DLR-IB-RM-OP-2016-347

Cooperative Control for Landing a Fixed-Wing Unmanned Aerial Vehicle on a Ground Vehicle

Masterarbeit

Linnea Persson

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Institut für Robotik und Mechatronik	BJ.: 2016
	DLR-IB-RM-OP-2016-347

MASTERARBEIT

COOPERATIVE CONTROL FOR LANDING A FIXED-WING UNMANNED AERIAL VEHICLE ON A GROUND VEHICLE

Freigabe:	Der Bearbeiter:	Unterschriften
	Linnea Persson	
	Betreuer:	
	Tin Muskardin	
	Der Institutsdirektor	
	Dr. Alin Albu-Schäffer	

Dieser Bericht enthält 86 Seiten,69 Abbildungen und 6 Tabellen

Ort: Oberpfaffenhofen	Datum: 16.12.2016	Bearbeiter: Linnea Persson	Zeichen:

EXAMENSARBETE I TEKNISK FYSIK 300 HP, AVANCERAD NIVÅ
STOCKHOLM, SVERIGE 2016

Cooperative Control for Landing a Fixed-Wing Unmanned Aerial Vehicle on a Ground Vehicle

LINNEA PERSSON

KTH KUNGLIGA TEKNISKA HÖGSKOLAN

SKOLAN FÖR ELEKTRO- OCH SYSTEMTEKNIK

Abstract

High Altitude Long Endurance (HALE) platforms are a type of Unmanned Aerial Vehicle (UAV). With their relatively easy deployment and independence of a fixed orbit, HALE UAVs have the potential to replace satellites for certain tasks in the future. A challenge with this technology is that the current platforms are too heavy to fly for a long period of time. A suggested method for reducing the weight is to remove the landing gear to instead use alternative methods for take-off and landing. One such alternative method is to land the UAV on top of a cooperating ground vehicle. In this thesis, the cooperative controller and the experimental setup of such a landing have been investigated. The cooperation between the systems was analyzed and evaluated analytically, through simulations and with flight tests. Using a PID controller for the position alignment and a modified flare law for the descent, feasibility of the landing was verified by performing a landing of a Penguin BE fixed-wing UAV on top of a cooperating ground vehicle.

Sammanfattning

Så kallade HALE - High Altitude Long Endurance -farkoster är en växande teknik inom området för autonoma flygplan. Med fördelar som exempelvis en möjlighet att röra sig oberoende av en omloppsbana samt en mer effektiv implementering– och utvecklingsprocess har de visat potential att i framtiden kunna ersätta satelliter inom vissa områden. Ett problem är i dagsläget svårigheten att bygga tillräckligt lätta farkoster för att kunna flyga under en längre tidsperiod. För att minska vikten har det bland annat föreslagits att landningsställ kan tas bort för att istället använda alternativa start- och landingsmetoder. I detta projekt har en metod undersökts där idén är att landa ett autonomt flygplan på en mobil plattform. Samarbetet mellan systemen har analyserats både analytiskt och genom tester. Slutligen verifieras att en kooperativ landning är genomförbar genom att en landning av ett obemannat flygplan på en samarbetande bil utförs.

Acknowledgment

I would like to express my gratitude to the Flying Robots group at the institute of Robotics and Mechatronics at DLR for welcoming me into their group and letting me work with them.

In particular I would like to thank my supervisor Tin Muskardin for his support throughout the project, as well as everyone else working with the flight experiments and various other things related to the EC-SAFEMOBIL project, among them Georg Balmer, Sven Wlach, Maximilian Laiacker, Jonatas Santos, Stojan Stevanovic and Dr. Konstantin Kondak.

I would also like to thank my supervisor at KTH, Prof. Bo Wahlberg, for providing support throughout my thesis project despite the distance.

Contents

Symbols and Abbreviations

Symbols

α	Angle of attack
β	Sideslip angle
γ	Vertical path angle
ω	Motor RPM
ϕ	Roll angle
ψ	Yaw angle
θ	Pitch angle
χ	Course angle
η	Elevator input
ξ	Aileron input
ζ	Rudder input
κ	Flaps input
p	Roll rate
q	Pitch rate
r	Yaw rate
u	Velocity x-direction
v	Velocity y-direction
w	Velocity z-direction
V_a	Airspeed
V_k	Groundspeed

Abbreviations

CAS	Control Augmentation System
CG	Center of Gravity
DLR	Deutschen Zentrum für Luft- und Raumfahrt (German Aerospace Center)
FE	Flight Experiment
HALE	High Altitude Long Endurance
HASP	High Altitude Pseudo Satellite
MIMO	Multiple Input Multiple Output
MPC	Model Predictive Control
SAS	Stability Augmentation System
SISO	Single Input Single Output
TECS	Total Energy Control System
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle

Chapter 1

Introduction

This master's thesis project was performed at the Robotics and Mechatronics Institute at the German Aerospace Institute (DLR). The work was done in the Flying Robots research group within Project EC-SAFEMOBIL[1]. The main goal of this project is to investigate feasibility of landing a fixed-wing Unmanned Aerial Vehicle (UAV) on top of an Unmanned Ground Vehicle (UGV). This is done as a part of the work of the Flying Robots group on High Altitude Long Endurance (HALE) platforms, also known under the name High Altitude Pseudo Satellites (HAPS). This introductory chapter starts by placing the thesis in the context of this research topic, as well as a summary of what is meant by cooperative control. This is followed by a problem formulation and contributions, where the intent of the thesis and its consequences are clarified. The chapter ends with an outline of the thesis.

1.1 High Altitude Long Endurance UAVs

A high altitude long endurance UAV is a type of fixed-wing unmanned aerial vehicle operating at high altitudes where conventional aircraft do not reach. In addition to the advantage of the lack of interference from commercial air traffic, these platforms benefit from the possibility of solar power generation and of a calm atmosphere, since the main wind currents are well below their operating altitude.

A HALE platform would enable a completely new set of mission profiles for fixed-wing UAVs, with applications including communication, surveillance, atmospheric research and long-term earth observation. Today, such tasks are commonly given to satellites, but HALE UAVs would provide advantages against satellites such as quick deployments and relatively simple repairs and replacements. For satellites this process is often both lengthy and expensive. Another advantage of such a platform is its independence of an orbit, making it possible to pinpoint an area of interest as opposed to being limited to following a predetermined trajectory.

Solar powered aircraft have a history spanning back to the 1970s [1]. Since

[1] http://www.ec-safemobil-project.eu/

then, developments in both solar energy and UAV technology have greatly improved the performance of these vehicles, and today there are several research and development projects working on making solar-powered UAVs capable of continuous flight. Previous efforts include NASA's Helios project [2], QinetiQ's Zephyr UAV [3], and the ELHASPA platform at DLR [4] which is shown in Figure 1.1. A problem with existing platforms is their inability to carry a profitable amount of payload, making this technology not yet commercially successful.

The total weight of a UAV is critical for how much time a it can stay flying, since a lighter aircraft requires less lift. While the HALE platforms typically have a large wingspan and a lightweight structure, the maximum payload is very limited. Weight becomes even more critical when considering night flying UAVs. During the daytime it is possible to fly on power generated from the solar panels, but for flight during the night the plane requires rechargeable batteries that have a relatively high weight compared to the rest of the structure.

The landing gear is one part of the fixed-wing UAV that potentially could be removed. Since it is being active only during takeoff and landing, the landing gear is dead weight for everything but a fraction of the mission life time, especially when considering long endurance flights. Removing the landing gear would free weight for use by more critical subsystems, such as payload or batteries.

1.2 Cooperative control

Cooperative control deals with the problem of controlling two or more dynamically separated systems to work together to accomplish a common goal. Made possible by the development of reliable wireless communications systems, this field is relatively new and growing. The areas where cooperative control has

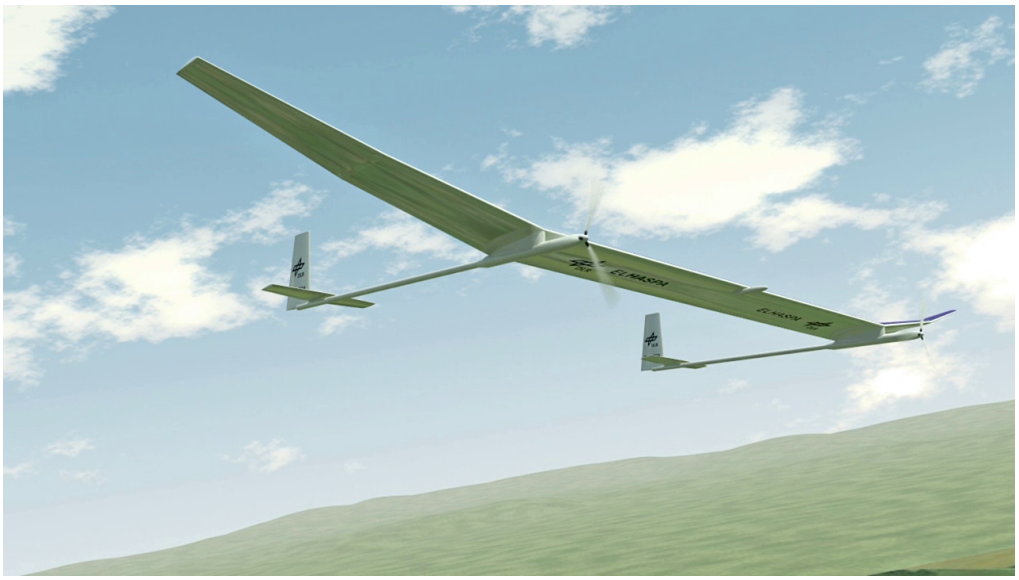


Figure 1.1: The ELHASPA platform from DLR. The UAV has a large wingspan and is covered by solar panels on the top.

applications are numerous, and the number is likely to grow as new technologies emerge. Some areas where cooperative systems can be found today are in coordination of autonomous vehicles [5], spacecraft docking([6], [7]), autonomous landing ([8],[9],[10]) aerial refueling ([11],[12]) and formation flight ([13], [14], [15]).

A distinguishing feature of cooperative control is distributed sensing, control and decision making. For systems consisting of a very large number of subsystems, even if a centralized controller was available, it might be computationally inefficient to use it. In addition, decentralized cooperative control offers a more flexible and robust solution in which the roles of agents more easily can be assigned dynamically in case of unforeseen events, as compared to a centralized controller. The distributed architecture is what gives cooperative control some of its defining challenges regarding information sharing, consensus and task division.

1.3 Problem formulation

In the framework of the EC-SAFEMOBIL project, the objective is to analyze the feasibility of landing a fixed-wing UAV without the use of a landing gear. The particular solution examined is to use cooperative control to land the UAV on top of a UGV. The vehicles will collaborate in the lateral and longitudinal alignment to make landing possible, and the UAV will, in a safe way, descend toward a platform on top of the UGV. Feasibility is validated by demonstration using a system consisting of a Penguin BE UAV and a semi-autonomous car. Performing this demonstration successfully is the main goal of the project.

The main focuses of this thesis are an analysis of the cooperation between the vehicles during the landing phase, as well as work with the overall experimental setup. The proposed control strategies are evaluated analytically, using simulation as well as with flight tests. The objective is to derive a controller with a satisfactory performance that will make the landing possible, and to perform an autonomous landing with it.

1.4 Contributions

The main contribution is to verify that an autonomous landing of a fixed-wing UAV on top of a UGV is possible. Verifying this is the first step toward developing alternative UAV landing techniques further. By performing an autonomous landing the concept is taken into reality for the first time. When doing this, strengths and weaknesses of this approach to landing can be identified. This knowledge is valuable for future developments.

This thesis contains a theoretical analysis of the problem, including a derivation of a control strategy for the cooperative landing, as well as a practical part where this strategy is implemented. Through analysis, simulations, hardware-in-the-loop testing, and flight experiments we are able to analyze the performance of the suggested controller in several frameworks, all providing different insights

into the problem.

The analysis together with the simulations and the hardware-in-the-loop testing increases our understanding of the system and helps us in finding a suitable initial setup for a controller. By implementing the controller in a physical model, it is possible to directly observe the way that the system handles combinations of time delays, measurement uncertainties and disturbances. With this we gain a deeper understanding of the interactions of the closed loop system with the environment. It is then possible to make suggestions for future developments according to any problems that were observed in flight experiments.

1.5 Outline of thesis

Chapter 2 begins with a summary of the theory of fixed-wing aircraft flight dynamics, including a description of some useful reference frames and a linear version of the system dynamics.

Chapter 3 contains descriptions of the different parts of the system used in the flight experiments, including the UAV, the UGV, the landing platform, and the communication setup. Chapter **??** explains general flight control concepts, including a description of a Total Energy Control System (TECS). A stabilizing controller along with an autopilot for following specified paths for the Penguin UAV is thereafter derived using classical control approaches.

In Chapter 5, the experimental procedure is explained in more detail. Chapter 6 deals with the problem of controlling the vehicles to align themselves in the x and y-direction, and Chapter 7 treats some different descent control approaches.

The results of the flight testing are discussed in Chapter 8. These results form the basis for the suggested improvements presented in Chapter 9. Finally, Chapter 10 offers an analysis of the project, with conclusions and some suggestions for future work.

Chapter 2

Flight Dynamics

Two basic concepts of flight are *lift* and *drag*, and the ability to control these forces is essential to flying. This chapter provides a summary of the dynamics of flight and flight control in order to give an understanding of how lift and drag can be controlled. Starting with a short summary of some general flight theory and the dynamical equations of flight, the chapter continues with explaining how the flight equations can be linearized. The chapter ends with a derivation of a state space formulation of the flight dynamics of the Penguin BE UAV.

2.1 Theory of Flight

Lift is generated on an aircraft surface moving through air as it creates a pressure differential between the upper and lower surfaces. If the surface moves fast enough and generates enough lift, it can overcome its weight and achieve flight. Drag is similarly created by the movement of the aircraft through the air, and can be described as air resistance, opposing the forward movement generated by the thrust (Figure 2.1).

The lift and drag forces of an aircraft are dependent on the shape of all aircraft surfaces, the flight velocity, and of the air density. A fixed-wing aircraft uses airfoil surfaces on the wings and on the tail to create enough lift for flight. The shape and size of the wings together with the fuselage and empennage will

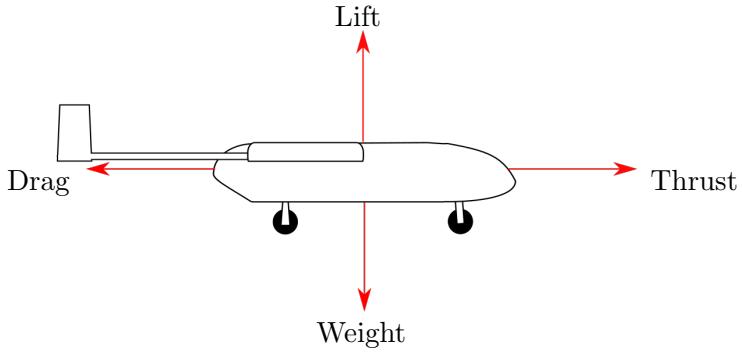


Figure 2.1: Four fundamental forces acting on a flying aircraft are the lift, thrust, drag and weight.

influence the lift-to-drag ratio and thereby the flight dynamics. The aircraft geometry, together with its weight and balance, will also define the moments acting on the aircraft and thus determine important flight properties related to stability and control. The properties of a perfect airfoil of infinite length can be described in terms of three dimensionless coefficients - lift, drag and pitching moment. When considering a real aircraft the surfaces are of finite length and so the airfoil model is only an approximation. In addition to this, the flow around neighboring surfaces create interference effects, resulting in complex flows which are difficult to describe analytically. This, combined with the fact that the details of how flight is achieved is still not completely understood today, makes flight theory largely dependent on either experimental measurements in wind tunnels, flight testing or numerical methods such as computational fluid dynamics (CFD) or simpler methods like VLM (vortex lattice methods). For small UAVs the latter is typically sufficient for modeling, although testing on real models will in general give more precise results.

