjordes tre landingar av vilken den minst lyckade slutade 35 cm från målet.

På grund av en drift i GPS-mätningarna är det omöjligt att landa helikoptern med hög precision med bara en GPS. Därför har ett bildbehandlingssystem som använder en kamera samt ett mönster på platformen utvecklats. Bidbehandlingssystemet mäter positionen och orienteringen av helikoptern relativt platformen. Dessa mätningar används kompensera för GPS-mätningarnas drift. Snabba bildbehandlingsalgoritmer har utvecklats för att kunna använda bildbehandlingssystemet i realtid. Systemet är mycket snabbare än 30 bilder per sekund vilket är kamerans hastighet.

# Acknowledgments

Special thanks to my supervisor at CybAero, Andreas Gising, for many great ideas and discussions. Thanks to our two test pilots, Claes Mejier and Robert Veenhuizen, that has been of great help. Without them putting up with really cold and nasty weather the project would have been impossible. I would also like to thank Johan Mårtensson and Magnus Sethsson at CybAero for offering such an interesting thesis.

From the University I would like to thank my supervisor Martin Skoglund and my examiner Thomas Schön for all the help and great discussions.

# Contents

# Nomenclature

## Mathematical notation

| | |
|---|---|
| $X$ | Helicopter position in west → east direction ($x$-position in the ground coordinate system). |
| $Y$ | Helicopter position in south → north direction ($y$-position in the ground coordinate system). |
| $Z$ | Helicopter altitude above the ground ($z$-position in the ground coordinate system). |
| $\theta$ | Helicopter heading. |
| $\varphi$ | Helicopter roll. |
| $\gamma$ | Helicopter pitch. |
| $P_X$ | Platform position in west → east direction ($x$-position in the ground coordinate system). |
| $P_Y$ | Platform position in south → north direction ($y$-position in the ground coordinate system). |
| $P_Z$ | Platform altitude above the ground ($z$-position in the ground coordinate system). |
| $P_\theta$ | Platform orientation. |
| Index $g$ | Ground coordinate system. |
| Index $p$ | Platform coordinate system. |
| Index $h$ | Helicopter coordinate system. |
| Index $c$ | Camera coordinate system. |
| $R_{\mathrm{x2y}}$ | Rotation matrix from coordinate system $x$ to coordinate system $y$. |
| $T^z_{\mathrm{x2y}}$ | Translation vector from coordinate system $x$ to coordinate system $y$ given in coordinate system $z$. |
| $I$ | The identity matrix. |
| $\bar{0}_{n\times m}$ | Matrix of only zeros of size $n \times m$ |
| $u_{\mathrm{elev}}$ | The elevation control signal (controls the helicopter tilt forward/backward). |
| $u_{\mathrm{roll}}$ | The roll control signal (controls the helicopter tilt right/left). |
| $u_{\mathrm{yaw}}$ | The yaw control signal (controls the heading of the helicopter). |
| $u_{\mathrm{pitch}}$ | The collective pitch control signal (controls the lift of the helicopter). |

## Abbreviations

| | |
|---|---|
| pose | position and orientation. |
| blob | binary large object. |

# Chapter 1

# Introduction

## 1.1 Background

CybAero develops and manufactures unmanned helicopters. Because of an increasing demand in maritime applications they are working on a project to autonomously launch and land these helicopters on ships. The project is called Mobile Automatic Launch and Landing Station (MALLS). The project includes guidance of the helicopter to the platform, precision landing on the platform, gyro stabilization of the platform to compensate for wave movements and a locking device to secure the helicopter onto the platform when it has landed. This thesis is a part of the MALLS project, where a quarter scale helicopter is to be landed on a landing platform with high precision.

## 1.2 Problem

The problem is to autonomously land a UAV (Unmanned Aerial Vehicle) helicopter on a platform. A GPS (Global Positioning System) receiver is mounted on the helicopter and an approximate location of the platform is given. This is used for guiding the helicopter to the platform. When the platform is reached, a camera together with a pattern on the platform is used to improve the quality of the helicopter pose estimate and the platform location. An accurate pose is needed in order to perform a more precise landing.

To complete the task there are two already existing support systems. A pilot support system that is stabilizing the helicopter and a vision system which uses images of the platform to estimate the pose of the helicopter [1].

## 1.3 Purpose and goal

The goal of this thesis is to land a model helicopter autonomously on a stationary platform. In order to reach this goal the following milestones has been identified.

- **Milestone 1:** Autonomous hovering at a GPS coordinate.
- **Milestone 2:** Autonomous hovering above the platform.
- **Milestone 3:** Autonomous landing on the platform.

This thesis is one step in the development towards launching and landing UAV-helicopters on moving vehicles.

## 1.4 Method

In order to solve the problem real flight tests were performed on a quarter scale helicopter. A simple model of the helicopter was identified from flight data and a set of control parameters was obtained from simulations with this model. An extended Kalman filter was used to fuse data from several sensors for estimating the helicopter state.

During the flight tests a professional pilot assisted at all times. A switch on the helicopter allowed the on-board control computer to master some of the control signals, see Section 2.2, while the pilot controlled the rest. In the beginning of the project the on-board computer only had control of one of the control signals at a time. During the development more and more of the control signals were given to the on-board computer.

## 1.5 Thesis outline

Chapter 2 gives an overview of the system. This chapter also describes the helicopter and its control signals. The coordinate systems used in this thesis are described in Chapter 2 as well. In Chapter 3 the helicopter model, the sensor data and the helicopter state estimation is described. The controller given in Chapter 4 uses the state estimates to calculate the control signals. In Chapter 5 the results of the thesis are described and Chapter 6 concludes the project. Suggestions for future work are also given in Chapter 6.

## 1.6 Related work

In a recent overview of aerial robotics [2], the problem of autonomously landing a UAV is pinpointed as an active research area, where more work is needed. Despite this, the problem is by no means new, several interesting approaches have been presented, see e.g., [3, 4, 5, 6, 7]. In [3] the authors makes use of a GPS and a vision system to autonomously land a UAV on a pattern similar to the one used in this work. They report 14 test flights and the average landing accuracy is 40 cm. Compared to this, the 15 landings performed in this work resulted in a average landing accuracy of 34 cm. Furthermore, the vision system in [3] only provides estimates of the horizontal position and the heading angle, whereas the sensor fusion framework in this report also provide estimates of the altitude and the roll and pitch angles.

A vision system similar to the one in this thesis was developed in [8]. This vision system was also made for landing an UAV. They tested their system in a real flight with an autonomous helicopter. In order to test the performance of their vision system, the pose estimation of the helicopters navigation system was compared with the pose estimation of the vision system. Their conclusion from this was a maximum pose estimation error of 5 cm in each axis of translation and 5 degrees in each axis of rotation.

Another system for autonomous helicopter landing using image processing was reported in [9]. They are using another type of pattern that provides the position, heading angle and the altitude of the helicopter.

# Chapter 2

# System overview

## 2.1 Physical system

The system mainly consists of a helicopter and a ground control station. The ground control station receives data from the helicopter through a wireless network in order to continuously provide the operator with information about the helicopter state. The operator can also send commands over the network to the helicopter.

The helicopter can be operated in two modes; autonomous mode and manual mode. In manual mode the pilot has command over all the control signals and in autonomous mode the computer controls an optional number of the control signals and the pilot the rest of them. This is done by a switch, see Figure 2.1, which is controlled by the pilot. This means that the pilot can take control of the helicopter at any time.

In both manual and autonomous mode the helicopter is controlled through a pilot support system called Helicommand, see Figure 2.1. The Helicommand uses accelerometers and gyros to stabilize the helicopter. The measurements from the Helicommand are not available.

Three different sensors were used, a GPS, a compass and a camera which provide the control computer with measurements. The measurements are used by the computer to estimate the state of the system in order to make control decisions.

- Helicopter: Align TREX 600E, see Section 2.2.
- GPS: Thales DG16.
- Compass: Honeywell HMR3000, three axis compass that uses the magnetic field and a bubble level for orientation measurements.
- Camera: Point Gray Firefly MV USB 2.0.
- PC: fitPC2 1.6 GHz Linux computer with the control system + solid state hard drive.

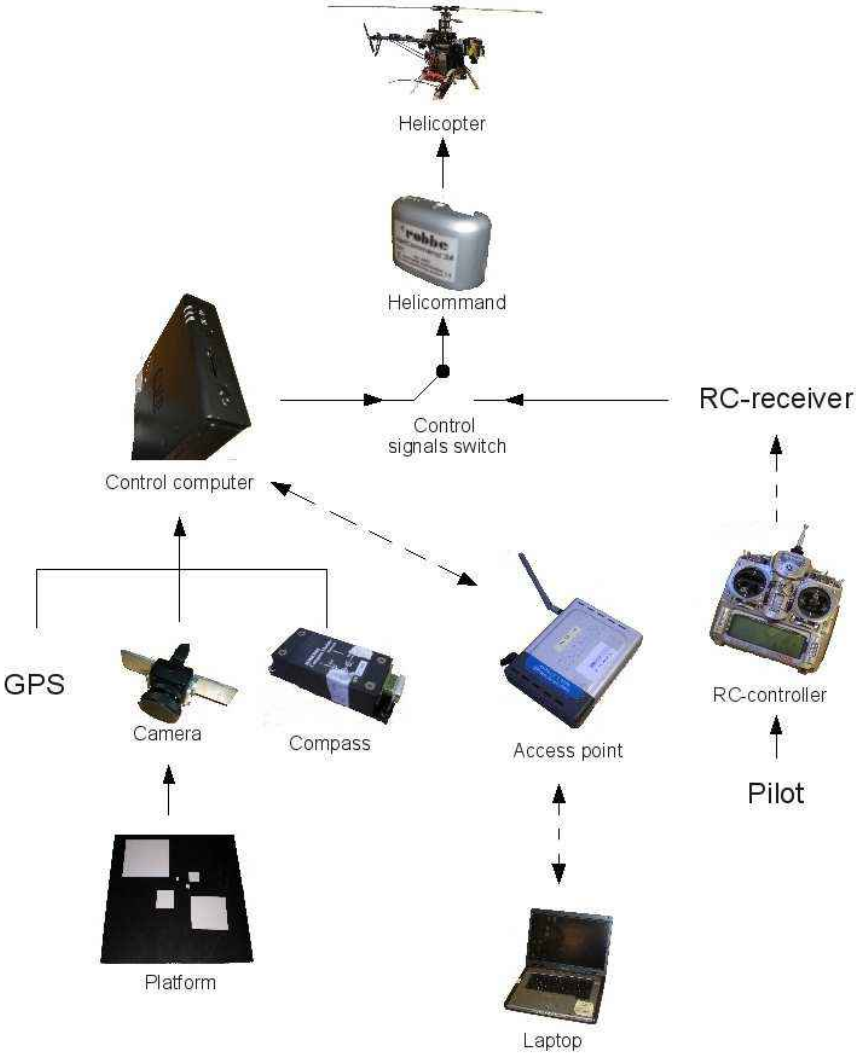



**Figure 2.1.** An overview of the system. The communication direction is marked with arrows.

- Switch: Used to switch between manual and autonomous mode. In autonomous mode the switch passes the signals from the PC to the Helicommand system. In manual mode the signals from the RC-receiver is passed on to the Helicommand system.