The way that the physical structure of the aircraft affects the flight dynamics is typically described with a set of aerodynamic derivative coefficients, which can be attained experimentally or from simulations. These coefficients are then used to describe the aircraft dynamics as a set of nonlinear equations.

2.2 Reference Frames

There are four main reference frames in use when describing the dynamics of an aircraft. The angles between these reference frames are the definitions of some important angles for describing the aircraft motion and calculating the forces and moments acting on it.

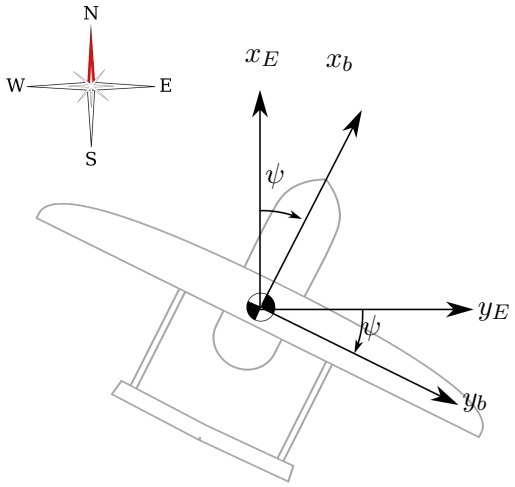
An geodetic reference frame is used to relate the aircraft motion to the earth. It typically has the X-direction to the north, the Y-direction to the east, and the Z-direction pointing downward.

The body-fixed reference frame has its origin in the center of gravity (CG) of the aircraft. The X-direction is pointing forward in the aircraft's vertical symmetry plane and the Z-direction is pointing downward with respect to the aircraft. How the roll, pitch and yaw angles are defined is shown in Figure 2.2 (with each subfigure showing the case where it is assumed that both other angles are zero).

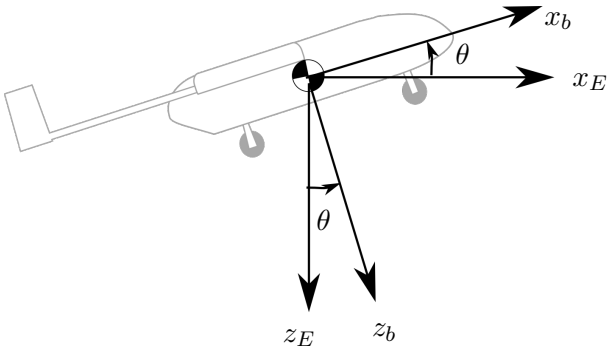
The aerodynamic reference frame has the X-axis pointing in the direction of the relative wind. From this reference frame the two aerodynamic angles can be defined - angle of attack α and sideslip β, two critical angles affecting the aerodynamic forces and moments acting on the aircraft. The angle of attack has its main effect on the longitudinal forces on the aircraft, while the sideslip angle is more related to the lateral dynamics. These angles can be seen in Figure 2.3.

Finally there is the path-fixed reference frame, also called flight-path reference frame. The x-axis points in the direction of movement in this reference frame. The angle that this path makes around the geodetic z-axis is called the lateral path angle and is denoted χ. The angle that the path makes with the

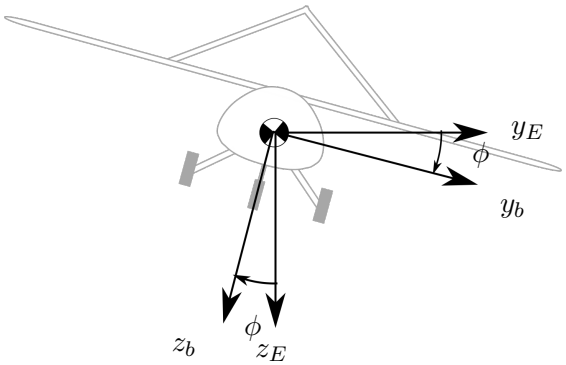
geodetic reference frame around the y-axis is the vertical path angle γ. These path angles are important for navigational control.



(a) Yaw angle

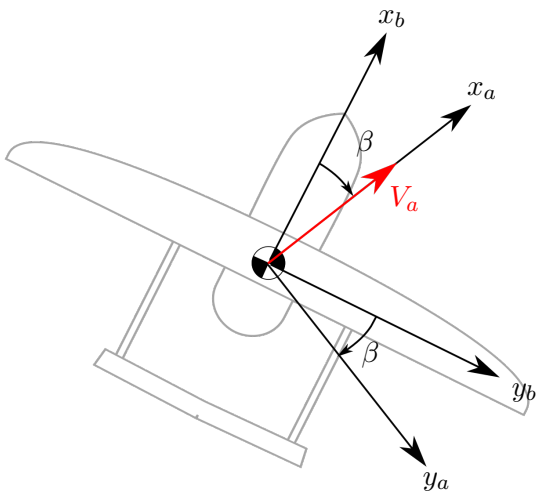


(b) Pitch angle

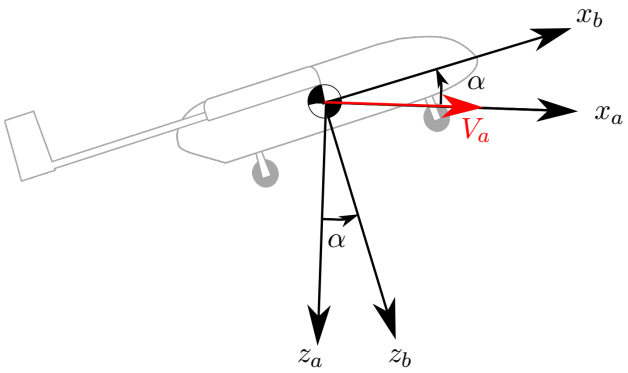


(c) Roll

Figure 2.2: By turning the aircraft around any of its principal axes, roll, yaw and pitch motion is induced. The body frame is denoted with subscript b, and the geodetic with subscript E. All pictures show rotation around one axis at a time.



(a) Sideslip angle



(b) Angle of attack

Figure 2.3: Transformation from the body-fixed reference system (subscript b) to aerodynamic reference frame (subscript a). The aerodynamic angles α and β strongly affects the dynamic properties of the aircraft. Both pictures show rotation around a single axis.

2.3 Equations of Motion

The aircraft is affected by the gravitational force, and aerodynamic forces and moments generated by the movement through the air. Similarly as for an airfoil, these forces are described in terms of lift, drag, and pitching moment. In addition to these there is also sideforce, rolling moment and yawing moment when considering the complete three dimensional structure. These coefficients are

Drag	$C_D = f_D(\alpha, \beta, f, M, ...)$	Roll	$C_p = f_p(\alpha, \beta, f, M, ...)$
Lift	$C_L = f_L(\alpha, \beta, f, M, ...)$	Pitch	$C_q = f_q(\alpha, \beta, f, M, ...)$
Sideforce	$C_C = f_C(\alpha, \beta, f, M, ...)$	Yaw	$C_r = f_r(\alpha, \beta, f, M, ...)$

where the coefficients are nonlinear functions depending on the aerodynamic angles, Mach number, angular velocities, and control surfaces. Approximations of these equations can be found by fitting wind tunnel data or data from numerical methods to polynomials of these variables. The resulting forces and moments will be of the form

$$F^b_{aero} = \begin{bmatrix} X^b \\ Y^b \\ Z^b \end{bmatrix} = \begin{bmatrix} -\bar{q}SC_D \\ \bar{q}SC_C \\ -\bar{q}SC_L \end{bmatrix} \qquad M^b_{aero} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \bar{q}SbC_p \\ \bar{q}S\bar{c}C_q \\ \bar{q}SbC_r \end{bmatrix} \tag{2.1}$$

where $\bar{q} = \frac{1}{2}\rho v^2$ is the dynamic pressure, S is the total wing area, and b is the wing span. The mean aerodynamic chord $\bar{c}$ is the mean distance between the leading and trailing edge of the wing. The total force on the aircraft in the aircraft body frame is therefore given as

$$F^b = F^b_g + F^b_{aero} = mg \begin{bmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\sin\theta \end{bmatrix} + \begin{bmatrix} X^b \\ Y^b \\ Z^b \end{bmatrix}. \tag{2.2}$$

These forces and moments give rise to velocities and angular velocities, that in the body-fixed reference system are denoted by

$$V^b = \begin{bmatrix} u & v & w \end{bmatrix}^T \tag{2.3}$$

$$\Omega^b = \begin{bmatrix} p & q & r \end{bmatrix}^T \tag{2.4}$$

Combining Equations 2.2-2.4 leads to

$$\begin{aligned} mg(\dot{u} + qw - rv) &= X^b - \sin\theta \\ mg(\dot{v} + ru - pw) &= Y^b + \cos\theta\sin\phi) \\ mg(\dot{w} + pv - qu) &= Z^b + \cos\phi\sin\theta \end{aligned} \tag{2.5}$$

$$\begin{aligned} I_x\dot{p} + (Iz - Iy)qr - I_{xz}(pq + \dot{r}) &= L \\ I_y\dot{q} + (Ix - Iz)pr - I_{xz}(p^2 - r^2) &= M \\ I_z\dot{r} + (Iy - Ix)pq - I_{xz}(qr - \dot{p}) &= N \end{aligned} \tag{2.6}$$

where I_{ij} indicates instances of the inertia tensor. Due to the symmetry of the (x, z)-plane, we get that the inertia I_{xy} and I_{yz} are zero.

2.4 Flight Control Surfaces

The dynamic properties of an aircraft are strongly related to its shape. During flight, the attitude and flight path of the aircraft can be affected by deflecting the control surfaces, changing the forces and moments acting on it. The type of control surfaces and how they affect the aircraft will depend on the type of aircraft. Three main control surfaces of a standard fixed wing aircraft are ailerons, rudder, and elevator. The main directional control surfaces are the ailerons. There is one aileron surface on the trailing edge of each wing of the aircraft. A positive aileron input will turn the aileron control surfaces so that the trailing edge of the right aileron is moved downward, and the left aileron is moved upward. This creates a negative rolling moment, and with this the lift vector is tilted, creating a side force and changing the path angle.

Rudder control is also related to the lateral motion of the aircraft, creating a yawing moment. Although it would be possible for some aircraft to use rudders to control the lateral path of the aircraft, this is much slower than using the ailerons. Instead, rudders are used together with the ailerons to assist in coordinating the turn, making the nose point in the direction of flight. The rudder is usually placed on the trailing edge of the vertical stabilizer. A positive rudder input will turn the rudder to the left, changing the heading and sideslip angle of the aircraft.

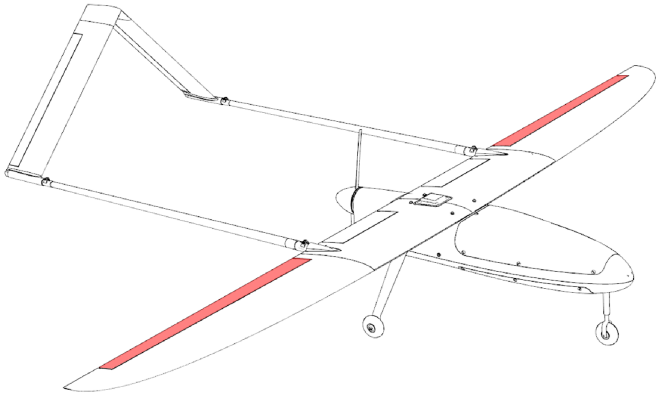
Elevators are placed in pairs on the trailing edge of the tail, and are moved together to create a pitching moment, tilting the lift vector to change the vertical force on the aircraft. A positive elevator input tilts the elevators downward, creating a negative pitching moment. The lift vector then tilts, affecting the vertical path. Altitude changes are strongly coupled with velocity changes in aircraft, so an elevator input will in general not only affect the pitch angle, but at the same time also the velocity. Also, a thrust command will not only change the velocity of the aircraft but also the generated lift force and thus altitude.

On the UAV used in this project, the rudder and elevator are combined into one type of control surface called ruddervator. The control surfaces for the Penguin BE are illustrated in Figure 2.4.

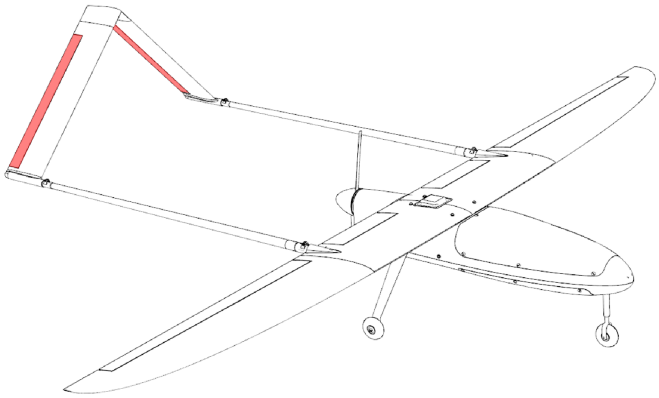
2.5 Automatic Flight Control

Control of aircraft has become increasingly sophisticated since the early history of flight. At the start of manned flight the control was completely mechanical, making the pilot directly move the control surfaces with the use of cables and pulleys. This required constant attention from the pilot in order to fly the plane in a safe way. With the evolving technologies of the 20th century came the possibility to do this electrically, using so called “fly-by-wire” techniques. This made it easier to use feedback control to change the responses to the different inputs.

In automatic control of aircraft, there is a distinction made between control systems depending on their task. Stability Augmentation System (SAS) is used to damp eigenmodes of the system, in particular the short period and the dutch



(a) Aileron



(b) Ruddervator

Figure 2.4: Control surfaces of the Penguin BE

roll modes. A Control Augmentation System (CAS) is used to improve the control response of certain control inputs. Finally, an autopilot is a control system that fully automates the control of the desired path, velocity, altitude etc.

In manual flight mode where the inputs are directly given by a pilot moving the control surfaces, feedback control can be used to improve the handling qualities of the aircraft by shaping the closed loop response. This makes the aircraft easier to fly by damping oscillatory terms and stabilizing possibly unstable modes. In an unmanned aircraft on the other hand, inputs are given in terms of desired path, altitude and velocity. We are not directly interested in the dynamical response from the control surfaces, but rather how well the aircraft can follow given reference values. Still, looking at and compensating for the stability and damping of these modes and trying to get a good control surface deflection to state response is a good staring approach for a completely automatic controller.

Flight control is to a large extent based on cascade control, where the inner loops are successively closed to attain a desired performance. This requires both system knowledge and experience from the control designer when choosing the structure, and so making or changing an existing controller can become a large and time consuming effort. With increasingly complex flight systems, modern control techniques are becoming more popular, with methods like eigenstructure assignment, LQR, and robust control being among some of the techniques that have been used in aircraft control systems.