- Robbe Helicommand: A pilot support system that uses gyros and accelerometers to stabilize the helicopter.

## 2.2 The helicopter

The helicopter used in this work is an Align TREX 600E model helicopter. The helicopter is controlled through the Helicommand system which has four control signals; collective pitch, roll, elevation and yaw.

The collective pitch controls the helicopter vertically during normal flight. It changes the collective angle of attack of the main rotor. This means that the collective pitch control signal increases or decreases the angle of attack $\beta$ with the same amount at all positions $\xi$ of the rotor blades, see Figure 2.3.

The roll and elevation signals control the helicopter in the horizontal plane. Roll makes the helicopter tilt left and right, while elevate makes it tilt forwards and backwards.

The roll and elevation control signals change the cyclic angle of attack of the main rotor. This means that the angle of attack $\beta$ differs at different positions of the rotor blades $\xi$, see Figure 2.3. This makes the rotor lift more on one side of the helicopter and less on the other, causing the rotor or the whole helicopter to tilt towards a certain direction.

The yaw control signal adjusts the total angle of attack of the tail rotor. This allows controlling of the heading of the helicopter.

The explanation above is simplified. The control signals have many cross couplings. For example, a change in the roll or elevation control signal will affect the helicopter vertically as well. The Helicommand reduces these cross couplings and in this work they are neglected.

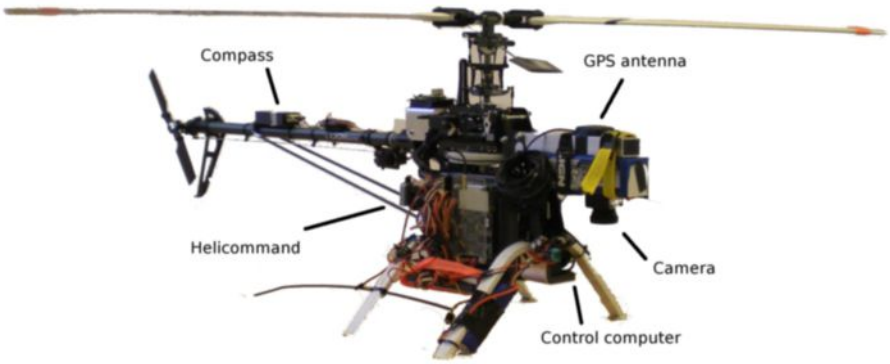
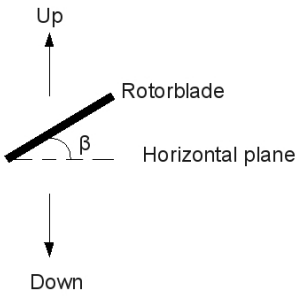
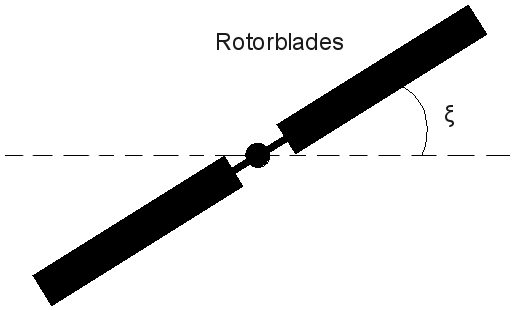



**Figure 2.2.** The helicopter and its equipment.




(a) A rotor blade viewed from a short end. $\beta$ is the angle of attack.




(b) Two rotor blades viewed from above. $\xi$ defines the position of the rotor blades.

**Figure 2.3.** The collective pitch control signal changes $\beta$ with the same amount at all position of the rotor blades $\xi$. The roll and the elevate control signals changes $\beta$ differently depending on $\xi$. In other words, the collective pitch control signal changes the collective angle of attack and the roll and the elevate control signals changes the cyclic angle of attack.

## 2.3 Coordinate systems definitions

There are four different coordinate systems. The ground ($g$), platform ($p$), helicopter ($h$) and camera ($c$) coordinate system. Figure 2.4 illustrates coordinate system $g$, $h$ and $c$. Coordinate system $g$ is fixed to the earth with the $x$-axis pointing to the east, the $y$-axis to the north and the $z$-axis upwards. Coordinate system ($p$) is fixed in the platform, where the helicopter is supposed to land. The platform is always lying flat on the ground. This means that coordinate system ($g$) and coordinate system ($p$) have their $z$-axes in the same direction. Coordinate system ($h$) is fixed in the helicopter. The $x$-axis is pointing to the right, the $y$-axis to the front and the $z$-axis upwards. Coordinate system ($c$) is fixed in the camera. The camera is fixed in the helicopter which means that the transformation between coordinate system $c$ and coordinate system $h$ is constant.

The helicopter orientation relative to the ground coordinate system is parametrized using Euler angles $\theta$, $\varphi$ and $\gamma$. $\theta$ is the heading, $\varphi$ the roll and $\gamma$ the pitch of the helicopter, see Figure 2.5. In many similar applications unit quaternions are used to avoid the singularities when the Euler angles switches between $-\pi$ and $\pi$. For $\varphi$ and $\gamma$ this singularity causes a problem when the helicopter is upside down and for $\theta$ when the helicopter is pointing south. In this thesis the helicopter never flies upside down. When the helicopter is pointing south a special fix has been implemented, which will be discussed more in Section 3.3.2.

Figure 2.4 shows the relations between three of the coordinate systems. The transformations between the coordinate systems are described with a translation vector and a rotation matrix. The translation vector is called $T^z_{\mathrm{x2y}}$ which means the translation vector from coordinate system $x$ to coordinate system $y$ given in coordinate system $z$. The translation vector from the helicopter coordinate system

$(h)$ to the camera coordinate system $(c)$ given in the helicopter coordinate system would be called $T^{h}_{\mathrm{h2c}}$. The rotation matrix is called $R_{\mathrm{x2y}}$ which means the rotation matrix from coordinate system $x$ to coordinate system $y$. The rotation matrix from the ground coordinate system $(g)$ to the helicopter coordinate system $(h)$ is called $R_{\mathrm{g2h}}$.

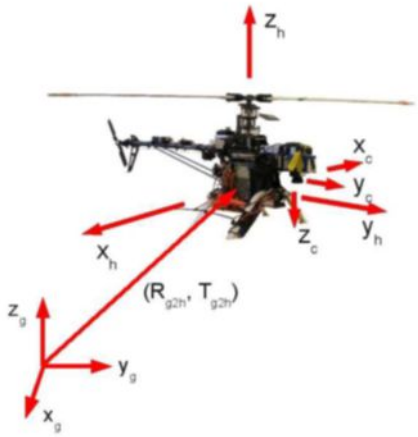



**Figure 2.4.** Definition of the helicopter and the camera coordinate system and their relation to the ground coordinate system.

(a) Definition of $\theta$ (b) Definition of $\varphi$ (c) Definition of $\gamma$

**Figure 2.5.** The helicopter orientation is parametrized using Euler angles $\theta$, $\varphi$ and $\gamma$, where $\theta$ is the heading, $\varphi$ the roll and $\gamma$ the pitch of the helicopter.

## 2.4 Coordinate systems transformations

This section describes the transformations between some of the coordinate systems described in Section 2.3.

### 2.4.1 Ground to helicopter coordinate system

The helicopter orientation is defined by its roll ($\varphi$), pitch ($\gamma$) and yaw ($\theta$) angle, see Figure 2.5. If $x_g$ is an arbitrary vector in the ground coordinate system the corresponding vector $x_h$ in the helicopter coordinate system becomes

$$x_h = R_{\text{g2h}}(x_g - T^g_{\text{g2h}}) = ABC\left[x_g - \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}\right], \tag{2.1}$$

where $A$, $B$ and $C$ describe the rotation in roll, pitch and yaw and $X$, $Y$ and $Z$ are the helicopter position in the ground coordinate system

$$A = \begin{pmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{pmatrix}, \tag{2.2}$$

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & \sin(\gamma) \\ 0 & -\sin(\gamma) & \cos(\gamma) \end{pmatrix}, \tag{2.3}$$

$$C = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.4}$$

One should know that the order of the rotations are important. In other words $ABC \neq ACB$. In [10] Diebel gives more throughout description of Euler angles.

### 2.4.2 Platform to helicopter coordinate system

If $x_p$ is an arbitrary vector in the platform coordinate system the corresponding vector $x_h$ in the helicopter coordinate system becomes

$$x_h = R_{\text{p2h}}(x_p - T^p_{\text{p2h}}) = ABC\left(x_p - T^p_{\text{p2h}}\right), \tag{2.5}$$

where $A$, $B$ and $C$ describes the rotation in roll, pitch and yaw and $T^p_{\text{p2h}}$ is the vector from the platform to the helicopter given in the platform coordinate system.