2.6 Linearized Equations of Motion

It is often more convenient to consider the linear dynamics around the point of interest rather than the entire nonlinear model when doing control design and analysis. The linearization is performed in steady-state flight, where the sums of the forces and moments in all directions equals zero. We thus want the following equalities to hold

$$\left.\begin{array}{ll} \text{Angular velocities} & p = q = r = 0 \\ \text{Sideslip} & \beta = 0 \\ \text{Angle of attack derivative} & \dot{\alpha} = 0 \\ \text{Airspeed derivative} & \dot{V}_a = 0 \end{array}\right\} \tag{2.7}$$

The linearized equations can then be attained from these conditions either with simulation or from analytically linearizing the non-linear equations of motion.

The linearization results in a set of differential equations, which by inspection can be seen to be decoupled into two separate sets - the symmetrical and the asymmetrical forces and moments.

$$\begin{bmatrix} \dot{x}_{long} \\ \dot{x}_{lat} \end{bmatrix} = \begin{bmatrix} A_{long} & 0 \\ 0 & A_{lat} \end{bmatrix} \begin{bmatrix} x_{long} \\ x_{lat} \end{bmatrix} + \begin{bmatrix} B_{long} & 0 \\ 0 & B_{lat} \end{bmatrix} \begin{bmatrix} u_{long} \\ u_{lat} \end{bmatrix} \tag{2.8}$$

The first part corresponds to the longitudinal (symmetrical) dynamics, which describes the aircraft motion in the (X_b, Z_b)-plane. It has three degrees of free-

dom corresponding to pitch, longitudinal motion and vertical motion. The second part is the lateral (asymmetrical) dynamics, describing the aircraft motion around the Z_b-axis, which consists of roll, yaw and lateral motion.

A linear model of the Penguin BE UAV was derived using MATLAB and Simulink. The Simulink model is described in Section 3.1.1. The *trim* function of MATLAB was used to find the equilibrium state under the constraints in (2.7) and the desired airspeed 21 m/s. After an equilibrium state was found, a linear state space representation was found using the *linmod* function. This could then be decoupled into two systems:

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -0.28 & 1.9 & -21 & 9.8 & 0 \\ -0.40 & -14 & 2.5 & 0 & 0 \\ 1.3 & -2.1 & -1.2 & 0 & 0 \\ 0 & 1 & 0.08 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} 0.69 & 4.5 \\ -131 & -3.9 \\ -19 & -23 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -0.10 & 0.39 & -1.4 & -9.8 & 0.006 \\ -0.64 & -3.6 & 22 & -0.6 & 0 \\ 0.19 & -2.8 & -5.6 & 0 & -0.001 \\ 0 & 0 & 1 & 0 & 0 \\ 21 & 1.2 & 0 & 0 & -2.6 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0.38 & 0 \\ -7.3 & 0 \\ -65 & 0 \\ 0 & 0 \\ 0 & 2027 \end{bmatrix} \begin{bmatrix} \eta \\ f \end{bmatrix} \tag{2.10}$$

for the lateral and longitudinal dynamics respectively. The inputs to the lateral dynamics are $u_{lat} = \begin{bmatrix} \zeta & \xi \end{bmatrix}^T$ with rudder deflection ζ and aileron deflection ξ. The inputs to the longitudinal system $u_{long} = \begin{bmatrix} \eta & f \end{bmatrix}^T$ are the elevator η and throttle f.

In Chapter 4, the stability and dynamical response of these two systems are analyzed and an autopilot controller is derived.

Chapter 3

System Description

This chapter provides an overview of the system setup for the flight experiments, with the main parts being the Penguin BE UAV and a ground vehicle consisting of a car modified with a driver interface and a landing platform. The communication and positioning are described briefly.

3.1 Aerial Vehicle

The UAV used in the flight experiments is a Penguin BE from UAV Factory, a 3.3 m wide and 2.3 m long fixed-wing UAV with an inverted V-tail. It runs with an electric motor and has aileron, ruddervator and flaps control surfaces. This UAV is commercially available, and has at DLR been extended with a flight control system and different sensors including GPS, IMU and pitot tube for flow measurements.

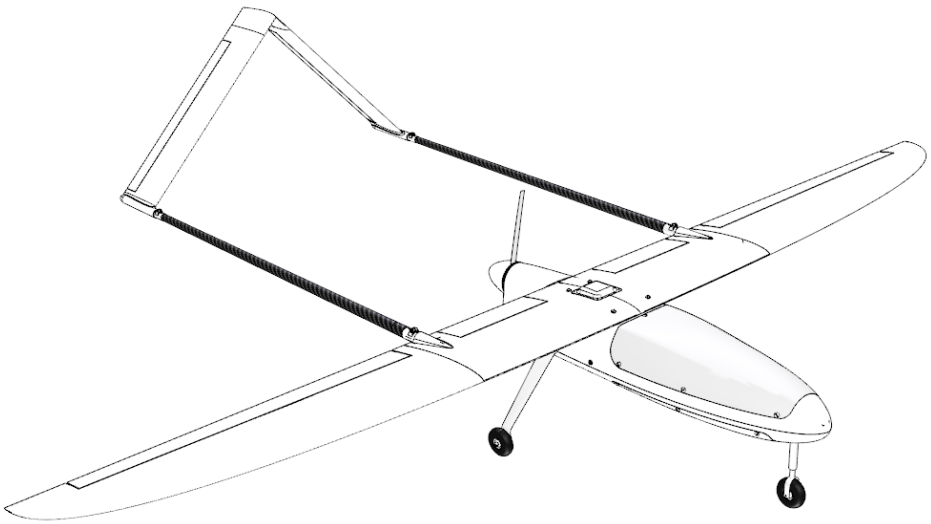


Figure 3.1: Penguin BE

3.1.1 Simulink Model

A 6 DOF model of the aircraft and its controller structure have previously been built in Simulink using the Aerosim blockset[1]. The aerodynamic properties of the UAV were found using the potential flow solver AVL[2]. A thorough description of the model can be found in [16]. In addition to the aerodynamics, the model also contains atmospheric dynamics and a specific propulsion model. The Aerosim blocks support wind disturbance inputs and turbulence to be used in simulation.

3.2 Ground Vehicle

The intention for the final application is that both the UAV and the UGV will be fully autonomous vehicles. The low availability of a large enough autonomous ground vehicle motivated the choice of instead using a semi-autonomous vehicle for these initial proof-of-concept tests. In this solution a human driver is executing control commands provided by the ground vehicle controller through a graphical interface. Having a human actuator in the loop introduces several possible challenges. First of all there will be a natural time delay corresponding to the reaction time of a human. Secondly, it is difficult to make the human reliably follow these commands without adding extra control himself. Another possible complication is that a human could unconsciously take other things into consideration when applying control, such as the sound of the UAV motor or intuition of how the experiment should go. It might also be more difficult to have advanced control settings such as simultaneously instruct the driver to accelerate and steer, since the driver will most likely want to look at the road as well.

For this project it was early on established through conceptual simulation that a simple car dynamic would be sufficient for the landing. By limiting the motion to be relatively straight and the changes to be slow, the human driver can more easily follow the control commands. A short review of the performance of the human as an actuator can be found in Appendix B. For the scope of this project, the simplicity of this solution outweighed any benefits from using a more complex system.

The vehicle is an Audi A6 equipped with a monitor giving the driver directions in the form of two bars on a screen (Figure 3.2). As the vertical bar moves, the driver turns the steering wheel to change the direction and course of the car. The horizontal bar controls the acceleration. An upward movement indicates to the driver that he should accelerate and a downward movement indicates that he should decelerate. The goal of the driver is to keep the two bars in the center at all times.

3.3 Landing Platform

A simple structure had been designed to catch the UAV in this project (Figure 3.3). The structure consists of hollow aluminum bars with a net spanned in

[1] `http://www.u-dynamics.com/aerosim/`

[2] `http://web.mit.edu/drela/Public/web/avl/`

Figure 3.2: The driver of the ground vehicle receives directions in the form of two bars to follow. The vertical bar commands steering wheel angle rate and the horizontal commands the throttle/gas setting of the car.

between them. The net tension was based on the expected impact of the UAV landing and was tested by letting the UAV fall onto the net.

The platform is attached on top of the car for the landing, and the final landing is carried through with the net simply catching the UAV and holding it locked in the net with a barb like mechanism on the wheel of the UAV. The platform is 4 meters long and 5 meters wide, which gives the Penguin approximately 1 m extra space in each direction if it is placed in the middle. To have some margins, it is assumed that the Penguin can land at most 0.8 m away from the center to any side.

Figure 3.3: The landing platform consists of a net stretched in a metal frame.

3.4 Positioning and Communication

Having accurate knowledge of the position is essential for controlling the vehicles. Both vehicles are equipped with NovAtel Real Time Kinematic (RTK) GPS, a

type of differential GPS receiver providing real-time positioning information with the help of an additional base station, returning an accuracy in the centimeter range. The Penguin is also equipped with a camera for direct relative state estimation using optimal marker tracking methods.

Both vehicles are given access to all information from the other vehicle, but with a varying time delay. The communication between the vehicles is done at a rate of 10 Hz, and from previous flight tests the time delays have been estimated to be between 0.05 and 0.2 s, with a 0.05 s wide time resolution on the timestamps. In the final phase of the landing, the vehicles will have a velocity of around 20 m/s. Between the sampling of data and the usage in the controller the position might move as much as four meters, and so the position estimation is required. The real x position is estimated as

$$\begin{aligned} x_{rel}^{UGV}(t) &= x_{UGV}(t) - (x_{UAV}(t-\Delta t) + \Delta t \cdot u_{UAV}(t-\Delta t)) \\ x_{rel}^{UAV}(t) &= (x_{UGV}(t-\Delta t) + \Delta t \cdot u_{UGV}(t-\Delta t)) - x_{UAV}(t) \end{aligned} \tag{3.1}$$

and correspondingly for the y and z-directions. Here u represents the velocity in the forward direction. In this prediction model, a constant velocity in between the samples is assumed. This is a simplification, in particular during the phase when the UGV accelerates from 0 m/s to 20 m/s in around 10 seconds. An estimation of how large the error can be can be found by approximating the acceleration to be a =2 m/s^2. The velocity and position at this time is given by

$$\begin{aligned} v(t_0 + \Delta t) &= v(t_0) + a\Delta t \\ \Rightarrow x(t_0 + \Delta t) &= x(t_0) + v(t_0)\Delta t + \Delta t^2 \end{aligned}$$

while the estimated will be $x(t_0) + v(t_0)\Delta t$. The error in position when using Equation (3.1) is therefore at most $\Delta t^2 = 0.01$ m with$a$ =2 m/s^2 and sampling rate of 10 Hz. After this initial phase, as more precision is needed in the controller, the relative acceleration will be smaller than this and so the error will be smaller. Considering that the precision of the GPS measurement is of the same size order, and that the required accuracy for landing is 0.8 m, this simple prediction of the relative positions is accurate enough.

Chapter 4

Flight Control

The dynamical behavior of an aircraft is often described and analyzed using linearized models and classical control theory methods. The positions of the system poles are related to the characteristic flight behavior of the aircraft. This can be tuned with the use of feedback control such as stability and control augmentation systems, or autopilots for completely autonomous flight. This chapter presents the dynamical modes of the Penguin BE as calculated from the Simulink model. Thereafter the autopilot system is explained and appropriate gains are derived.

4.1 Aircraft Modes

By applying certain inputs to the aircraft it is possible to excite the different dynamical modes. The longitudinal dynamics are characterized by the phugoid mode and the short-period mode, and the way that these poles are placed is related to the dynamic response of a longitudinal control input. The three main modes for the lateral dynamics are the dutch roll mode, the roll subsidence mode and the spiral mode. These modes are typically exited by using aileron or rudder inputs. The dutch roll is linked to a combined oscillation in yaw and roll. The two remaining modes, which typically have quite long time constants, are exponential and are related to a damping of the roll rate and to a spiral motion of the aircraft.

Three of these modes; the dutch roll, the roll subsidence and the short period modes, are primary related to a rotational motion of the aircraft. Their time constants are short compared to the other three modes and they depend on the inertia properties of the aircraft. The two remaining modes affect the aircraft path, and tend to have much larger time constants. As opposed to the fast rotational modes, the slower ones are typically not difficult to control for a human, but it would be tiring to continuously do so for longer flights.

The poles of the linearized lateral dynamics of the Penguin BE UAV are found

from the eigenvalues of the lateral state space model as derived in Section 2.6

-13.680 + 0.000i	Roll subsidence
-0.952 ± 5.387i	Dutch roll
0.103 + 0.000i	Spiral mode

The eigenvalues corresponding to the dutch roll are placed at $-0.9629 \pm 5.2046i$. This corresponds to a damping of 0.1819 and a period of 1.2072 s. The roll subsidence eigenvalue is at -13.4672 which gives it a time constant of 0.0743 s. The spiral mode eigenvalue is 0.1047 and it is unstable. Having an unstable spiral mode is not uncommon, and it is usually not difficult to stabilize due to the relatively long time constant. The transfer function from aileron to roll is

$$\frac{P(s)}{\Xi(s)} = \frac{(s-0.04)(s+0.95+5.1i)(s+0.95-5.1i)}{(s+0.96+5.2i)(s+0.96-5.2i)(s-0.10)(s+13)}$$

The dutch roll is close to the complex zero pair, and so for the given trim point its effect will almost be canceled. Figure 4.1 shows the roll response to an aileron doublet input.

The poles of the longitudinal model as described in Section 2.6 are equal to

-4.596 ± 7.796i	Short-period
-0.031 ± 0.535i	Phugoid
-2.647 + 0.000i	Motor dynamics

The short period mode has a damping of 0.5094 and a period of 0.9716 s. The phugoid time period is 9.6769 seconds long and its damping is 0.0491. The result of an elevator input on the pitch can be seen in Figure 4.2. The phugoid mode is clearly visible, whilst the short-period mode as expected cannot be observed at all.

4.2 Lateral Stability Augmentation and Autopilot

The Penguin BE is equipped with a vertical stabilizer, generating restoring moments and forces to act stabilizing around the yaw axis. A sufficient turning performance was found in flight experiments using ailerons only, and so no rudder control was required.