The orientation of the platform coordinate system is the ground coordinate system rotated around the z-axes $P_\theta$ degrees and translated with some vector $T^g_{\text{g2p}}$ depending on the platforms position. Because of this simple relation between coordinate system $g$ and coordinate system $p$ the rotation matrix $R_{\text{p2h}}$ is similar to $R_{\text{g2h}}$ in (2.1)

$$A = \begin{pmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{pmatrix}, \tag{2.6}$$

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & \sin(\gamma) \\ 0 & -\sin(\gamma) & \cos(\gamma) \end{pmatrix}, \tag{2.7}$$

$$C = \begin{pmatrix} \cos(\theta - P_\theta) & \sin(\theta - P_\theta) & 0 \\ -\sin(\theta - P_\theta) & \cos(\theta - P_\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.8}$$

### 2.4.3 Camera to helicopter coordinate system

If $x_c$ is an arbitrary vector in the camera coordinate system the corresponding vector $x_h$ in the helicopter coordinate system becomes

$$x_h = R_{\text{c2h}} x_c + T^h_{\text{h2c}}, \tag{2.9}$$

where $R_{\text{c2h}}$ describes the rotation between the coordinate systems and $T^h_{\text{h2c}}$ is the vector from the helicopter to the camera in the helicopter coordinate system. Both $R_{\text{c2h}}$ and $T^h_{\text{h2c}}$ are constant because the camera is fix in the helicopter. $T^h_{\text{h2c}}$ depends on where in the helicopter coordinate system the camera is fixed. In this case it has been measured with a ruler to

$$T^h_{\text{h2c}} = \begin{pmatrix} -1 \\ 21 \\ -4 \end{pmatrix} [\text{cm}].$$

Figure 2.4 shows that the $x$-axis of the camera coordinate system is in the opposite direction of the $x$-axis of the helicopter coordinate system. The $y$-axes of the two coordinate systems are in the same direction and the $z$-axes of the coordinate systems point in opposite directions. This gives the rotation matrix

$$R_{\text{c2h}} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

# Chapter 3

# Sensor fusion

## 3.1 Motion model

A model is used to predict the motion of the helicopter. In this model, constant acceleration and constant Euler angle rates are assumed. A more realistic assumption than constant Euler angle rates would be constant angular velocity. But assuming constant Euler angle rates gives a more simple implementation. Also, when the roll $\varphi$ and pitch $\gamma$ angles are small, which they are most of the time, constant Euler angle rates is a good approximation for constant angular velocity.

The control signals are not modeled because a simple linear correlation between the control signals and the output has not been found.

### 3.1.1 Continuous-time model

The model given on state space form is

$$\dot{\bar{X}} = A\bar{X} + B\bar{u} + \bar{w},$$

where $\bar{X}$ is the state vector, $\bar{u}$ the control signals vector and $\bar{w}$ the noise. The state, $\bar{X}$, of the helicopter is

$$\bar{X} = \begin{bmatrix} \ddot{X} & \ddot{Y} & \ddot{Z} & \dot{X} & \dot{Y} & \dot{Z} & X & Y & Z & \dots \\ \dots & \dot{\theta} & \dot{\varphi} & \dot{\gamma} & \theta & \varphi & \gamma & P_X & P_Y & P_Z & P_\theta \end{bmatrix}^T \tag{3.1}$$

where $X$, $Y$ and $Z$ are the position of the helicopter in the ground coordinate system. $\theta$, $\varphi$ and $\gamma$ describes the orientation of the helicopter, see Figure 2.5. $P_X$, $P_Y$ and $P_Z$ are the position of the platform in the ground coordinate system and $P_\theta$ describes the orientation of the platform which is assumed to lie flat on the ground.

Constant acceleration and constant Euler angle rates results in the following $A$ matrix

$$A = \begin{bmatrix} \bar{0}_{3x6} & \bar{0}_{3x3} & \bar{0}_{3x3} & \bar{0}_{3x7} \\ I_{6x6} & \bar{0}_{6x3} & \bar{0}_{6x3} & \bar{0}_{6x7} \\ \bar{0}_{3x6} & \bar{0}_{3x3} & \bar{0}_{3x3} & \bar{0}_{3x7} \\ \bar{0}_{3x6} & \bar{0}_{3x3} & I_{3x3} & \bar{0}_{3x7} \\ \bar{0}_{4x6} & \bar{0}_{4x3} & \bar{0}_{4x3} & \bar{0}_{4x7} \end{bmatrix}.$$

$B = \bar{0}$ because the control signals are not modeled. The noise $\bar{w}$ is caused by model inaccuracy and the wind.

Experiments to include the influence of the control signals in the model have been made without success. In these experiments Matlab (System Identification Toolbox) was used with data from several flights. The output of the model was the helicopter position given by the GPS and the input the control signals. Different model structures were used, but none performed better than just a constant acceleration model. Perhaps, including an IMU, measuring the accelerations and angular velocities, would result in a better model estimation.

### 3.1.2 Discrete-time model

A discrete-time version of the state space model

$$\dot{\bar{X}}_{t+1|t} = F\bar{X}_{t+1|t} + G\bar{u}_t + \bar{w}_t,$$

is required for the extended Kalman filter described in Section 3.3. According to [11], the $F$ and $G$ matrices can be calculated from the continuous state space matrices as

$$F = e^{AT}, \tag{3.2}$$

$$G = \int_0^T e^{AT} B dT. \tag{3.3}$$

The Maclaurin series of (3.2) gives

$$F = I + AdT + \frac{A^2}{2}dT^2 + \frac{A^3}{6}dT^3 + ... + \frac{A^n}{n!}dT^n + ... \tag{3.4}$$

By studying the $A$ matrix it turns out that

$$A^3 = \bar{0}. \tag{3.5}$$

This gives

$$A^n = \bar{0} \text{ for } n \geq 3. \tag{3.6}$$

Furthermore, (3.4) and (3.6) give

$$F = I + AdT + \frac{A^2}{2}dT^2. \tag{3.7}$$

$$F = \begin{bmatrix} I_{3x3} & & & & & & \\ dTI_{3x3} & I_{3x3} & & & & & \\ dT^2\bar{I}_{3x3} & dTI_{3x3} & I_{3x3} & & & & \\ & & & I_{3x3} & & & \\ & & & dTI_{3x3} & I_{3x3} & & \\ & & & & & I_{3x4} \end{bmatrix}.$$

Finally, (3.3) gives $G = \bar{0}$ because $B = \bar{0}$.

## 3.2 Sensors

Three different sensors were used, a GPS, a compass and a camera. The GPS provides biased absolute position measurements and velocity measurements. The compass measures the heading of the helicopter which is required to know the helicopter's orientation relative to the GPS measurements. The camera provides measurements of the platform's position relative to the helicopter. The different sensors are described in more detail below.

### 3.2.1 GPS

The GPS gives the position and the velocity of the helicopter by measuring the time it takes the radio signal to travel from the satellites to the receiver. The distance from the receiver to the satellites can be calculated by knowing the speed of the radio signals (the speed of light). If the distances to at least four satellites and the positions of these satellites are known, the position of the receiver can be calculated using geometry. The velocity is estimated from the position measurements.

There are a few different sources of errors in the GPS measurements. Geometry is important, if the angles between the satellites seen from the receiver are small the measurement becomes inaccurate. Other error sources are inaccurate estimation of satellite positions, the signals from the satellites reflecting on objects in the environment before reaching the receiver and time measurement inaccuracies. All these errors result in a time varying bias in the GPS measurement. Figure 3.1 shows the output from a GPS lying still on the ground. During three minutes only, the stationary GPS seems to have moved more than one meter.

The error in the GPS output is also affected by which satellites it is using as different satellite combinations give different errors. This causes a sudden change in the position measurements when the satellite configuration changes, see Figure 3.1.

Without the landing platform nearby, there are no other position measurements than the GPS. Therefore, it is impossible to compensate for the bias in the GPS measurements. Instead the measurement with the bias is accepted as being the truth. This makes it interesting to know the variance of the noise in the measurements excluding the bias. This variance was estimated by approximating the bias with a polynomial of degree two. The polynomial was fitted to GPS measurements over a short time interval. Figure 3.2 shows the polynomial fitted

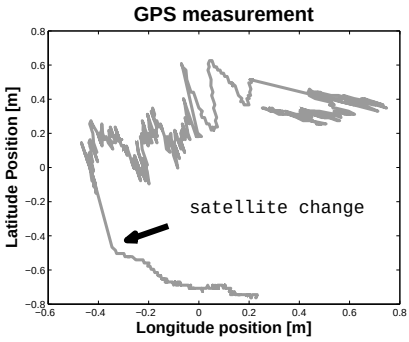



**Figure 3.1.** Measurements from the GPS lying still on the ground for three minutes. The bias in the measurements is obvious as the actual position is constant. The arrow shows a sudden jump in the measurements caused by a change in the satellites included in the GPS position estimation.

to the GPS measurements and the measurements with the bias removed. The variance of the noise came out as 0.0017 $m^2$ corresponding to standard deviation of approximately 0.04 m.

It is also interesting to know the variance of the GPS velocity measurements when the helicopter is moving. The actual change in velocity is approximated with a second order polynomial and is removed from the measurements. The variance comes out as 0.0025 $m^2/s^2$ which corresponds to a standard deviation of 0.05 m/s.

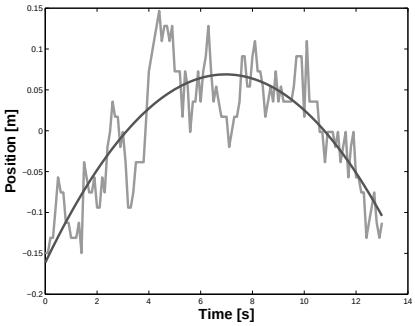
### 3.2.2 Compass

The compass gives the orientation of the helicopter by measuring the magnetic field in three directions and the direction to the gravitational force measured with a tilt sensor. The tilt sensor works like a bubble level, it measures the angle between the compass and a liquid surface. Therefore the tilt sensor is affected by external forces acting on the compass. This means that during acceleration the compass roll and pitch measurements are inaccurate. Therefore, only the heading measurement of the compass is used.

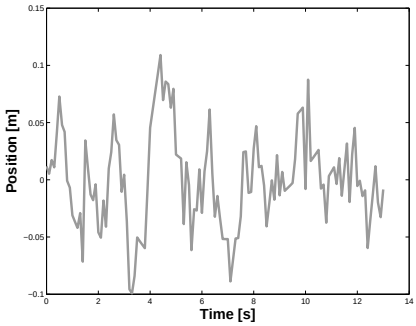
Figure 3.3 shows the calculated variance of the compass measurements. The data comes from a real flight test where the helicopter is turning slowly. In order to calculate the variance, the movement is approximated with a second order polynomial. The variance of the measurements is 0.00016 $rad^2$ which gives a standard deviation of 0.013 rad.

### 3.2.3 Camera/Vision system

Knowing the GPS coordinates of the landing platform is not enough for an accurate landing because of the bias in the GPS measurements, see Section 3.2.1. Therefore, a vision system is used together with the GPS in the landing phase. The vision

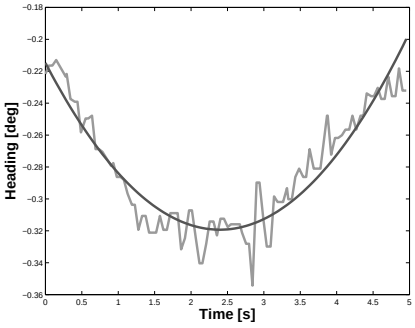



(a) A second order polynomial fitted to the GPS measurements.

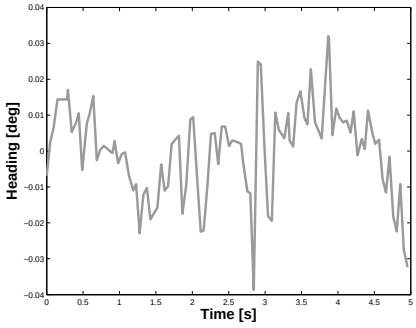



(b) GPS measurements with the bias removed.

**Figure 3.2.** Latitude measurements from the GPS lying still on the ground for 12 seconds. The bias is approximated by fitting a second order polynomial to the measurements in (a). (b) shows the measurements with the bias removed. The variance of the measurements in (b) is 0.0017 $m^2$.




(a) A second order polynomial fitted to the compass measurements.




(b) Compass measurements with the movement removed.

**Figure 3.3.** Measurements from the compass during a flight test. The helicopter is turning and this movement is approximated with a polynomial in (a). (b) shows the measurements with the movement removed in order to get the variance. The variance of the measurements in (b) is 0.00016 $rad^2$.

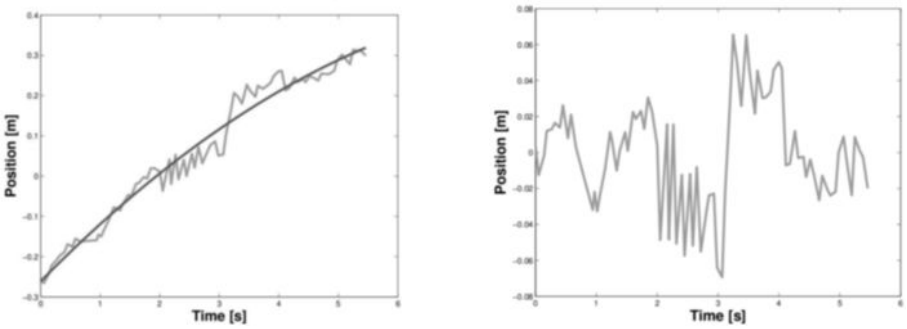
system uses camera images together with a predefined pattern on the platform, see Figure 3.4, to estimate the helicopter pose relative to the platform. This is done by finding points in the camera image with known positions in the platform coordinate system. In this case the pattern consists of rectangles and the corners of the rectangles are these known points.

**Figure 3.4.** A photograph of the pattern provided landing platform. The platform is 125x125 cm.

The vision system is based on another thesis previously carried out at CybAero [1]. In that thesis Henrik Salomonsson and Björn Saläng developed a similar system. Their system though, was too slow and the pattern used was too small for the application so major changes to their algorithm and the pattern have been made. Before the changes the system ran at 3 Hz on a 640x480 pixels image. The system developed in this thesis is limited by the camera frame rate, which is 30 Hz. The speedup comes from using a tracking algorithm. When the pattern has been found for the first time an approximate position of the pattern is known. By using this knowledge, only a smaller part of the image needs to be used in order to find the corners which results in a much faster system. A larger pattern is used in this thesis than in [1] to be able to detect the pattern at higher altitudes.

In [1] the vision system was tested using a robot arm in order to get a ground truth to compare the estimates with. Their tests showed that the accuracy is far better than required for this application. It would have been interesting to make a similar test of the new vision system, but the schedule did not allow this. Instead the variance of the estimates are approximated by using data with no ground truth given. The movement of the camera is approximated with a least-square fitted second order polynomial. Figure 3.5 (a) shows the $y$-position measurements when the camera is approximately 2.4 m above the ground. In Figure 3.5 (b) the movement of the camera has been removed from the measurements in order to get the noise. The standard deviation of the noise was estimated to 2.8 cm. The same calculations were made for all the six measurements of the camera at both 2.4 m above the platform and at 1.5 m above the platform. The result is shown in Table 3.1.

The vision system algorithms are described in two parts. The first part describes how the corners of the rectangles are found and the second part describes how the corners are used to estimate the helicopter pose.




(a) A second order polynomial fitted to the camera measurements.

(b) Camera measurements with the movement removed

**Figure 3.5.** Position estimates in the $y$-direction from the vision system. The camera movement is approximated with a polynomial in (a). (b) shows the measurements with the movement removed in order to get the variance. The variance of the measurements in (b) is 0.00078 $m^2$.

| Height[m] | $x$[cm] | $y$[cm] | $z$[cm] | $\theta$[deg] | $\varphi$[deg] | $\gamma$[deg] |
|---|---|---|---|---|---|---|
| 2.4 | 2.5 | 2.8 | 1.1 | 0.69 | 0.97 | 0.63 |
| 1.5 | 1.4 | 1.7 | 0.6 | 0.57 | 0.86 | 0.51 |

**Table 3.1.** The standard deviation of the camera measurements at different heights.

### Finding pattern corners

The algorithm to find the pattern corners can be divided into two parts. The first part finds the position of the camera with no previous knowledge. The second part uses the knowledge of an approximate camera position in order to find the pattern faster. Algorithm 3.1 describes both parts in detail.

The first part is based on another thesis [1] previously carried out at CybAero. It searches for the pattern in the whole image. In [1] two different patterns were used, one small for short distances and a larger one when the camera is further away. The smaller pattern consists of three rectangles and the larger of two rectangles in order to separate the patterns. In this thesis though, only one pattern, with rectangles of different sizes, is used. Any corner of the pattern that is currently visible in the image is used for estimating the camera position.

In order to find the pattern for the first time the corners of the two largest blobs in the image are found using [1]. The problem is to identify which rectangles these corners belong to. All likely combinations of two blobs are tested. For each combination the camera pose is calculated assuming that the corners belong to the current combination. Based on that, all other corners in the image are found using this camera pose. In order to validate the pose the expected positions of the corners in the image are calculated using this pose. The number of corners that

**Algorithm 3.1** Finding pattern

```
if pattern found == false then
  Find the two largest blobs b_1 and b_2
  for all combinations of two blobs b_a and b_b in the platform pattern do
    calculate the camera pose given that b_1 := b_a and b_2 := b_b
    if camera position is valid then
      pattern found := true
      return
    end if
  end for
else
  track pattern according to Algorithm 3.2
  calculate position
  if camera position is not valid then
    pattern found = false
  end if
end if
```

are in a certain distance from its expected corner are calculated. If the number is larger than eight the pose is valid, otherwise the next combination of two blobs is tried. This algorithm also takes care of the case when one of the blobs found does not belong to the pattern. Then the pattern will still be considered as not found and a new search is performed on the next image.

The second part of the algorithm uses the knowledge of an approximate pose of the camera in order to find the pattern faster. In this case it is not necessary to search through the whole image. The corners of the rectangles are found by first calculating their expected centers and the expected corners given the pose of the camera from the last image. After that, the corners of the rectangles are found by using Algorithm 3.2.

### Estimating the helicopter pose

The goal is to estimate the pose of the helicopter relative to the platform in the platform coordinate system. The position of the helicopter relative the platform is called $T^p_{\text{p2h}}$, see Figure 3.6, and the orientation is given by the rotation matrix $R_{\text{p2h}}$.

Assume that the corners of the pattern rectangles have been found using Algorithm 3.1. The function cvFindExtrinsicCameraParams2 in OpenCV [12] uses the position of these corners in the image, together with their true position in the platform coordinate system, to calculate the pose of the platform relative to the camera. The function also requires the calibration of the camera described in [13]. The result of the function is the translation vector $T^c_{\text{c2p}}$ and the rotation matrix $R_{\text{p2c}}$.