The aircraft needs stabilization around the roll axis. First a roll rate feedback is added. Then a PI feedback of the roll angle is added to enable active control of the roll angle. The MATLAB *sisotool* was used for this purpose. With gains $K_\phi = 3.35$ and $K_{i\phi} = 0.5$, the closed loop system poles are

-31.215 + 0.000i	Roll subsidence
-1.035 ± 4.557i	Dutch roll
-2.027 + 0.000i	Spiral mode
-0.170 + 0.000i	ϕ feedback integrator

The spiral mode is now stabilized and has a time constant of 0.4934 seconds. The dutch roll has a slightly longer time period and is slightly more damped than

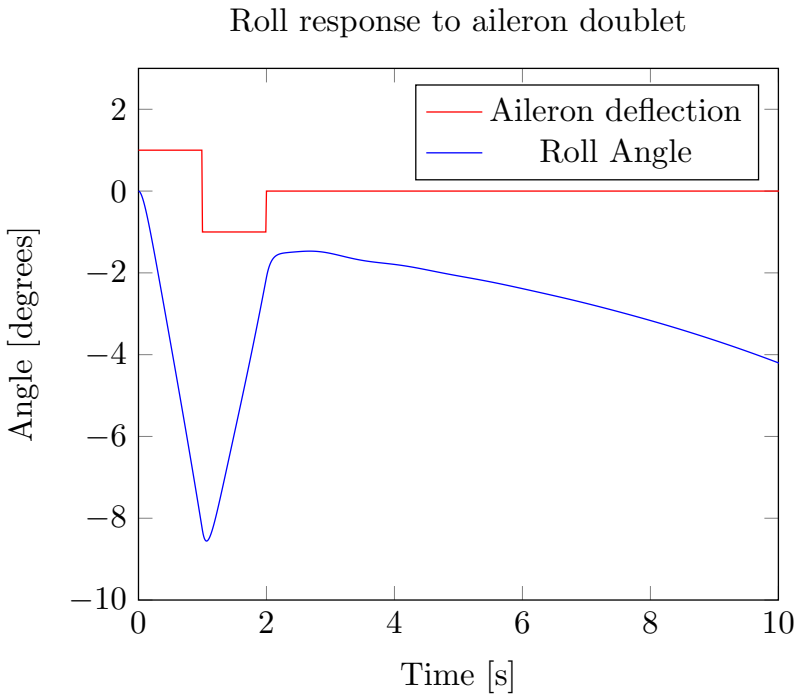


Figure 4.1: The roll response to a aileron doublet. The rotational modes are barely visible, instead the unstable spiral mode is dominating the response.

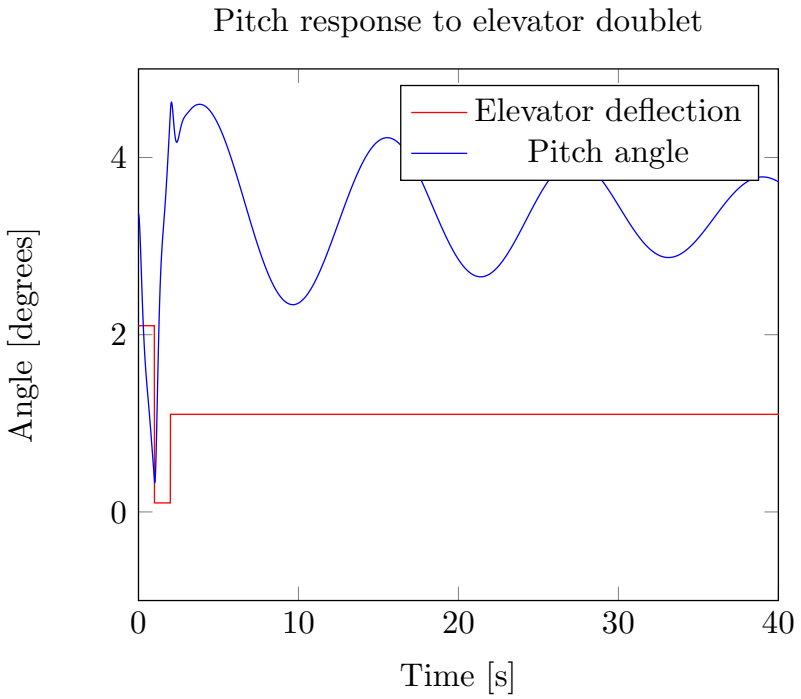


Figure 4.2: The pitch response to an elevator doublet. The low damping of the phugoid mode is clearly visible.

before (1.3787 seconds, 0.2215). The closed loop response to a doublet reference is shown in Figure 4.3.

The path angle was chosen as the controlled variable for the lateral autopilot. This is implemented by closing the outermost loop of the controller with a PI feedback of the path angle χ. The final lateral controller has the structure shown in Figure 4.5. The response from the reference signal χ_{des} to the course angle χ can be seen in Figure 4.4.

4.3 Longitudinal Stability Augmentation and Autopilot

The underdamped phugoid mode of the longitudinal dynamics is handled by adding a pitch rate feedback. This moves the phugoid poles to -0.0523 ± 0.2381i.

The ability of the autopilot to precisely control altitude and velocity is essential for the landing maneuver. During the descent of the UAV, it is necessary for the UAV to control both the sink rate and the position. At the same time it needs to keep a desired velocity suitable for landing. The aircraft will at a given moment contain some kinetic and some potential energy stored in the current altitude and speed. This makes the control of these two variables naturally linked by the way that the total energy of the system is added and distributed, since the available control inputs cannot control altitude and speed separately. This connection between speed and altitude control causes some challenge in the simultaneous and independent control of these variables.

4.3.1 Total Energy Control System

Traditional flight control applies a SISO approach to design the autopilot controllers for velocity and altitude separately. Throttle is then commanded depending on the difference in desired speed and actual speed, while the desired glide slope is commanding the elevator deflection. Such a controller will suffer from poor damping and overshooting when trying to perform certain coupled changes in altitude and speed. As an example of this one might consider an autopilot trying to change the glide slope and at the same time keep a constant velocity. As the flight path angle is changed, the aircraft gains speed. The controller now lowers the speed by changing the thrust, resulting in a reduced lift force and thus a change of the glide slope. The controller must therefore go through extensive testing in order to reach a design that minimizes these effects for the desired flight slope.

The idea behind using a Total Energy Control System (TECS) [17] is to utilize the energy content distribution directly in the design of the controller. TECS was developed by Boeing and NASA in the 80s as a method of coupling the elevator and throttle inputs in the flight controller, and it uses a control law derived from the underlying physical properties of the system.

The basic idea behind TECS is to use the fact that the only way that energy content is added to the aircraft is from the propulsion system, which is regulated by the throttle. This additional energy is then either used for increasing the

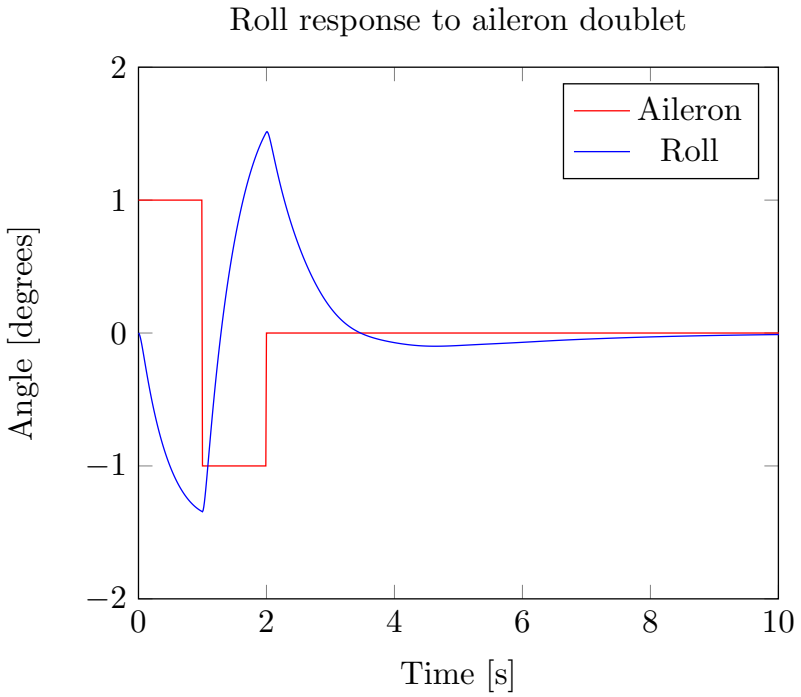


Figure 4.3: The roll response to a aileron doublet after after lateral stability augmentation and active roll attitude control. Compare to Figure 4.1

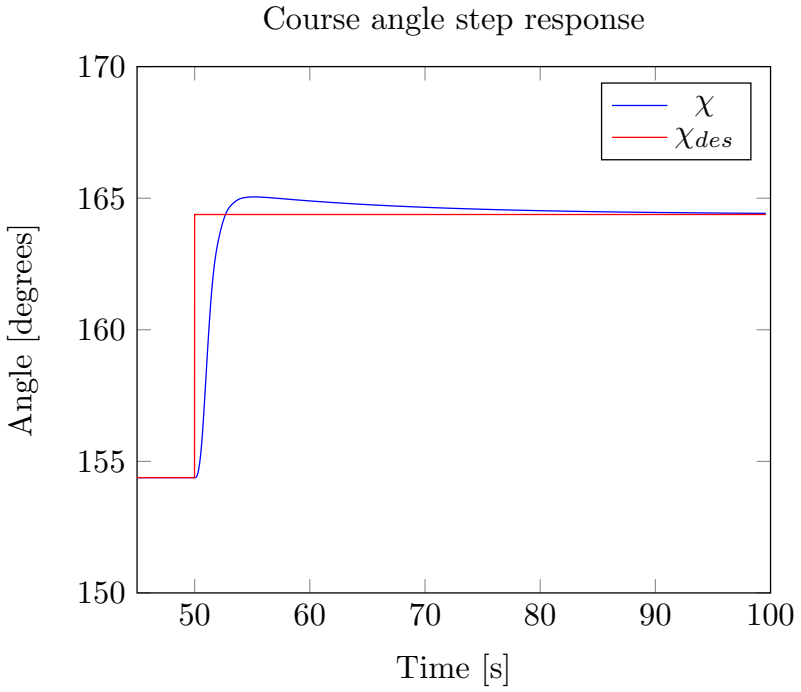


Figure 4.4: Course angle χ response to a step command.

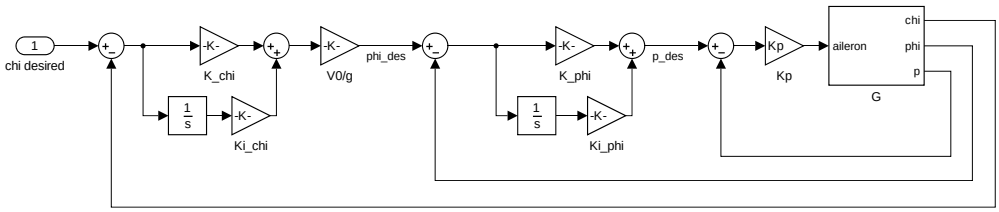


Figure 4.5: The lateral controller structure. The path angle χ is controlled with aileron inputs via roll ϕ and roll rate p feedbacks.

kinetic or the potential energy. The part of this additional energy that is taken by the potential energy is highly dependent on the elevator deflection. With this reasoning, the throttle output should equal the desired total energy increase, and then the energy will be distributed between potential and kinetic using the elevator. Let m be the total mass, h the altitude and v the velocity of the aircraft. The energy rate error is then given by

$$\dot{E}_e = \left(\frac{\dot{V}_{des}}{g} - \frac{\dot{V}}{g} \right) + (\gamma_{des} - \gamma) \tag{4.1}$$

The energy distribution rate error is given as

$$\dot{L}_e = \left(\frac{\dot{V}_{des}}{g} - \frac{\dot{V}}{g} \right) - (\gamma_{des} - \gamma) \tag{4.2}$$

The energy rate error and the energy distribution errors are fed into two separate inner controllers, and the output of these controllers are the system input signals of thrust command and elevator command.

Due to limitations in thrust, the total energy content of the aircraft cannot change unconstrained. When this limit is reached, the aircraft is put into either *path priority mode* or *speed priority mode*. In the former, acceleration command $\dot{v}_{des}$ is interrupted and γ_{des} is kept in its previous control, and in the latter the opposite is done, leaving the flight path in open loop control.

A detailed description of the implementation and testing of TECS into the Penguin UAV can be found in [16].

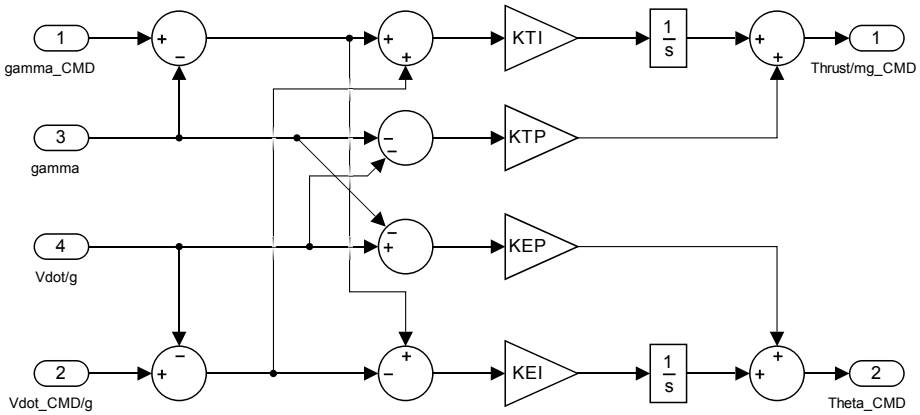


Figure 4.6: TECS takes the total energy need into account and distributes it between potential energy and kinetic energy with the use of the elevator.

Chapter 5

Experimental Procedure

During each landing attempt the aircraft goes through several different flight modes. The relation between the starring orders of these modes is shown in Figure 5.1. Prior to the start of each landing, the UAV is placed in a satisfactory starting position with the use of manual flight and waypoint navigation. If it would happen that the UAV during the landing goes into an undesirable state, then it will go into go-around mode and interrupt the landing.

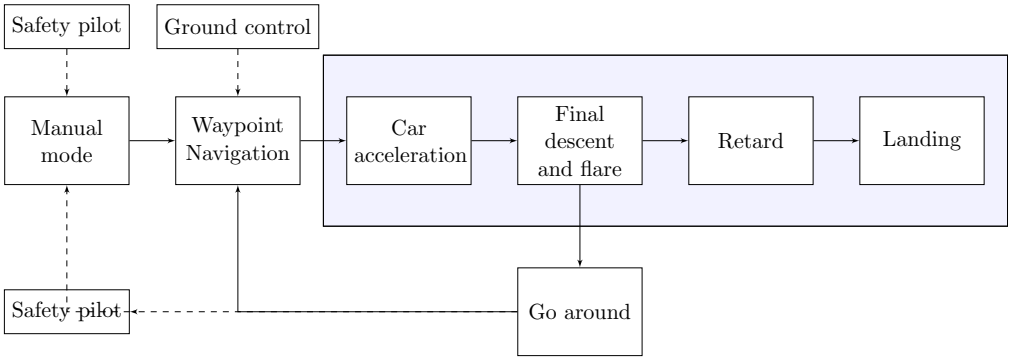


Figure 5.1: Flowchart of the landing process. In the blue box are the parts directly related to the landing.

5.1 Pre-Landing Flight

The UAV is manually started from the runway by the safety pilot. When it has reached a certain altitude the UAV is taken into the autonomous navigation mode, where the open source software QGroundControl[1] is used (Figure 5.2) to define waypoints for the UAV to follow. The UAV tries to reach these pre-specified waypoints in a specific order. As the UAV approaches the UGV (between waypoint 2 and waypoint 3 in the figure), an operator initiates the start of the landing maneuver from the ground station.

[1] `http://qgroundcontrol.com`

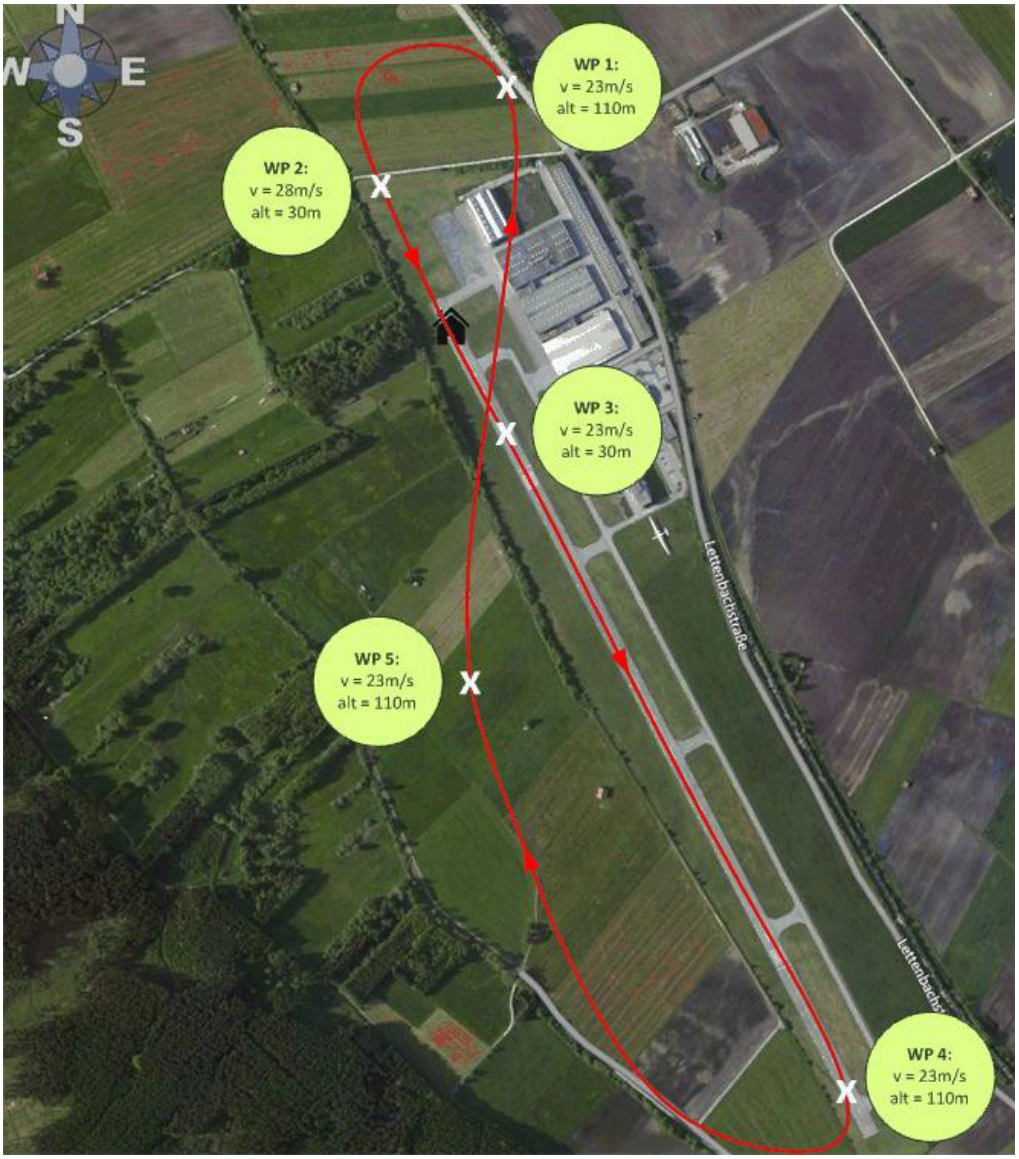


Figure 5.2: After takeoff, the pilot puts the UAV into waypoint mode where the UAV autonomously tracks some prespecified waypoints. This takes the UAV into a position from where the landing maneuver can be initiated.

The initial conditions of the autonomous landing will be dependent on the coordinates chosen for the waypoint navigation, the external conditions of the flight and on when the landing is initiated by the operator. Ideally the initial conditions for the landing maneuver would be the same to allow for repeatable landing maneuvers. Inaccuracies due to manual waypoint mode could be decreased using methods such as wind estimation, which automatically chooses a waypoint pattern according to current wind conditions, or by adding adaptive techniques to the UAV autopilot to make the controller less dependent on external factors.

Since the safety pilot must be able to take over the control over the UAV at all times, the waypoints cannot be too far from the runway start without having a sufficiently high altitude. Therefore, all waypoints that are far away from the safety pilot have an altitude of 110 m. To make it possible for the pilot to see the UAV throughout the landing, the pilot sits in the backseat of a convertible car, which follows the UAV as it flies down the runway. In between waypoint 1 and 2, the UAV needs to loose a large part of its altitude in order to start the landing phase. As this is done a higher velocity (28 m/s) is commanded in order to facilitate a steeper descent. The velocity is thereafter commanded back to the initial 23 m/s.

5.2 Acceleration Phase

As the UAV is flying in waypoint navigation mode, the UGV stays still at the start of the runway waiting for a starting command, after which the UGV starts to accelerate to reach the same speed as the UAV. The starting command is given when the UAV reaches a certain point behind the starting position of the UGV. Ideally, this point is chosen as to make the vehicles reach the same velocity and position at the same time.

To find a suitable point for the UGV to start accelerating, an estimation of when the vehicles will align is running in the UGV computer. The UGV acceleration command is calculated under the assumption that the UAV maintains its speed, while the UGV has a constant acceleration

$$\begin{aligned} V_{UAV} &= V_A \\ V_{UGV} &= a \cdot t \end{aligned}$$

Assuming that the acceleration of the UGV starts at $t = t_0$ and then accelerates with a m/s^2, then it will take V_A/a seconds and

$$\int_0^{V_A/a} V_{UGV}(t)dt = \frac{V_{UAV}^2}{2a} \tag{5.1}$$

meters for the UGV to reach the same speed as the UAV. Since the UAV would have traveled V_A^2/a meters in this time, it means that the acceleration should start when the UAV is $V_{UAV}^2/2a$ meters behind the UGV starting position. This estimation is in real time updating the estimated starting point of the acceleration based on the current velocity of the UAV V_A.

The assumptions for this model are not always correct for several reasons. First of all, in the general case the UAV speed will not be constant. Secondly, the UAV might at times deviate considerably from its assumed trajectory straight in the direction of the runway. This also means that the velocity in the x-direction will be varying. As such, the assumption that the UAV is flying with its velocity in the runway direction will be correct only to a varying degree depending on how the velocity control is implemented and on how well it follows the desired trajectory.

Still, initial simulations showed that this estimation works fairly well. This can be explained by the fact that the UGV in its acceleration phase will adapt its acceleration according to the current position difference, at the same time as the UAV is correcting for errors. In that sense, the estimation is self-correcting during this phase.

5.3 Go-Around Logic

In order to prevent crashes and dangerous maneuvers, the UAV control system contains logic for preventing these states to occur. Would such a state occur, the UAV automatically turns into "go-around mode" and autonomously cancels the landing, pulls up to reach a safe altitude and increases its velocity. The go-around mode is initiated in any of the following cases

- The UAV is attempting to land with the UGV too far in front, to the sides, or in the back
- The lag in data transmission is too high.

5.4 Retard and Ground-Lock

In order to avoid damages to the UAV, the UGV, and the landing platform, the throttle needs to be cut-off as the UAV reaches a certain point above the UGV. This phase is referred to as the retard phase, and it occurs during the very last, critical seconds of the landing. An effect of the engine shut down is that the UAV TECS implementation will go into either speed priority mode or path priority mode. Speed priority was chosen since the vertical velocity could be adjusted with an appropriate selection of retard and ground lock activation altitude.

When the UAV goes into speed priority mode, the UAV tries to regain speed with the use of the elevator, making it pitch down and increase its sinking rate. This creates a downward motion relative to the ground vehicle. Since the retard is initiated during the very last seconds of the landing, when the UAV is supposed to land, this behavior is not only acceptable but also desirable since we want the UAV to "dive".

Finally, the UAV also stops controlling its surfaces and instead puts them into their final landing mode:

- Ailerons are deflected upward to reduce lift

- Elevators are deflected downward to generate a nose down moment. They are deflected to 50% of the maximum deflection to allow for active pitch damping.
- Flaps are fully retracted (0 degrees) to reduce lift

This is done to avoid a situation where the UAV regains altitude after the touchdown.

Chapter 6

Vehicle Position Alignment

The cooperative landing control problem can roughly be divided into two parts - lateral/longitudinal and vertical control. The lateral/longitudinal control should align the two vehicles in a way that makes it safe to land, and this should happen independently of the altitude difference. The aim of the descent control on the other hand needs to be dependent on the vehicle position for safety. The position control is a multiple input multiple output control problem, which has many potential solutions with different degrees of sophistication and complexity. Examples from formation control to solve similar problems includes PID control [14], artificial potentials [5], and optimal control with MPC [18]. This chapter presents a PID control approach and evaluates how it changes under different feedback structures.

6.1 Groundspeed Control

Typically, the UAV would follow an airspeed command since it is the airspeed and not the groundspeed that determines the aerodynamic properties of the aircraft. Following a groundspeed command could be dangerous in windy conditions since this might force the vehicle out of the range of safe airspeeds. With the use of flight envelope protection we can still command groundspeeds without the risk of putting the UAV in a dangerous state. The aircraft will then simply stay within the limiting airspeeds

$$V_a^{min} \leq V_a \leq V_a^{max}.$$

By using groundspeed commands both vehicles directly control the same physical quantity, which facilitates the alignment process. In addition, the aircraft will actively compensate for atmospheric disturbances such as winds or the close proximity effects from the car. As the UAV approaches the UGV, the airflow around the car will start to affect the airspeed of the UAV. With an airspeed command, this will result in sudden speed changes in the very last seconds of the landing. This is not desirable because of the added crash risk and the extra time that it will take to land.

In order to make the UAV follow groundspeed, the following airspeed command is given

$$V_a^{des} = (V_k^{des} - V_k) + V_a$$

The ground vehicle lacks the velocity envelope protections that are present in the UAV, making it more flexible when it comes to velocity adaptation. Thus, it is easier to make the UGV velocity follow the UAV velocity, while the UAV tries to reach a certain desired landing velocity V_{land}.

6.2 Control Structure

The objective is to come up with a strategy that forces convergence to zero for the size of the position errors

$$\left\| \begin{matrix} x_{UGV} - x_{UAV} \\ y_{UGV} - y_{UAV} \end{matrix} \right\|.$$

Inputs to the vehicles are the desired velocities and the desired course angles. With the assumption that the course angles χ_{UAV} and χ_{UGV} are small we obtain the approximation

$$\begin{aligned} \Delta\dot{x} &= V_{UGV}\sin\chi_{UGV} - V_{UAV}\sin\chi_{UAV} \\ &\approx V_{UGV} - V_{UAV} \qquad (6.1) \\ \Delta\dot{y} &= V_{UGV}\cos\chi_{UGV} - V_{UAV}\cos\chi_{UAV} \\ &\approx V_{UGV}\chi_{UGV} - V_{UAV}\chi_{UAV} \qquad (6.2) \end{aligned}$$

and so it seems as if a good strategy would be to control velocity- and x-deviation with the velocity inputs, and to control course angle and y-deviation with the angular inputs.

6.3 Longitudinal Feedback

Using groundspeed as input to the longitudinal system, three basic feedback structures for the longitudinal control can now be identified as

$$\left.\begin{aligned} V_{des}^{UAV} &= V_{land} \\ V_{des}^{UGV} &= V_{UAV} + k_1\Delta x \end{aligned}\right\} \qquad (6.3)$$

$$\left.\begin{aligned} V_{des}^{UAV} &= V_{land} + k_2\Delta x \\ V_{des}^{UGV} &= V_{UAV} \end{aligned}\right\} \qquad (6.4)$$

$$\left.\begin{aligned} V_{des}^{UAV} &= V_{land} + k_2\Delta x \\ V_{des}^{UGV} &= V_{UAV} + k_1\Delta x \end{aligned}\right\} \qquad (6.5)$$

where V_{land} is an airspeed appropriate for landing, and k_1 and k_2 are constants. The first structure lets the ground vehicle do all the longitudinal control while

the UAV keeps a constant velocity. The second lets the UAV do the positioning control while the UGV adapts its speed to that of the UAV. The final control structure combines the two previous and makes both vehicles simultaneously react to differences in position.

In Sections 2.6 and 2.5, a linear model of the UAV and its control system has been derived. The UGV can in a simplified manner be described as a 3DOF vehicle, with an inner controller from desired velocity and course angle to throttle and steering wheel rotational inputs. The UGV linear model can be found in Appendix C.

By adding the UAV and UGV systems together, the total longitudinal system can now be expressed as one single system decoupled into the parts represented by the UAV and the UGV

$$\begin{aligned}\begin{bmatrix}\dot{x}_{UAV}\\ \dot{x}_{UGV}\end{bmatrix} &= \begin{bmatrix}A_{UAV} & 0\\ 0 & A_{UGV}\end{bmatrix}\begin{bmatrix}x_{UAV}\\ x_{UGV}\end{bmatrix} + \begin{bmatrix}B_{UAV} & 0\\ 0 & B_{UGV}\end{bmatrix}\begin{bmatrix}u_{UAV}\\ u_{UGV}\end{bmatrix}\\ \begin{bmatrix}y_{UAV}\\ y_{UGV}\end{bmatrix} &= \begin{bmatrix}C_{UAV} & 0\\ 0 & C_{UGV}\end{bmatrix}\begin{bmatrix}x_{UAV}\\ x_{UGV}\end{bmatrix}\end{aligned} \tag{6.6}$$

where x is the state vector, u the input vector, and y the output vector. The inputs to the longitudinal system are the desired velocities of each vehicle, and the outputs are the positions and velocities. The system is closed by choosing a feedback according to one of Equations (6.7)-(6.5). The structure of the longitudinal closed system is illustrated in Figure 6.5, with $F(s)$ being equal to

$$F(s) = \begin{bmatrix}-(k_P^2 + k_I^2\frac{1}{s} + k_D^2 s)\\ k_P^1 + k_I^1\frac{1}{s} + k_D^1 s\end{bmatrix} \tag{6.7}$$

where k_i^j is a positive constant.

The root locus of the system transfer function from input to output under different feedback are shown in Figures 6.2-6.4. Here $k_I^1 = k_I^2 = k_D^1 = D_D^2 = 0$.

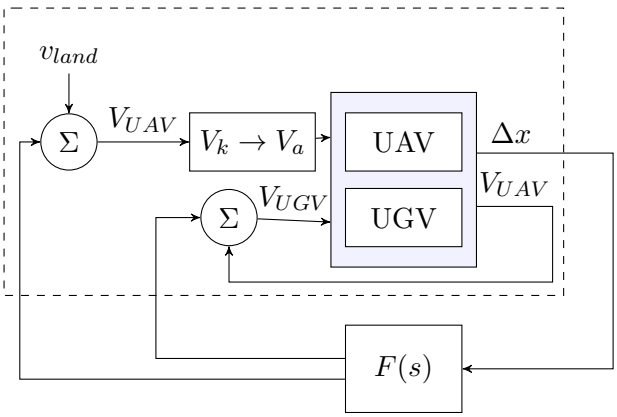


Figure 6.1: The longitudinal controller is made up of two parts; the velocity command, and a correction term that is dependent on the error in position.

Figure 6.2 shows the root locus when the relative position is fed back only to the ground vehicle ($k_P^2 = 0$), and Figure 6.3 shows the case when the position is fed back only to the UAV ($k_P^1 = 0$). The final root locus in Figure 6.4 shows the case when $k_1 = k_2$ is varied.