In the following discussion $x_c$, $x_p$ and $x_h$ are arbitrary vectors in coordinate systems $c$, $p$ and $h$. The camera and the platform coordinate systems are related

**Algorithm 3.2** Track pattern

calculate the expected corners and centers of the blobs given an approximate position of the camera
calculate a *threshold* according to the old vision system [1]
**for** each blob $b$ **do**
  get the expected middle $m$ of $b$
  **for** each expected corner $c$ of $b$ **do**
    set $v$ to the normalized vector from $m$ to $c$
    set the new corner $c_{\text{new}} = m$
    set $moved = true$
    **while** $moved$ **do**
      set $moved = false$
      set $v_{45}$ to $v$ rotated 45 degrees CW
      set $v_{-45}$ to $v$ rotated 45 degrees CCW
      **if** image intensity at $c_{\text{new}} + v \geq threshold$ **then**
        set $c_{\text{new}} = c_{\text{new}} + v$
        set $moved = true$
      **else if** image intensity at $c_{\text{new}} + v_{45} \geq threshold$ **then**
        set $c_{\text{new}} = c_{\text{new}} + v_{45}$
        set $moved = true$
      **else if** image intensity at $c_{\text{new}} + v_{-45} \geq threshold$ **then**
        set $c_{\text{new}} = c_{\text{new}} + v_{-45}$
        set $moved = true$
      **end if**
    **end while**
    the new position of $c$ is $c_{\text{new}}$
  **end for**
**end for**

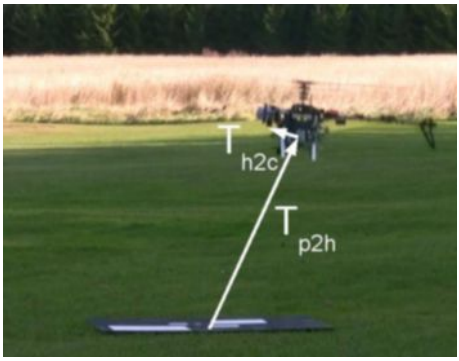



**Figure 3.6.** Definition of $T_{\mathrm{p2h}}$ and $T_{\mathrm{h2c}}$

as

$$x_c = R_{\mathrm{p2c}} x_p + T^c_{\mathrm{c2p}}. \tag{3.8}$$

Calibration of the camera pose relative the helicopter gives $R_{\mathrm{c2h}}$ and $T^h_{\mathrm{h2c}}$, see Figure 3.6, and the transformation,

$$x_h = R_{\mathrm{c2h}} x_c + T^h_{\mathrm{h2c}}, \tag{3.9}$$

between these coordinate systems. The goal is to calculate $R_{\mathrm{p2h}}$ and $T^p_{\mathrm{p2h}}$ so that

$$x_h = R_{\mathrm{p2h}}(x_p - T^p_{\mathrm{p2h}}). \tag{3.10}$$

From (3.8) and (3.9) we have

$$x_h = R_{\mathrm{c2h}}(R_{\mathrm{p2c}} x_p + T^c_{\mathrm{c2p}}) + T^h_{\mathrm{h2c}} = R_{\mathrm{c2h}} R_{\mathrm{p2c}} x_p + R_{\mathrm{c2h}} T^c_{\mathrm{c2p}} + T^h_{\mathrm{h2c}}. \tag{3.11}$$

Identification with (3.10) gives

$$R_{\mathrm{p2h}} = R_{\mathrm{c2h}} R_{\mathrm{p2c}}, \tag{3.12}$$

$$T^p_{\mathrm{p2h}} = -R^T_{\mathrm{p2h}} R_{\mathrm{c2h}} T^c_{\mathrm{c2p}} - R^T_{\mathrm{p2h}} T^h_{\mathrm{h2c}} = -R^T_{\mathrm{p2c}} T^c_{\mathrm{c2p}} - R^T_{\mathrm{p2c}} R^T_{\mathrm{c2h}} T^h_{\mathrm{h2c}}. \tag{3.13}$$

The Euler angles can be calculated from the rotation matrix as,

$$\varphi = \arcsin(-R_{\mathrm{p2h}_{13}})$$

$$\theta - P_\theta = \begin{cases} \arcsin(\frac{R_{\mathrm{p2h}_{12}}}{\cos(\varphi)}) & \text{if} \quad \frac{R_{\mathrm{p2h}_{11}}}{\cos\varphi} > 0 \\ \pi - \arcsin(\frac{R_{\mathrm{p2h}_{12}}}{\cos(\varphi)}) & \text{if} \quad \frac{R_{\mathrm{p2h}_{11}}}{\cos\varphi} < 0 \text{ and } \arcsin\left(\frac{R_{\mathrm{p2h}_{12}}}{\cos(\varphi)}\right) > 0 \\ -\pi - \arcsin(\frac{R_{\mathrm{p2h}_{12}}}{\cos(\varphi)}) & \text{otherwise} \end{cases}$$

$$\gamma = \arcsin\left(\frac{R_{\mathrm{p2h}_{23}}}{\cos(\varphi)}\right),$$

when assuming that $-\frac{\pi}{2} < \varphi < \frac{\pi}{2}$ and $-\frac{\pi}{2} < \gamma < \frac{\pi}{2}$ which is the case during normal flight.

## 3.3 State estimation

An extended Kalman filter (EKF) was used to fuse the sensor data into a state estimate of the helicopter and the platform. The filter consists of two parts. One part (the time update) uses a model of the system, see Section 3.1, to predict the state of the helicopter. This part will increase the uncertainty of the state estimate. The other part (the measurement update) uses sensor data to update the state. This part decreases the uncertainty of the state estimate.

### 3.3.1 Predict using the model

In this work a linear model of the helicopter motion is used, see Section 3.1.2. Therefore, the extended Kalman filter simplifies to a Kalman filter for the prediction step. The equations are given in [14] as:

$$\bar{X}_{[\mathrm{t+1|t}]} = F\bar{X}_{[\mathrm{t|t}]} + Gu_t, \tag{3.14}$$

$$P_{[\mathrm{t+1|t}]} = FP_{[\mathrm{t|t}]}F^T + Q, \tag{3.15}$$

where $F$ and $G$ are the discrete state space matrices, $Q$ is the process noise covariance, $P$ is the covariance matrix of the state estimation and $\bar{X}$ is the state given in (3.1).

The process noise $Q$ is hard to estimate. The model assumes constant acceleration which, of course, is not even close to reality. The acceleration is changing and might change fast. Therefore a large process noise variance in the acceleration is expected but a too large variance gives a noisy estimate of both acceleration and velocity. Real flight tests show that a process noise variance of $0.4\ (\mathrm{m/s^2})^2$ for the acceleration gives an acceptable noise level in the predicted velocity. The process noise for the velocity and position is small because the model for the velocity and position is correct. The same reasoning explains a large process noise variance for the angular velocity ($0.17^2\ (\mathrm{rad/s})^2$) and a small process noise variance for the orientation. The process noise for the platform position comes from the bias in the GPS measurement. Instead of modeling this bias, the platform position is corrected. Therefore the process noise variance for the platform position should correspond to the variation of the bias in the GPS measurements which is approximately $0.05^2\ \mathrm{m^2/s^2}$. The process noise in the platform orientation is small because the platform does not rotate which corresponds with the model. To summarize

$$Q = dT \cdot \mathrm{diag}(0.4^2, 0.4^2, 0.4^2, 0.0001^2, 0.0001^2, 0.0001^2, 0.0001^2, 0.0001^2, 0.0001^2, 0.17^2, 0.17^2, 0.17^2, 0.0001^2, 0.0001^2, 0.0001^2, 0.05^2, 0.05^2, 0.05^2, 0.0001^2).$$

### 3.3.2 Measurement update using sensor data

The update step is given in Algorithm 3.3. $y_{t+1}$ is the measurement vector and $R$ the measurement covariance matrix. $h(\bar{X}_{[t+1|t]})$ is the expected measurement given the state $\bar{X}_{[t+1|t]}$ and $H$ is the Jacobian of $h$ with respect to $\bar{X}$,

$$H = \frac{\partial h}{\partial \bar{X}} = \begin{bmatrix} \frac{\partial h_1}{\partial X} & \cdots & \frac{\partial h_1}{\partial P_\theta} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \frac{\partial h_n}{\partial X} & \cdots & \frac{\partial h_n}{\partial P_\theta} \end{bmatrix}.$$

**Algorithm 3.3** Extended Kalman filter measurement update

$\epsilon_{t+1} = y_{t+1} - h(\bar{X}_{[t+1|t]})$
$S = HP_{[t+1|t]}H^T + R$
$K = P_{[t+1|t]}H^T S^{-1}$
$\bar{X}_{[t+1|t+1]} = \bar{X}_{[t+1|t]} + K\epsilon_{t+1}$
$P_{[t+1|t+1]} = (I - KH)P_{[t+1|t]}$

If the expected measurement is a linear combination of the state, the extended Kalman filter simplifies to a Kalman filter. In that case the predicted measurement becomes

$$h(\bar{X}_{[t+1|t]}) = H\bar{X}_{[t+1|t]}.$$

#### Update using GPS data

The GPS measurements are transformed into the ground coordinate system before they are used in the filter,

$$y = \begin{bmatrix} \text{longitude position} \\ \text{latitude position} \\ \text{altitude position} \\ \text{longitude velocity} \\ \text{latitude velocity} \\ \text{altitude velocity} \end{bmatrix}.$$

The observation model is linear and $H$ comes out as

$$H = \begin{bmatrix} \bar{0}_{6x3} & I_{6x6} & \bar{0}_{6x10} \end{bmatrix}.$$

The variance of the GPS measurements are calculated in Section 3.2.1. As discussed in this section there are errors in the measurements which will give a different variance from time to time. Therefore, a larger variance is used than the one calculated in Section 3.2.1. For the position, the standard deviation was calculated to 0.04 m. In the filter, a standard deviation of 0.1 m is used. For the velocity, the standard deviation was calculated to 0.05 m/s which is increased to

0.1 m/s in the filter. The GPS do not measure the vertical position with the same accuracy as the horizontal position, hence a larger variance is used for the altitude measurements. The resulting measurements covariance matrix is

$$R = \begin{bmatrix} 0.1^2 & & & & & & \\ & 0.1^2 & & & & & \\ & & 0.2^2 & & & & \\ & & & 0.1^2 & & & \\ & & & & 0.1^2 & & \\ & & & & & 0.2^2 \end{bmatrix}.$$

**Update using compass data**

The compass measures the heading of the helicopter $\theta_{\text{compass}}$ in the interval $[-\pi, \pi]$. A problem occurs when the helicopter is pointing south. A possible situation would be that the compass measures something very close to $-\pi$ and the heading of the helicopter is close to $\pi$. This compass measurement agrees with the heading, but the filter would interpret this as a large error. Therefore, the compass measurement is transformed into the interval where it satisfies $|\theta_{\text{compass}} - \theta| < \pi$,

$$y = \begin{cases} \theta_{\text{compass}} & \text{if} \quad |\theta_{\text{compass}} - \theta| <= \pi \\ \theta_{\text{compass}} - \pi & \text{if} \quad \theta_{\text{compass}} - \theta < \pi \\ \theta_{\text{compass}} + \pi & \text{if} \quad \theta_{\text{compass}} - \theta > \pi \end{cases}$$

and

$$H = \begin{bmatrix} \bar{0}_{1\text{x}12} & 1 & \bar{0}_{1\text{x}6} \end{bmatrix}.$$

The variance of the measurement is 0.00016 rad$^2$, see Section 3.2.2, therefore

$$R = 0.00016.$$

**Update using vision system data**

The measurement from the vision system, see Section 3.2.3, is

$$y = \begin{bmatrix} T^p_{\text{p2h}} \\ \theta - P_\theta \\ \varphi \\ \gamma \end{bmatrix}.$$

The observation model $h(\bar{X})$ is given by

$$T^p_{\text{p2h}} = R_{\text{g2p}} T^g_{\text{p2h}} = R_{\text{g2p}}(T^g_{\text{g2h}} - T^g_{\text{g2p}}) = R_{\text{g2p}} \left[ \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \begin{pmatrix} P_X \\ P_Y \\ P_Z \end{pmatrix} \right],$$

and $\theta$, $P_\theta$, $\varphi$ and $\gamma$ are given directly from the state.

The variance of the noise in the vision system measurements depends on the distance of the camera from the platform. Tests have been made with the camera

at 1.5 m and 2.4 m above the platform only. The results from these tests are shown in Table 3.1. Let us assume that the standard deviation in the position grows linearly with the height above the platform. The noise is also assumed to be Gaussian and the noise of the six measurements are assumed to be independent of each other. From these assumptions and from Table 3.1 the measurement covariance matrix becomes

$$R = \begin{bmatrix} (0.01h)^2 & & & & & \\ & (0.01h)^2 & & & & \\ & & (\frac{0.01h}{2})^2 & & & \\ & & & 0.017^2 & & \\ & & & & 0.017^2 & \\ & & & & & 0.017^2 \end{bmatrix},$$

where $h$ is the camera height above the platform.

### 3.3.3 Implementation

Algorithm 3.4 gives a brief description of the implementation of the EKF. The state is predicted with the model according to Section 3.3.1. When new measurements are received from any of the sensors the state estimate is update using Algorithm 3.3. The algorithm also guarantees that the estimated heading of the helicopter $\theta$ stays in the interval $[-\pi, \pi]$.

**Algorithm 3.4** State estimation

**while** Controlling **do**
  try to read data from the GPS, compass and the camera
  do the prediction according to Section 3.3.1
  **if** new data was read from the GPS **then**
    update with the GPS data according to Section 3.3.2 and Algorithm 3.3
  **end if**
  **if** new data was read from the compass **then**
    update with the compass data according to Section 3.3.2 and Algorithm 3.3
  **end if**
  **if** new data was read from the camera **then**
    update with the vision system according to Section 3.3.2 and Algorithm 3.3
  **end if**
  **if** $\theta > \pi$ **then**
    $\theta = \theta - 2\pi$
  **end if**
  **if** $\theta < -\pi$ **then**
    $\theta = \theta + 2\pi$
  **end if**
**end while**

### 3.3.4 Examples

In this section there are two different examples given of the extended Kalman filter in use. The first example uses simulated GPS data to show the performance of the filter. The reason for using simulated data is that the ground truth is known. In the second example all sensors are fused using the extended Kalman filter with real flight data.

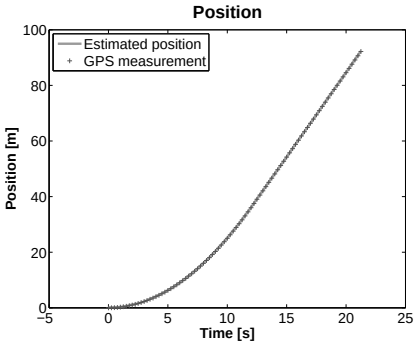
#### GPS simulation

To illustrate the EKF a simulation was run. For the simulation, GPS position data was calculated corresponding to constant acceleration forwards during twelve seconds followed by constant velocity. Before simulation, white noise with maximum error of 0.1 meter was added to the calculated GPS data. This noisy GPS data was used as input to the filter. Figure 3.7 shows the position, velocity and acceleration estimates given by the extended Kalman filter and their correspondences calculated from the undisturbed original GPS data.

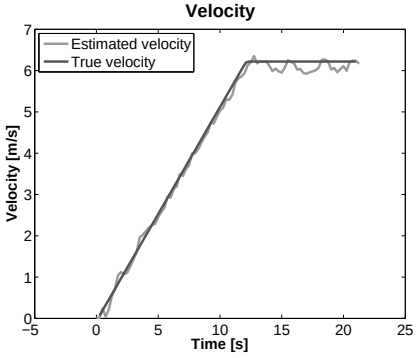
The position estimate is really good, while the velocity estimate is a bit worse and the acceleration estimation is bad. This is because the EKF estimates the velocity and acceleration from position measurements only. To get better acceleration and velocity estimates accelerometers could be used.

#### Real flight data

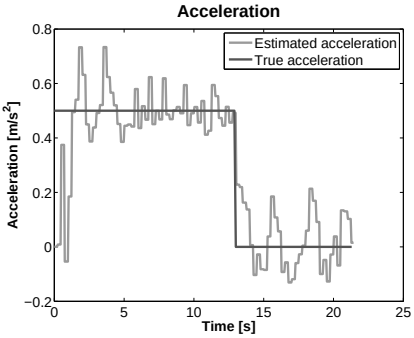
This is an example from a real flight. The EKF fuses data from all the sensors to compute an estimate of the helicopter state. First, the helicopter is hovering at the origin of the ground coordinate system and is not able to see the platform. After approximately 100 seconds it is commanded to fly to the platform. An approximate position of the platform has been given and the vision system is used to find the platform more accurately. In Figure 3.8 the GPS and the camera measurements are compared. The camera measures the platform position relative to the helicopter, but in this figure the measurements were transformed to the ground coordinate system. The reference position, which is the desired position of the helicopter, is also given in Figure 3.8. From the start the reference position is the origin of the ground coordinate system, but when the helicopter is commanded to the platform it is changed to the estimated platform position. In this test the pilot controlled the collective pitch control signal and hence the altitude. Therefore, no reference position for the altitude is given.




(a) Estimated position and the simulated GPS measurements

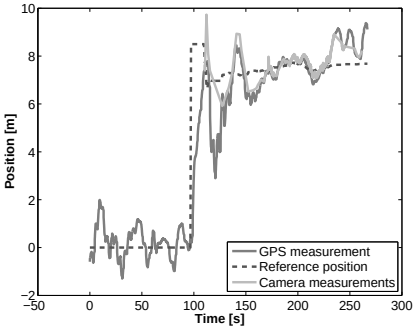



(b) Estimated velocity compared to the true simulated velocity

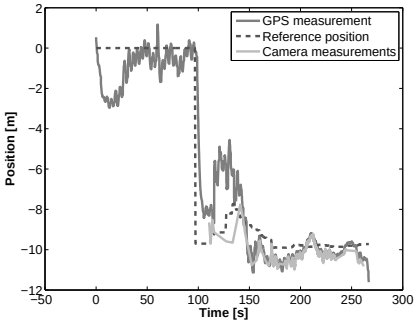



(c) Estimated acceleration compared to the true simulated acceleration

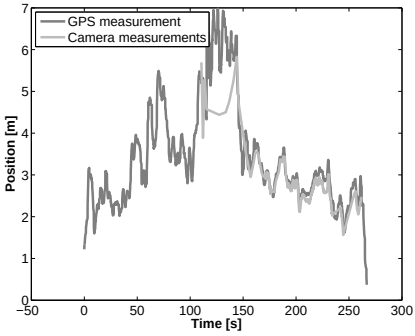
**Figure 3.7.** The EKF estimate compared to the ground truth using GPS data from a simulation.




(a) Longitude position




(b) Latitude position




(c) Altitude position

**Figure 3.8.** Figure (a)-(c) show GPS and vision system position estimates form real flight data. In order to compare the GPS and the camera measurements, the camera measurements have been transformed to the ground coordinate system. The reference position, which is the desired position of the helicopter, is also given. From the start the reference position is the origin of the ground coordinate system. After 100 seconds the helicopter is commanded to the platform and then the reference position equals the platform position estimation. The figure shows how well the camera and GPS measurements agrees in the short run.

# Chapter 4

# Control system

Figure 4.1 gives an overview of the control system. The low level controller, the high level controller and the state estimation has been developed in this work. The Helicommand is a pilot support system that uses accelerometers and gyros to stabilize the helicopter. The low level controller makes the helicopter move to a certain reference position and controls the helicopter through the Helicommand. The high level controller makes larger decisions as which position to fly to, when to land and so on. The high level controller uses the low level controller to actuate its decisions.