Noting that in the second case the open loop system contains no integrator, this would have to be added to the controller structure in order to guarantee a zero steady state error. Apart from this it is not directly clear if any structure is better than the other. In the case when the UGV controls the distance, all poles of the UAV are canceled and so the oscillatory dynamics of the UAV is not directly observable. In reality these stable modes will unintentionally be excited by disturbances.

6.4 Lateral Feedback

An advantage with doing the main part of the lateral control from the UAV is that the UAV has the possibility to correct for differences already prior to the start of the acceleration phase. The UGV will typically start from the middle of the runway, which is where the landing should ideally take place, and so it should not have to make a considerable lateral correction in any case.

One reason why it in this project is preferable with lateral feedback only to

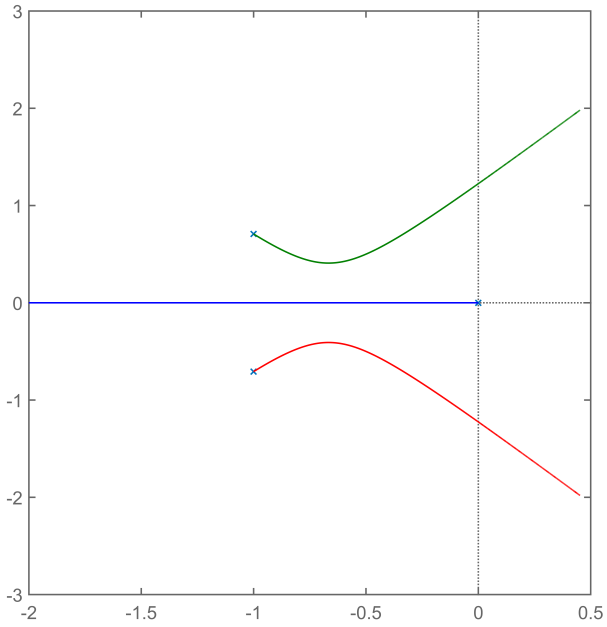


Figure 6.2: Root Locus Plot showing how the placement of the poles depend on the constant k_1 in Equation (6.7). The position difference is fed back only to the UGV. The UAV velocity is now constant and so the dynamic modes will never be excited. This makes the dynamic of the UAV unobservable with this feedback.

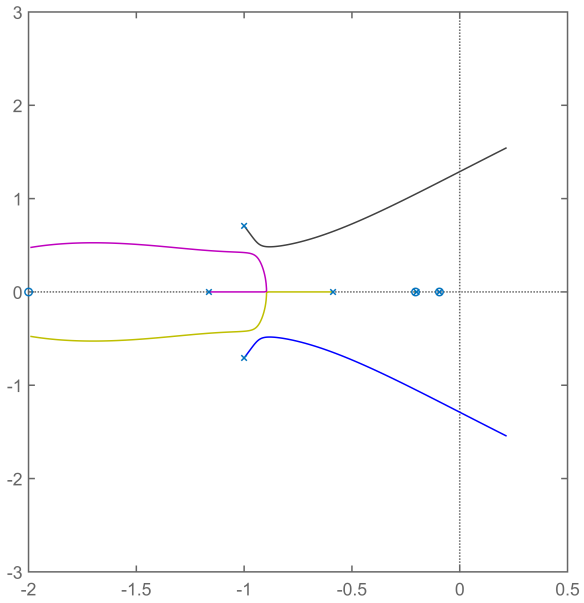


Figure 6.3: Root Locus Plot showing how the placement of the poles depend on the constant k_2 in Equation (6.4). The position difference is only been fed back to the UAV. This results in a removal of the integrator in the open-loop system, and so an integrator would have to be added in order to give a zero steady state error.

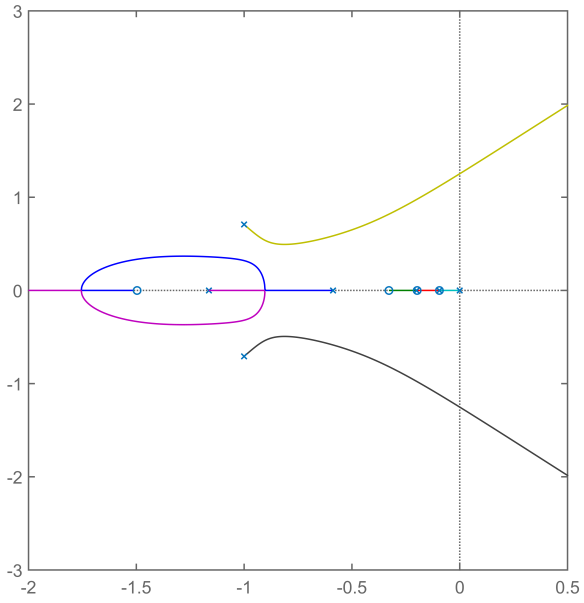


Figure 6.4: Root Locus Plot showing how the placement of the poles depend on the constants $k_1 = k_2$ in Equation (6.5). Here, the vehicle distance is fed back to both vehicles.

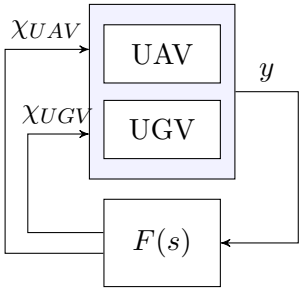


Figure 6.5: The lateral controller is dependent on the error in y-position.

the UAV is that it reduces the complexity for the human driver. Initial simulations have shown that for the conditions defined in the project requirements, it is not necessary for the human to perform lateral control.

6.5 Position Controller

It is not possible to motivate the choice of making the individual controllers cooperative from the above discussion alone. One thing that this analysis misses is that the linear model does not take saturation in velocity and angles into account. It also disregards the impact of the human actuator, possibly providing imperfect control and introducing extra time delays. Yet, it seems as if letting the UGV correct for any difference in $x-$position is a better choice - mainly since this feedback results in a more damped system. For the lateral controller the simplest choice is to make the UAV do all the control.

One thing to note about this choice is that out of the four available inputs (V_{UAV}, V_{UGV}, χ_{UAV}, χ_{UGV}) only two are used. This controller is simple and estimated to be good enough for a simple landing. It makes analysis easier since it is simpler to understand the effect of cross terms when fewer inputs are used.

There are still several reasons for using all four control inputs. For the sake of robustness, having two completely controllable systems is good when landing in conditions where one vehicle is heavily disturbed, e.g. by wind. There is also the possibility of forcing the alignment to happen faster than if only one vehicle could change its speed. Using all four inputs will make the system more complex and more tuning would be needed.

Another method for controlling the longitudinal and lateral positions could be to generate trajectories for the vehicles to follow, and then use feedback to control deviations from these trajectories. Trajectory generation and following is a quite large field for both unmanned ground and aerial vehicles ([19], [20], [21]) and so a number of solution methods exist. It was indeed suggested early on in the EC-SAFEMOBIL project that the pseudospectral collocation method would be used for trajectory generation, however, it was judged to be more advanced than what was needed at this time.

Chapter 7

UAV Descent

In Chapter 6, a simple control strategy was suggested and found to have an acceptable performance for the lateral and longitudinal vehicle position control. The next step is to consider the vertical control, or descent, of the UAV. The landing strategy needs to be safe and at the same time fast. In addition, since a landing should not happen before the alignment is completed, the descent is also limited by the performance of the lateral and longitudinal control.

This chapter starts with a more precise definition of the requirements of the landing, and is followed by an analysis of the initial descent strategy. The results from test flights are evaluated and new descent strategies are suggested and compared to the current ones.

7.1 Landing Requirements

The vertical control aims to land the UAV in a safe way. For this to happen, several things need to be taken into account. First of all there are the physical limitations of the system, limiting the safe sinking rates at different altitudes and the possible airspeeds at landing. Secondly there are the limitations related to the distance between the vehicles. The landing cannot happen too far to the front, back, or to any of the sides of the landing platform.

$$\begin{aligned} V_{land}^{min} \leq V_{land} &\leq V_{land}^{max} \\ \dot{h}_{land} &\leq \dot{h}_{land}^{max} \\ |\Delta x| &\leq \Delta x^{max} \\ |\Delta y| &\leq \Delta y^{max} \end{aligned}$$

Since the physical limitations for landing apply to any aircraft, many strategies for adapting speed and sink rate for landings exists and are in use. These are often referred to as *flare laws*.

For the distance limitations to be fulfilled the descent somehow needs to consider the state of the system to be able to give an estimation of how close it is to being able to land. It is then possible to adapt the sink rate to this information.

7.2 Flare Laws

An aircraft decent flight path angle typically consists of two parts. The first one has a more or less constant flight path angle, and therefore also a constant sink rate. In the later part of the descent, the aircraft needs to start reducing its vertical speed, and therefore its flight path angle, as it approaches landing. This part of an aircraft's trajectory is called the landing flare. A typical shape of such a landing trajectory is shown in Figure 7.1.

7.2.1 Altitude Dependent Flare Law

A common strategy for the flare is to have the reduction in sink rate proportional to the altitude, driving the sink rate to satisfy

$$\dot{h} + \frac{1}{T}(h + h_B) = 0 \tag{7.1}$$

where h_B is an offset and T is a time constant. To avoid transients when changing between the initial descent and the flare, the height at which the change is made is chosen as

$$h = -(T\dot{h} + h_B) = -(T \sin\gamma \cdot V_k + h_B) \tag{7.2}$$

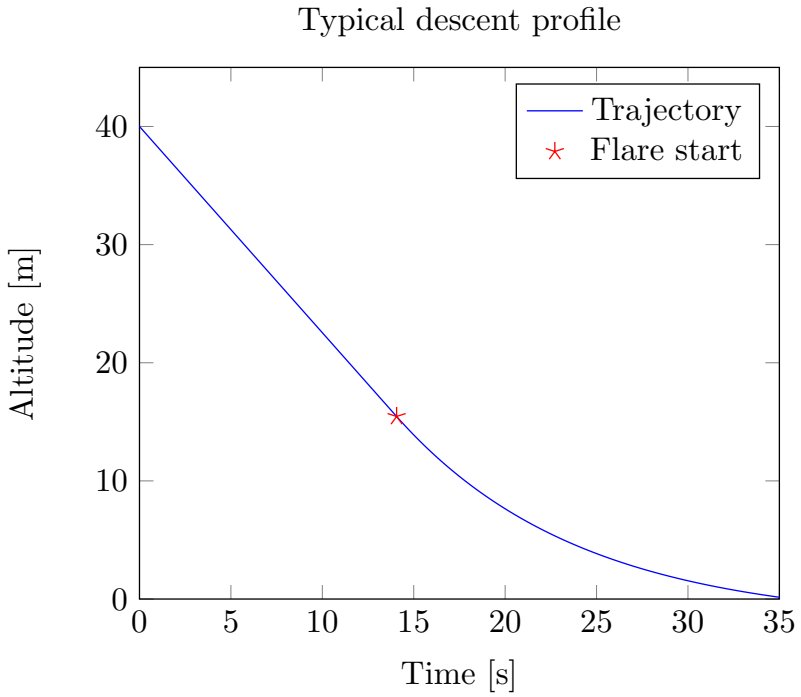


Figure 7.1: A typical landing trajectory starts with a constant descent and ends by smoothly decreasing the sink rate in the flare.

This gives a smooth transition between the descent parts. At the end of the flare, the sink rate at touchdown will be given by

$$\dot{h}_{land} = -\frac{1}{T} h_B \tag{7.3}$$

and so the offset h_B is used to get a desired vertical velocity at touchdown. A potential problem with this flare law is that it is independent of the groundspeed of the aircraft, and as an effect it does not take different wind conditions into account. This will result in the touchdown occurring at different positions on the runway depending on the groundspeed of the aircraft, making the shape of the path and the landing distance unpredictable.

7.2.2 Variable Time Constant Flare Law

An alternative flare law, proposed by Lambregts in [22], also takes the current velocity into account when controlling the sink rate. Here, the time constant is changed to $\frac{T_0 V_{G0}}{V_G}$, making the sink rate satisfy

$$\frac{T_0 V_{G0}}{V_G} \dot{h} + (h + h_B) \quad = \quad 0 \tag{7.4}$$

where T_0 is a time constant and V_{G0} is a nominal ground speed. This flare equation gives the flare a fixed shape independent of the groundspeed. This type of flare is convenient particularly when a narrow touchdown dispersion is required. The initiation altitude of the flare is independent of the groundspeed and equals

$$h_{flare} = T_0 V_{G0} \sin\gamma - h_B \tag{7.5}$$

and the flare length is equal to

$$x_{flare} = T_0 V_{G0} \ln\left(\frac{h_{flare}}{h_B} + 1\right) \tag{7.6}$$

And so by choosing appropriate constants the landing distance can be specified in advance. This flare law can thus be preferred in cases when the touch down position is of greater importance than the touch down sink rate.

7.3 Initial Descent Strategies

7.3.1 Flare Law Descent

The descent strategy used in the initial flight tests was to control the flight path angle γ by starting with a constant glide slope, and ending with the variable time constant flare law in Equation (7.4).

In addition to this, the initial descent γ was in the implementation limited on the lower side by

$$\gamma_0 \geq \gamma_{lim} = -\frac{\Delta h}{\Delta x}, \tag{7.7}$$

where Δh is the relative altitude and Δx is the distance in x, in order to avoid landing too far behind the UGV by limiting the descent when the distance between the vehicles is large.

7.3.2 Switching Descent Strategy

In other flight tests, a second type of descent was in use. The strategy is similar to the altitude dependent flare law in Equation 7.1 but with several constraints added to it. The intention with this switching descent law was to avoid landing too early by forcing the movement of the UAV into a certain volume. The descent should then only occur within this safe volume (Figure 7.2).

If the UAV leaves the volume during the time of the flare, then it will reach the "retry mode" and start to gain altitude until it reaches a predefined safe altitude again, and thereafter try to descend.