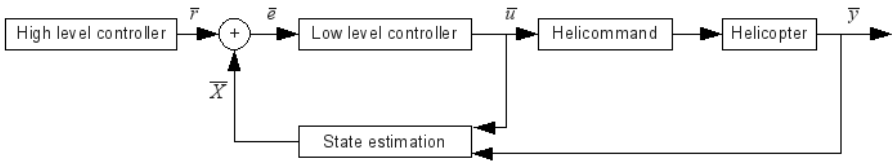



**Figure 4.1.** An overview of the controller hierarchy.

## 4.1 Low level controller

The low level controller consists of three PID-controllers and one customized P-controller. The roll, elevate and collective pitch control signals are controlled with PID-controllers and the yaw control signal uses the P-controller. The PID-controllers are implemented according to [11] p.98 which avoids integration windup.

The PID-parameters $K$, $T_d$ and $T_i$ were initially determined by step responses in the control signals. The tests were made by letting the pilot have the command of all control signals except one and applying a step in that control signal. Data from the GPS and the compass were logged during the tests. Since the system is unstable in the open loop, in other words a limited control signal can cause an

**Algorithm 4.1** PID controller

**while** controlling **do**
  get the control error $e$ and the rate at which the control error changes $\dot{e}$ from the state estimate
  set the time since last control step as $\triangle T$
  set the temporary control signal $v = Ke + I + KT_{\mathrm{d}}\dot{e}$
  **if** $u_{\min} \leq v \leq u_{\max}$ **then**
    $I = I + K\frac{\triangle T}{T_{\mathrm{i}}}e$
  **end if**
  **if** $v > u_{\max}$ **then**
    $u = u_{\max}$
  **else if** $v < u_{\min}$ **then**
    $u = u_{\min}$
  **else**
    $u = v$
  **end if**
**end while**

infinite position, common rules as Zieger Nichols rules could not be applied for the parameter determination. Instead a simple model assuming constant acceleration was fitted to the data and the parameters were adjusted by trial and error in simulations using these models.

In order to use the PID implementation in Algorithm 4.1 an estimate of the velocity and the position error for each PID-controller is needed. For the roll, elevate and collective pitch controllers this is the velocity and the position error given in the helicopter coordinate system. Assuming that the roll and the pitch angles are small results in

$$\begin{bmatrix} \dot{e}_{\mathrm{roll}} \\ \dot{e}_{\mathrm{elev}} \\ \dot{e}_{\mathrm{collective}} \\ e_{\mathrm{roll}} \\ e_{\mathrm{elev}} \\ e_{\mathrm{collective}} \end{bmatrix} \approx \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\dot{X} \\ -\dot{Y} \\ -\dot{Z} \\ \mathrm{ref}_{\mathrm{long}} - X \\ \mathrm{ref}_{\mathrm{lat}} - Y \\ \mathrm{ref}_{\mathrm{alt}} - Z \end{bmatrix}.$$

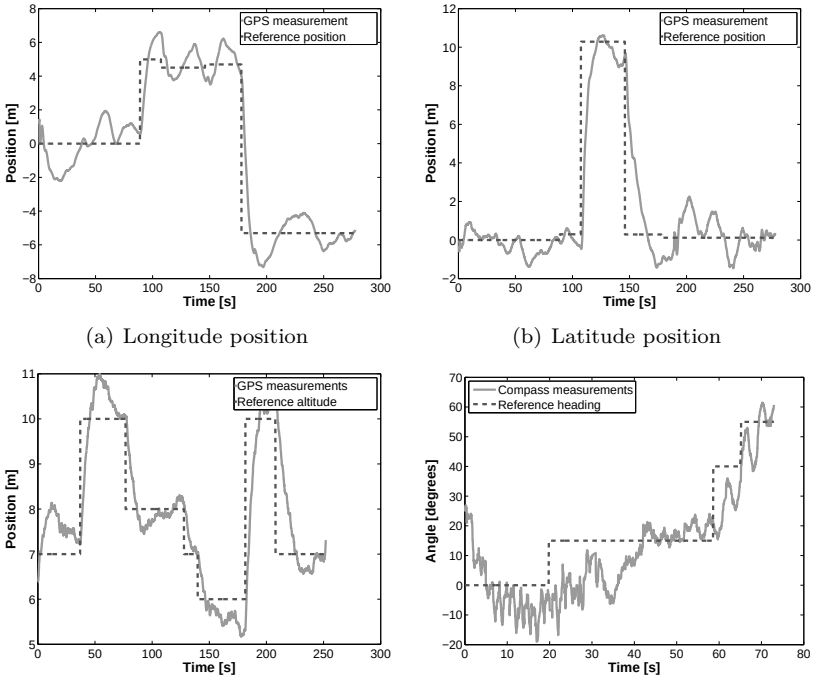
The P-controller for the yaw control signal is implemented with a dead band according to Algorithm 4.2. The reason for this dead band is that the yaw control signal is controlled through a gyro. This gyro is trying to keep the heading constant when the control signal is zero. The error in the heading, used by the P-controller, should be the smallest angle between the reference and the actual heading of the helicopter. The reference and the helicopter heading is defined in the intervals $-\frac{\pi}{2} < \theta_{\mathrm{ref}} \leq \frac{\pi}{2}$ and $-\frac{\pi}{2} < \theta \leq \frac{\pi}{2}$ which gives

$$e_{\mathrm{yaw}} = \begin{cases} \theta_{\mathrm{ref}} - \theta & \text{if} \quad -\frac{\pi}{2} \leq \theta_{\mathrm{ref}} - \theta \leq \frac{\pi}{2} \\ \theta_{\mathrm{ref}} - \theta - 2\pi & \text{if} \quad \theta_{\mathrm{ref}} - \theta > \frac{\pi}{2} \\ \theta_{\mathrm{ref}} - \theta + 2\pi & \text{if} \quad \theta_{\mathrm{ref}} - \theta < -\frac{\pi}{2} \end{cases}$$

**Algorithm 4.2** Customized P-controller for the yaw control signal

```
while controlling do
  get the control error e_yaw
  if e_yaw < 5 deg then
    u = 0
  else if e_yaw > −5 deg then
    u = 0
  else
    u = K e_yaw
  end if
end while
```

Figure 4.1 shows the performance of the controllers together with the extended Kalman filter described in Section 3.3. In the tests, the reference position was changed from time to time in order to test the controllers. The GPS measurements are compared to the reference positions. The performance of the controller is related to the weather and the wind. On windy days the performance is worse than on a calm day.




(a) Longitude position

(b) Latitude position

(c) Altitude position

(d) Heading angle

**Figure 4.2.** Figure (a)-(d) show the performance of the controllers together with the extended Kalman filter when the reference position is changed stepwise. The reference position is compared with GPS and the heading with the compass measurements.

## 4.2 High level controller

The high level controller takes top level decision for the control of the helicopter and it uses the low level controller to actuate these decisions. There are two different modes of the high level controller, assisted mode and landing mode. In assisted mode the user of the ground station is able to command the helicopter to certain positions. In landing mode the helicopter finds the platform and lands on it. These modes are described in more detail bellow.

### 4.2.1 Assisted mode

In assisted mode the user of the ground station computer is able to command the helicopter to different positions. It is also possible to change the altitude and the heading of the helicopter using the ground station laptop. In order to actuate these commands the reference positions of the controllers, see Section 4.1, are changed.

### 4.2.2 Landing mode

In landing mode the helicopter has to find the landing platform and land on it. An approximate position of the platform is given before the procedure starts. The helicopter flies to that position and detects the platform using the camera. The vision system provides measurements for the position of the platform, which are then used in the EKF, see Section 3.3. The horizontal reference position for the PID-controllers is set to the current estimate of the platform position. The altitude starts at five meters above the estimated height of the platform and is decreased when certain conditions in position and speed of the helicopter are achieved. See Algorithm 4.3 for more details.

One problem during landing is to actually get the helicopter down to the ground. When the helicopter is less than one meter from the ground the dynamics changes. The helicopter becomes more sensitive in its horizontal position and less lift is required to keep the helicopter flying. Therefore, to get the helicopter down to the surface, the last step in the landing phase is to set the reference in the altitude to 3.0 m below the platform. This takes the helicopter down fast to avoid drifting away in the horizontal position and still smooth enough to avoid damaging the helicopter.

**Algorithm 4.3** Landing algorithm

set the reference height above the platform, $rh = 5.0$m
**while** controlling **do**
  set the horizontal reference position to the estimated position of the platform
  set $d$ to the distance between the helicopter and the platform
  set $s$ to the helicopter speed
  set $a$ to the difference in the actual height above the platform and the reference height above the platform
  **if** $d < 0.7$m AND $s < 0.2$ AND $a < 0.2$ AND $rh > 2.0$m **then**
    $rh = rh$ - 1.0
  **else if** $d < 0.5$m AND $s < 0.15$ AND $a < 0.2$ AND $rh > 1.0$m **then**
    $rh = rh - 0.5$m
  **else if** $d < 0.2$m AND $s < 0.1$ AND $a < 0.2$ AND $rh > 0.0$m **then**
    $rh = -3.0$m
  **end if**
**end while**

# Chapter 5

# Results

The goal of this work is to land the helicopter on a platform with high precision. In the beginning of the project, see Section 1.3, the following milestones were identified.

- **Milestone 1:** Autonomous hovering at a GPS coordinate.
- **Milestone 2:** Autonomous hovering above the platform.
- **Milestone 3:** Autonomous landing on the platform.

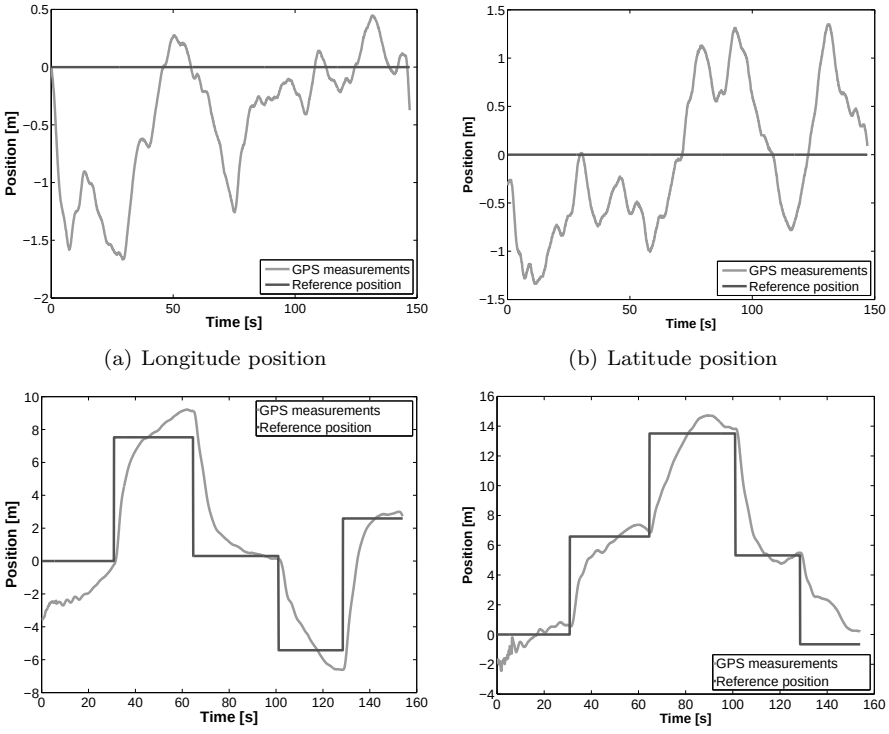
In this chapter the results of these milestones are described.

It is difficult to draw conclusions about the performance of the controllers. The controllers have been under development during the entire project and therefore there is no large data set from flights with the final version of the controllers. The helicopter is also affected by the weather conditions and especially gusty winds, which makes it even more difficult to make any general conclusions of the controllers performance.

## 5.1 Autonomous hovering at a GPS coordinate

During this milestone the control computer only had control of the roll and elevate control signals. In other words the pilot was controlling the collective pitch and the yaw control signals having control of the altitude and the heading. In Figure 5.1 the results from two different flights are shown. The GPS measurements are compared to the reference position. In the first test, Figure 5.1 (a) and (b), the helicopter was commanded to hover above the origin of the ground coordinate system. After some time when the controller had settled, the maximum error was a bit more than one meter in both the longitude and the latitude directions. In test two, Figure 5.1 (c) to (d), the reference position was changed from time to time in order to validate how the controller reacts on a step change in the reference position.

In these tests there are a few disturbances. The wind is one of them, and the helicopters angle to the wind is also important. For example the elevate




(a) Longitude position

(b) Latitude position

(c) Longitude position

(d) Latitude position

**Figure 5.1.** The GPS measurements compared to a reference position for two different tests. In the first test, (a) and (b), the helicopter was commanded to hover above the origin of the ground coordinate system. In test two, (c) to (d), the reference position was changed from time to time in order to validate how the controller reacts on a step change in the reference position.

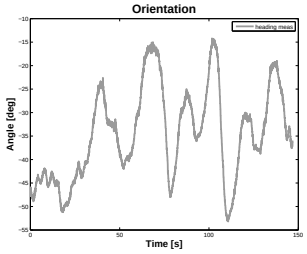



**Figure 5.2.** The orientation of the helicopter in the first test (Figure 5.1 (a) and (b)).

controller, which controls the tilt forward/backward, is mostly affected by the wind composant in the direction of the helicopter. If the heading of the helicopter changes, the wind composant affecting the elevate controller also changes. The effect of this behavior is similar to that of wind gusts. In these tests the pilot was controlling the heading and it is very hard for the pilot to hold a constant heading. Data from the onboard compass, see Figure 5.2, shows that the heading was changing with about 20 degrees back and forth. Another disturbance was the bias in the GPS measurements, see Figure 3.1.

It is important to notice that the performance of the controller has been increased since this milestone was first met.

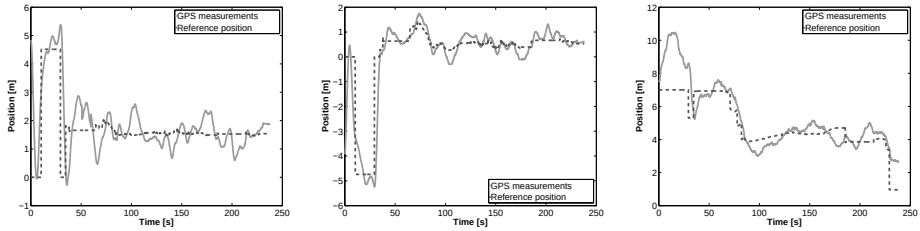
## 5.2 Autonomous hovering above the platform

When completing this milestone the control computer had control of all the control signals. This allows control of the horizontal position, the altitude and the heading. Figure 5.4 shows data from one test when the helicopter was hovering above the platform for about two minutes. In the figure the reference position is compared with the GPS measurements. The vision system compensates for the bias in the GPS measurements by changing the platform position. In this test the helicopter should hover above the platform, which causes the reference position to change with time. This test was done in heavy wind conditions for a helicopter of this size. It was gusty winds blowing at approximately 5 m/s.

## 5.3 Landing

The goal of this work was to land the helicopter with high precision on a landing platform. A perfect landing has the center of the helicopter in the middle of the platform. During the development and the tuning of the landing algorithm many landings have been performed. In the last tests the distance between the center of

**Figure 5.3.** Autonomous helicopter landing.




(a) Longitude position (b) Latitude position (c) Altitude position

**Figure 5.4.** Figure (a)-(c) show the GPS measurements compared to a reference position. The helicopter was commanded to hover above the landing platform. The vision system compensates for the bias in the GPS measurements which causes the reference position to change with time. The test was done on a very windy day.

the helicopter and the middle of the platform was measured. Fifteen landings were made in these tests and results are shown in Table 5.1. The average Euclidean distance from the landing target was 34 cm.

Figure 5.3 shows pictures from one of the landings. The helicopter starts about five meters above the platform and slowly descends in accordance to Algorithm 4.3.

| Landing | $x$[cm] | $y$[cm] | dist.[cm] | Landing | $x$[cm] | $y$[cm] | dist.[cm] |
|---|---|---|---|---|---|---|---|
| 1 | 29 | -7 | 30 | 9 | 11 | 62 | 63 |
| 2 | -22 | -27 | 35 | 10 | 25 | 20 | 32 |
| 3 | 11 | 9 | 14 | 11 | 38 | -9 | 39 |
| 4 | 20 | 10 | 22 | 12 | 44 | 7 | 45 |
| 5 | -13 | 38 | 40 | 13 | 42 | 6 | 42 |
| 6 | -27 | -4 | 27 | 14 | 7 | -13 | 15 |
| 7 | 18 | 12 | 22 | 15 | 2 | -46 | 46 |
| 8 | 13 | -40 | 42 | | | | |

**Table 5.1.** The horizontal distance between the center of the helicopter and the center of the landing platform for fifteen landings. The distance is both given as coordinates in the platform coordinate system and as the corresponding Euclidean distance.

# Chapter 6

# Concluding remarks

## 6.1 Conclusions

A control system for autonomous landing of an UAV with high precision, using a vision system has been developed. In the final tests of the landing control, fifteen landings were performed. In these landings, the average distance from the landing target was 34 cm.

A vision system to detect the platform has also been developed. The vision system gives 6-DOF measurements of the helicopter pose relative to the platform and can easily match the camera frame rate of 30 Hz. The accuracy of the vision system gets better and better the closer to the platform the camera is. At 1.5 m above the platform, the standard deviation of the vision system position estimates is about 1.7 cm.

## 6.2 Future work

- Verify robustness of the system by performing more landings in different weather conditions.
- Implementing more sensors such as accelerometers, gyros and a barometer to get better pose estimates from the EKF. Better pose estimates will result in a more accurate control.
- Landing on a moving target.
- Identifying the helicopter model and removing the pilot support system.

# Bibliography


[1] B. Saläng and H. Salomonsson, "Vision based pose estimation for autonomous helicopter landing," Master's thesis, Linköpings universitet, 2008.

[2] E. Feron and E. N. Johnson, *Ch 44. Aerial Robotics, in Springer Handbook of Robotics*, pp. 1009–1029. Berlin, Germany: Springer, 2008. Ed. Siciliano, B. and Khatib, O.

[3] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 371–380, June 2003.

[4] S. Saripalli and G. Sukhatme, "Landing a helicopter on a moving target," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Roma, Italy), pp. 2030–2035, Apr. 2007.

[5] C. Theodore, D. Rowley, D. Hubbard, A. Ansar, Matthies, S. Goldberg, and M. Whalley, "Flight trials of a rotorcraft unmanned aerial vehicle landing autonomously at unprepared sites," in *Proceedings of the American Helicopter Society 62nd Annual Forum*, (Phoenix, AZ, USA), May 2006.

[6] P. Corke, "An inertial and visual sensing system for a small autonomous helicopter," *Journal of Robotic Systems*, vol. 21, no. 2, pp. 43–51, 2004.

[7] F. Kendoul, Z. Yu, and K. Nonami, "Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles," *Journal of Field Robotics*, 2010. In press.

[8] C. Sharp and O. Shakernia, "A vision system for landing an unmanned aerial vehicle," tech. rep., University of California Berkeley, 2001.

[9] I. Jan and M. Khan, "Combined application of kalman filtering and correlation towards autonomous helicopter landing," in *IEEE International Conference on Computer, Control and Communication*, (Karachi), pp. 1–6, May 2009.

[10] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," tech. rep., Stanford University, 2006.

[11] *Industriell reglerteknik Kurskompendium*. Linköpings universitet, 2008.

[12] http://opencv.willowgarage.com/.

[13] http://www.vision.caltech.edu/bouguetj/calib_doc/.

[14] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.