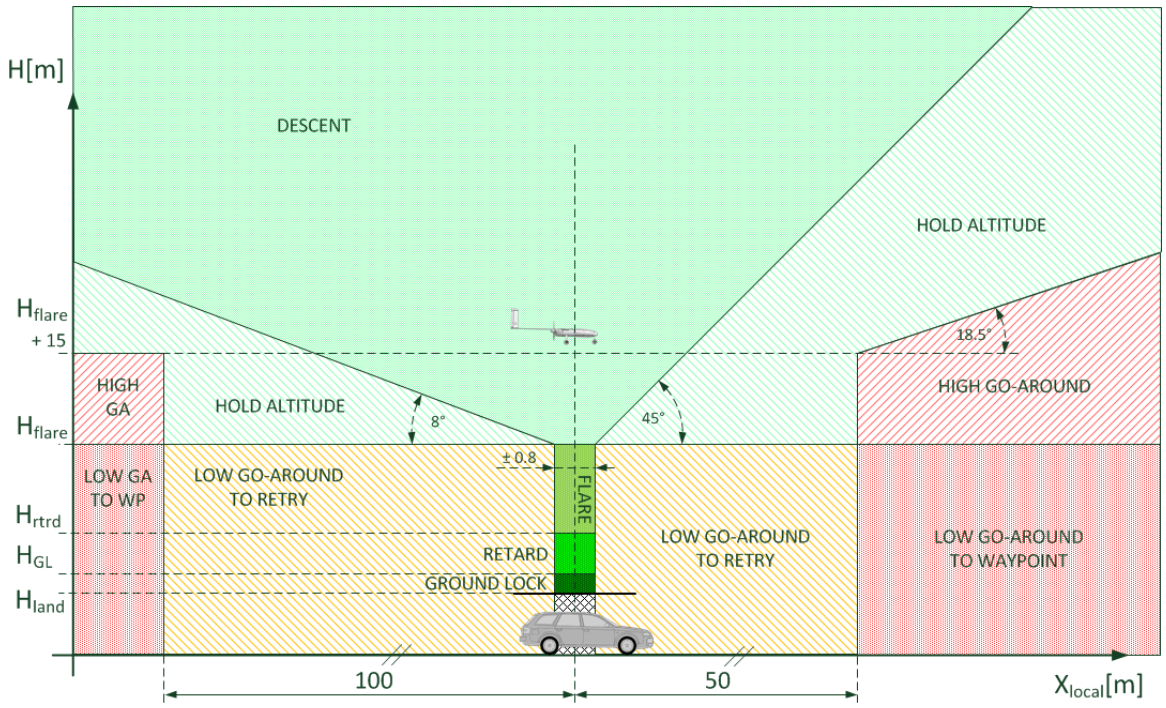


Figure 7.2: The sink rate is chosen in a way such that the flare is only active while inside a certain geometric limit.

Chapter 8

Experimental Results

Three main types of tests were performed; hardware-in-the-loop, flight tests with an altitude offset, and flight tests where the UAV tried landing on the actual landing platform. This chapter presents some results from these tests. Due to some of the collected data being found to be corrupted afterwards, with e.g. the GPS data being collected in a faulty way, and some of the flights being interrupted for other reasons, e.g. bad weather, the data presented in this chapter is mainly based on Flight Experiments 11-13, but also uses some data from Flight Experiments 7 and 9. A list of the performance of the controller in the last experiments can be found in Appendix A.

8.1 Hardware-in-the-Loop Tests

Initially we performed hardware-in-the-loop experiments, where the UGV and the communication setup was used at the same time as a real-time process on a PowerPC running the ROS QNX with a simulation of the UAV in Simulink. The main outcome of these tests was a better insight into the problems that the driver of the car would face. These tests were done under a variety of simulated conditions and controller settings. It was more difficult for the driver to do lateral and longitudinal steering at the same time than to only do the longitudinal control. The tests showed that it was in most cases not necessary for the UGV to do any lateral correction, in fact, it was less efficient which might be due to the difficulty in simultaneously following the commands.

8.2 Flight Tests With Virtual Runway

After the hardware-in-the-loop experiments, the UAV was incorporated into the experiments with an offset in altitude, creating a "virtual runway" for the UAV to land on. This offset ranged between 110 m and 0.75 m depending on what tests were being performed and how safe the altitude would be with regards to the weather circumstances. This testing mode was used for evaluating the acceleration phase, the lateral and longitudinal control, as well as the flare, ground-lock, and retard mode.

Waypoint Flight and Acceleration Phase The waypoints were set prior to each flight in accordance to the safety pilot's opinion on what was suitable. The weather conditions together with how these waypoints were set affected the landing initiation coordinates. The starting points of the last two flight days can be seen in Tables 8.1-8.2. A list of the initial values during the later flights is available in Appendix A.2. There is a relatively large spread in the initial values. These values provide an indication of the circumstances under which the controller is expected to work.

	h [m]	Δx [m]	Δy [m]	V_k [m/s]	χ [deg]
Mean	27.45	201.49	8.70	23.29	1.57
Min	25.42	152.84	−0.97	21.49	−3.8
Max	28.82	237.34	27.31	24.32	−4.1

Table 8.1: Initial values, Flight Experiment 12

	h [m]	Δx [m]	Δy [m]	V_k [m/s]	χ [deg]
Mean	24.23	230.67	1.96	22.60	−6.97
Min	24.13	113.59	−22.10	20.04	−18.7
Max	24.50	309.02	42.49	23.78	4.0

Table 8.2: Initial values, Flight Experiment 13

After exiting the waypoint mode the vehicles went into the acceleration phase. At first, the UGV is standing still and waiting for the UAV to approach. The starting point of the acceleration is calculated according to the description in Chapter 5.2. With a velocity of 21 m/s, this would make the start position of the acceleration phase to be

$$x_{start} = \frac{v_{UAV}^2}{2a} = \frac{21^2}{4} = 110.25 \text{ m}$$

behind the UGV starting position. The acceleration would under perfect conditions be constant until the desired speed is reached. This approximation was shown to be quite accurate in most cases. Many of the flight tests had acceleration phases such as the one shown in Figure 8.1, with a constant acceleration that ends close to the same time as the vehicle positions overlap, as was intended.

Not all landing attempts followed this pattern for the acceleration phase. This occurred for example as a result of poor initial conditions. In the example shown in Figure 8.2) there is a large initial error in Δy, making the UAV fly with a considerable part of its velocity in the y direction. This makes the UGV overestimate the time it will take for the vehicles to align, leading to an overshoot in Δx.

The acceleration phase strongly affects how effective the landing will be in terms of time and spent distance, which is why we need the start of the acceleration to be accurate. One way to do this would be to improve the waypoint flying in order to get better initial coordinates for the landing. Another way

to would be to use a more sophisticated model for calculating a starting point for the acceleration. Such a model would have to take the distance and velocity difference in the y-direction into account as well.

Longitudinal and Lateral Control The position development during most flight experiments can be found in Appendix A.3. Initially the performance of the controller was not particularly good, in part due to problems related to position estimation and inaccurate GPS data. After these initial problems were solved, and with some additional tuning of the control parameters, the system performed quite well. Even though the parameters were not changed more than one or two times per parameter during all flight tests, the PID approach gave results in the order of what was needed for this landing. If more time would have been spent on the tuning, it is possible that the performance would improve even more.

The two main problems that we fixed during the control tuning were a seemingly steady state error in the y-direction and overshooting and slow convergence in the x-direction. The former was solved by including an integrator in the lateral control. For the latter we increased the proportional part until the convergence speed was satisfactory.

Even though it was never necessary with more accuracy or speed for the sake of these experiments, it is likely that future applications will need more a more efficient controller. Other than tuning the current PID approach, there are many possibilities for other types of controllers that might show a better performance, as mentioned in Section 6.3.

Descent Using Adaptive Time Constant Flare Law Figure 8.3 shows an example of the performance of the adaptive time constant flare law during one of the test flights. In this example the virtual landing occurs before the UAV has had a chance to catch up with the UGV in the x-direction, and the y-direction did not aligned either. This is clearly an undesirable result. Table 8.3 presents the landing distances of all landing attempts during this flight experiment. The results are varied, with some successful tries but also many bad ones. Apparently, landing without any consideration to the actual feasibility of this task works only under specific circumstances.

Since we want to be able to land under a wide variety of conditions, this approach to landing is not good enough. In the case of the landing shown in Figure 8.3, would the sink rate have been controlled, the touchdown might not have occurred this early and the UGV would have had a chance to correct for the distance in x. The problem in this specific case was that the UAV was too slow and that it did not get enough time to reach the UGV, a problem which could have been avoided e.g. by changing the sink angle γ.

For the sake of robustness, having the descent law dependent on the distance between the vehicles is important since there are cases when the lateral/longitudinal control of the vehicles will not be as fast as intended. An example of this is flight in heavy wind conditions where the groundspeed of the UAV might be limited due to reaching the upper or lower limit in airspeed. This

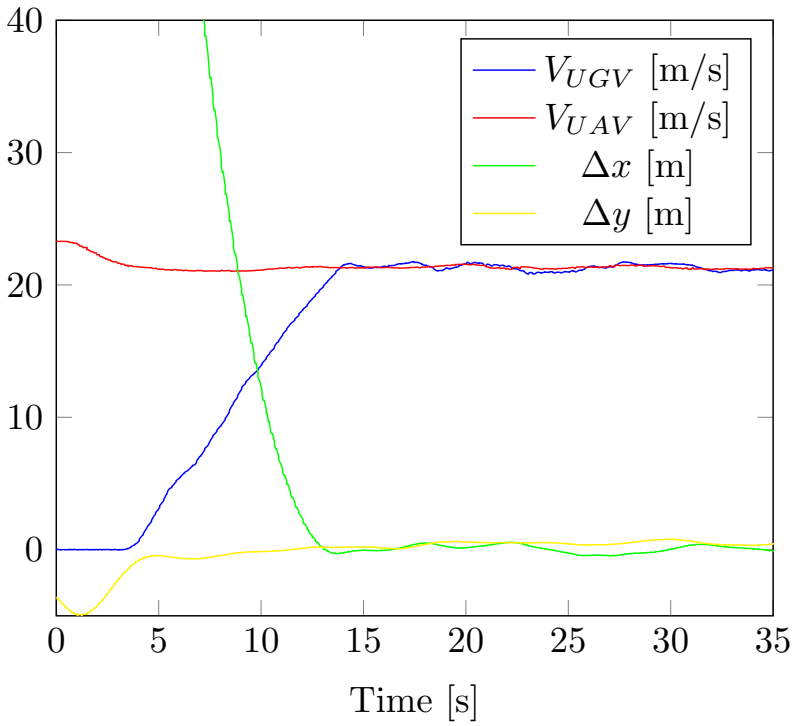


Figure 8.1: An example of the acceleration phase. The acceleration is close to 2 m/s^2 and lasts around 10 seconds and 120 m.

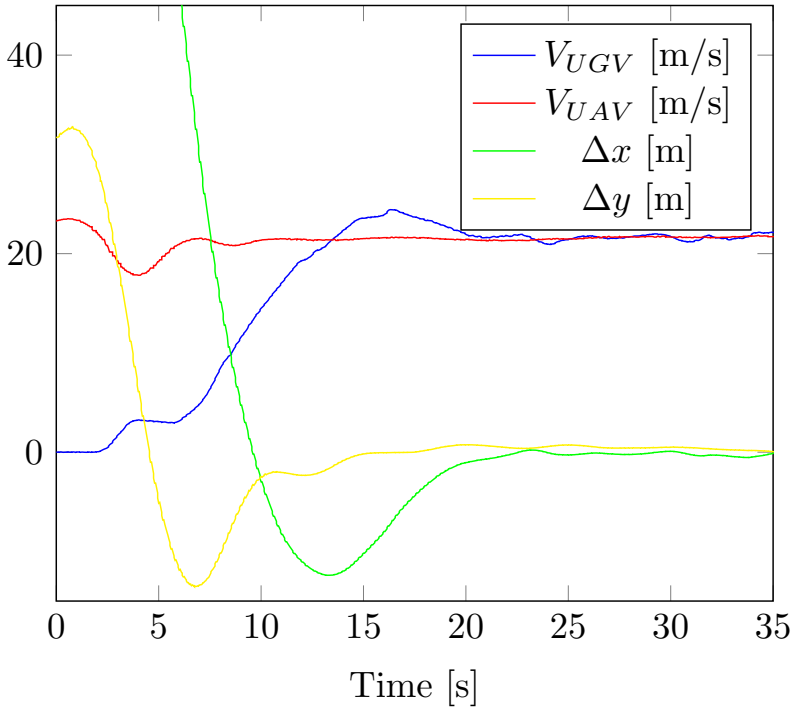


Figure 8.2: An example of the acceleration phase. The acceleration is not constant and the same speed is not reached at the same time as the positions are aligned. The slow acceleration leads to an overshoot in Δx.

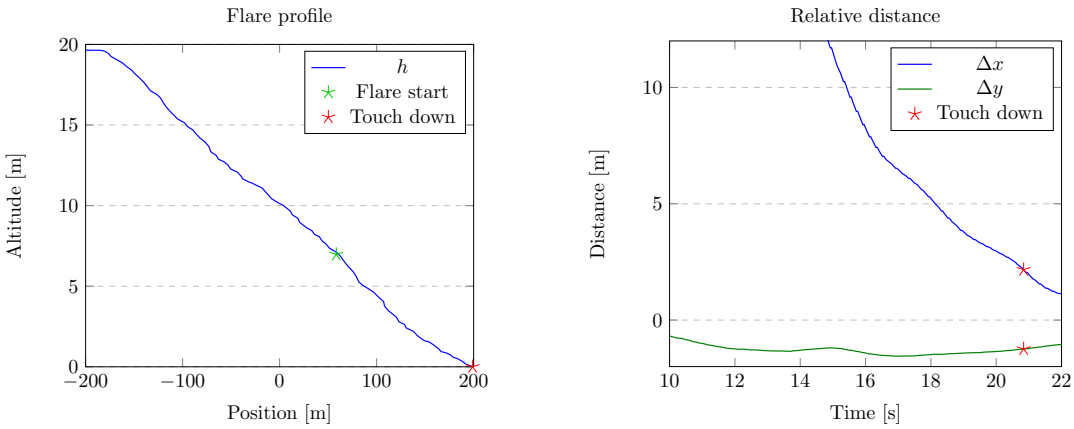


Figure 8.3: Example of failed attempt using the initial descent approach. When the distance is not taken into account, there is the risk that the vehicles will not be close enough at touchdown. Here (attempt 11.3) touchdown at the virtual runway occurs at $\Delta x = 2.1667$ m, $\Delta y = -1.2419$ m

flare law is too dependent on having good initial values and low external disturbances close to the landing for direct use in the context of landing the UAV on top of the UGV.

Touchdown precision, Flight Experiment 11								
Δx [m]	-9.070	12.597	2.1667	-0.283	-1.166	0.132	0.188	-0.997
Δy [m]	-0.731	-1.493	-1.2419	-0.632	-0.649	-0.333	-0.776	-0.856

Table 8.3: Final distance in x and y at the time of touchdown on the virtual runway. The x-controller was not used on the UAV, which in part explains the bad behavior there.

Descent using switching descent law Since it was evident that using a normal decent law would not be robust enough, testing turned towards using a switching strategy. The biggest drawback of such a method is that unexpected behavior might occur when going between different flight modes. Guaranteeing that a switching controller will perform a certain way can be difficult because of the hybrid nature of the total system. It is known that even a combination of stable subsystems might become unstable when used together with a switching control if used incorrectly [23]. Another reason why it might be difficult to design such a system is that it is difficult to foresee what might happen with different combinations of system states and control modes. An example of this type of problematic behavior can be seen in Figure 8.4, where the retry mode and the coupling in velocity and altitude causes the x position to oscillate as the altitude is going between $h_1 = 10$ m and $h_0 = 5$ m. The change in x causes further retries, forcing the system into a cycling between the retry mode and the descent mode. The UAV command in this mode was later changed to give a constant velocity

to avoid this problem.

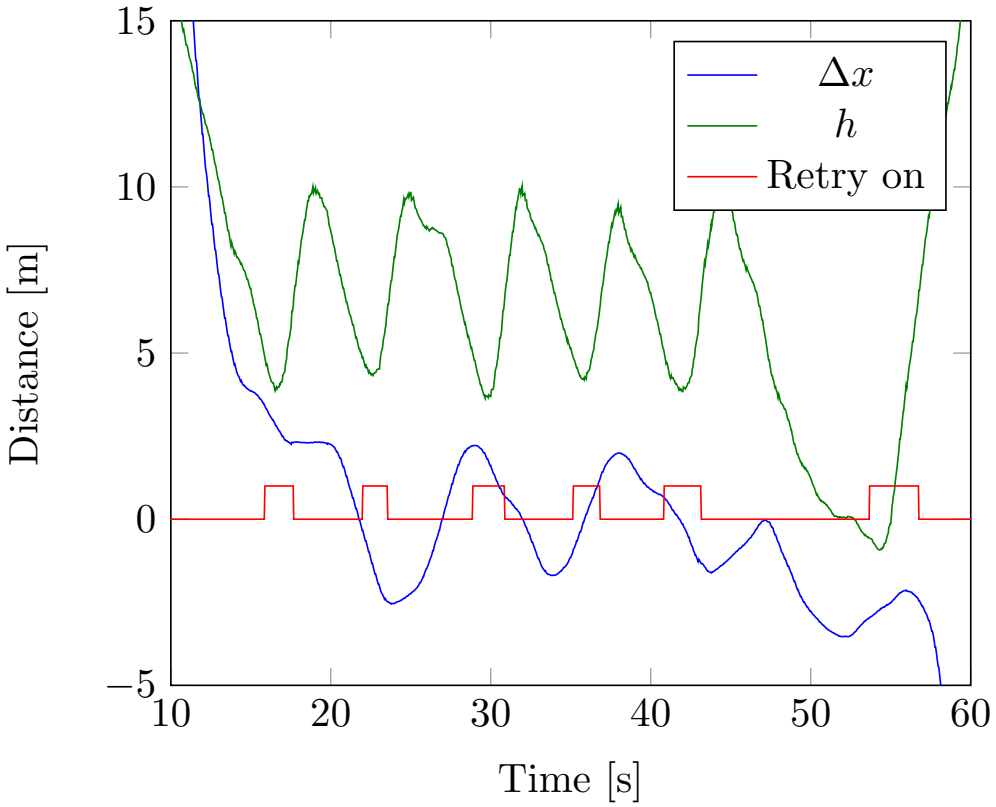


Figure 8.4: When the geometry based switching controller was used, a cycling switching between descent and retry modes was observed. Data from flight test 12.

Even when additional arguments are added to the switching logic to prevent such cycling, other problems might arise either due to switching transients or due to the UAV ending up in a state that it was not intended to enter. Although landing with this approach is possible it requires that the number of modes and the number of conditions for switching between them grow in a way that the control might be more complex than needed. Using retry switching as the default descent strategy rather than as a safety mechanism is also not ideal since it causes the system to go in and out of the retry mode, which probably is not the most effective solution.

Figure 8.5 shows landing attempts where the final phase is initiated too early, causing the UAV to go into retry mode several times. The initial glide slope was then decreased from -5 degrees to -3 degrees which partially made this problem avoided. Instead of manually having to change parameters to decrease the number of times the UAV goes into retry mode, a more sophisticated algorithm for descending could have adapted the descent to a more suitable value so that the UAV could have landed on the first attempt. The switching logic and the parameters could possibly be changed further in order to avoid this seemingly

unnecessary switching of modes.

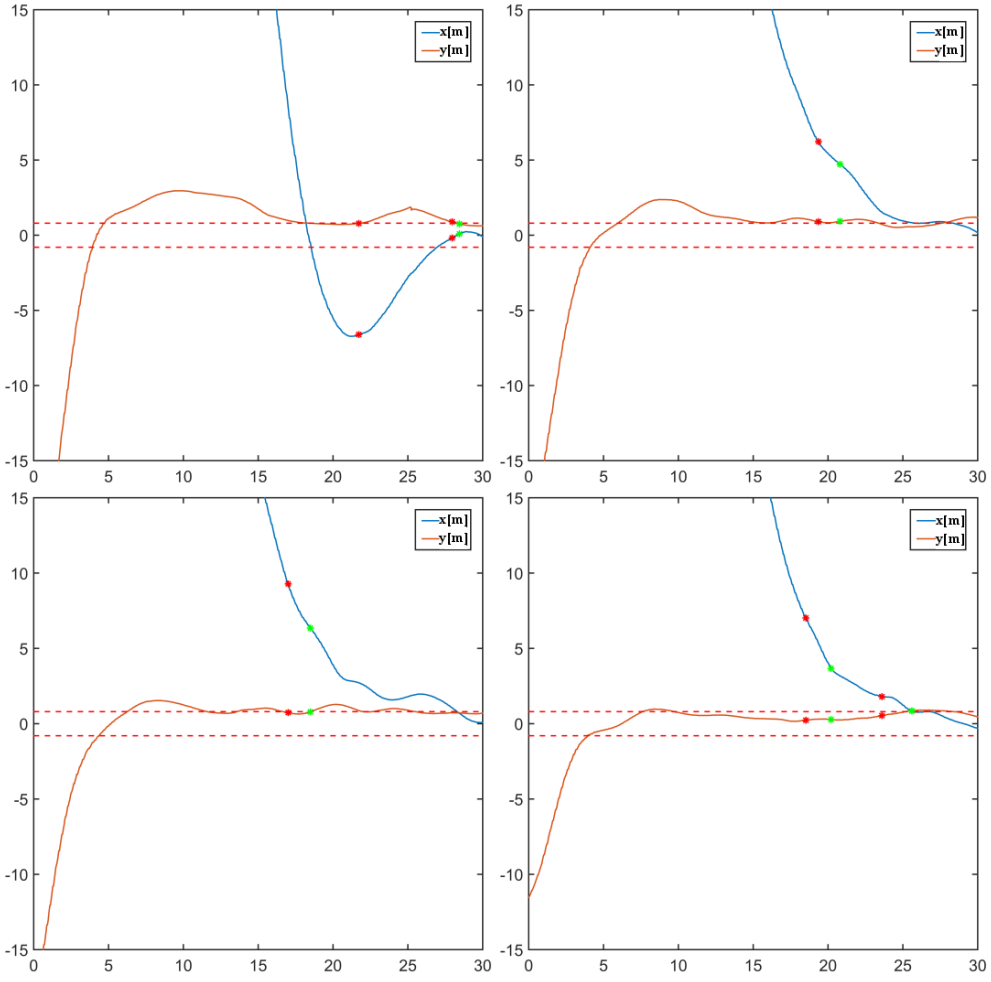


Figure 8.5: The start of flare mode (which happens when the altitude of the UAV is at 5m) is indicated by the green star, and the go around is indicated by the red stars. The blue and red lines represent the distance between the vehicles in x and y as a function of time. In these examples, delaying the final landing phase e.g. by slowing down the decent might have made the UAV avoid some of these go-arounds. Later, the initial angle was decreased manually.

8.3 Flight Tests Without Virtual Runway

Out of the five attempts to perform a landing on the actual landing platform, two were successful. Figure 8.6-8.7 show the relative positions and velocities during these landings. Both landing attempts switched to retry mode when reaching 5 m altitude because of their too large relative positions. Considering the relative positions, it appears as if the retry might not have been necessary, at least not in the case of the first landing.

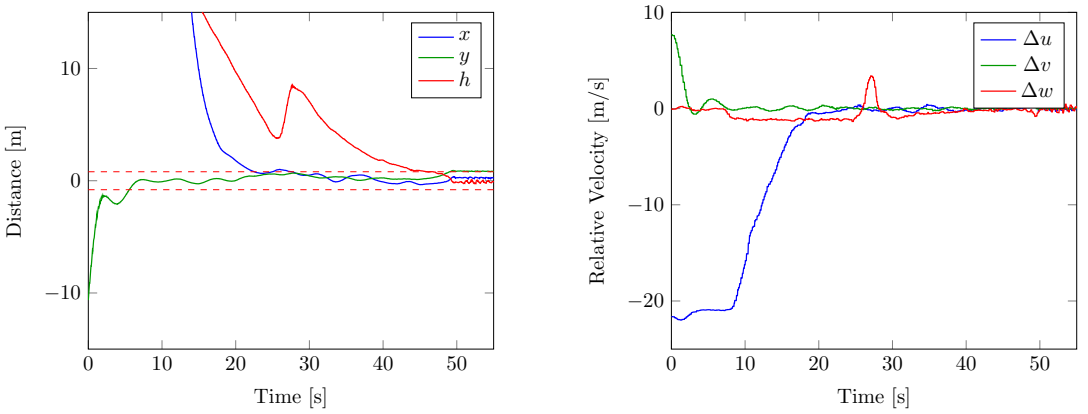


Figure 8.6: The first successful landing. The peak in h and w corresponds to the retry-mode being activated.

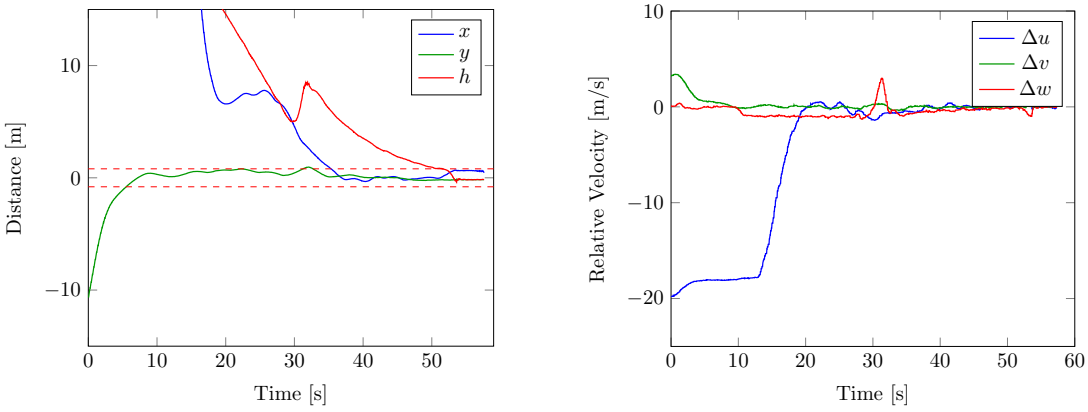


Figure 8.7: The second successful landing. The peak in h and w corresponds to the retry-mode being activated.

In both successful landings, instead of landing at around 30 seconds when the vehicles should be able to land, it takes an additional 20 seconds. It is not possible to know what would have happened without the retry, but the altitude gain to 10 meters does seem superfluous. A more effective approach might have

been to decrease or stop the sinking rate, which would have allowed the UAV to catch up with the UGV without adding a lot of extra time.

The remaining three landing attempts without a virtual runway (Appendix A.3, Attempts 13.11-13.13) switched to retry mode twice each. These retries were results both of position errors and of delays in the data transmission. For the smaller errors, such as delays of 0.35 s or errors in position of around 0.9 m, it seems as if the altitude gain did more harm than good for the cause of the landing, adding additional 20 s or 400 m used runway distance to the attempt. Since the length of the runway is limited, this possibly made the difference between success and failure.

Because these five landing attempts were initial proof-of-concept, safety was highly prioritized. The objective with these landings was to show feasibility, and this was achieved with the two successful landings. The reason why the landings were not as smooth as they might have been and that three of the attempts failed was the many safety actions taken in order to prevent a crash. As more confidence is gained in the system and its control, a more risky behavior might be tolerated. The go-around and retry modes were to a big part there to prevent crashes, and if we learn more about the system performance we might not need to make such strict limits for the retry mode.

Chapter 9

Improved Control Strategies

In the final flight experiment it was confirmed that landing was possible using the initial PID approach, although the performance could be improved. Several alternative landing approaches were investigated throughout the project. Due to time constraints they were not tested in flight, but there are plans to do this in the future. This chapter describes some of the efforts made in finding alternative control strategies. It begins with an analysis of the experiments, and tries to identify what could be changed in order to make the landings improve. Thereafter three different approaches for developing the control are suggested.

9.1 Analysis of Initial approach

The very first flight experiments used the unmodified variable time constant flare law in Equation (7.4) for the descent, and hence did not take the system state into consideration when computing the descent angle. As might have been expected, this method gave acceptable results only if initial coordinates were close to the runway centerline and the disturbances were small enough.

The next method, using a state machine to transition between descent and go-around, gave the system more tries at landing. Still, the UAV went into flare mode at times when it was apparent that it would lead to a go-around, and only two out of the final five landing attempts on the landing platform were successful. This shows that the idea behind it is good, but that there are still parts that need improvement. In particular, the current setup frequently makes the UAV reach states after which it transitions into go-around or retry mode. Avoiding ending up in such a state could increase the efficiency. Improvement of the controllers used within the different modes could prevent many of these unwanted transitions, so that the UAV only goes into flare mode when it has a reasonable likelihood of being able to land. Another way of dealing with this problem would be to extend the state machine to include logic for capturing many more types of events.

The only effort at tuning the sink rate autonomously during flight was to limit the initial glide slope γ_0 according to the relative distance between the vehicles (Equation 7.7). This method works well when the target (the UGV) is

standing still, by forcing the glide slope to be below a certain angle when the UAV is far away. This approach does not work when the target is moving, since the landing distance in this case is not equal to Δx. As such, this limit is only valid in the initial landing phase when the UGV is standing still. Nonetheless, the feature is still useful since it allows for the landing maneuver to start before the UGV starts to move.

Seeing as there was a lack of robustness and general performance in the evaluated flight experiments, it is clear that there is room for improvement of the control strategy. One way of improving it would be to proceed with the tuning of the existing controller, possibly adding more terms. Each parameter was tuned only a few times, and so better values could certainly to be found. Doing this would likely increase the performance of the controller, but the controller would probably still suffer from being sensitive to external conditions. Extending the control system with functions such as automating waypoint generation or using an adaptive controller or gain scheduling to decrease the influence of wind would further improve the performance without changing the descent approach.

Still, there might be other disturbances that have not been included. If for example the UGV temporarily needs to slow down, the desired behavior would be for the UAV to try to match this new speed, and at the same time slow down or stop its sink rate. This way, as the UGV gains speed, landing is resumed with minimal delay. In the current system this situation would have been followed by the UAV continuing landing until it reaches a condition for either go-around or retry mode. Using the retry mode is a great way of handing unexpected incidents at critical times, but it should be used as a back up and not as the main response.

As remarked in Section 8.2, in order to make the landing feasible under a wider range of circumstances and in a more efficient way, it would be beneficial to take the states of the vehicles into account when computing the descent angle. Ideally, the UAV would sink in a way such that zero relative altitude is reached at the same time as the position and velocity are aligned for the first time. This would minimize the time and distance required to land.

One method to make the descent dependent on the state would be to adapt some variable to the current error. This is studied in Section 9.2. Prediction could be used to change the rate of descent according to the time needed to position the vehicles. This is discussed in Sections 9.4 and 9.3.

9.2 Adaptive Flare Law

In this approach, the intent is to adapt the sink rate not only according to altitude, as in the case of the altitude dependent flare Law in Equation (7.1), but also according to the current error.

The error could be defined in many ways. A straightforward way would be to use the position difference, as was done in the estimation of the acceleration phase in Equation (5.1) and when calculating the limiting glide slope angle in Equation (7.7). These cases use relative position only in the x-directions, but the results are still satisfactory in most cases.

When it comes to more critical parts of the landing, such as the flare, this type of metric is inadequate. A geometrical constraint limiting the altitude based only on difference in position, such as the one described in Section 7.3.2, fails to capture essential parts of the landing problem. This is true in particular when the UAV path deviates considerably from being a straight line with constant velocity. This behavior was observed during several flight tests, where a combination of wind conditions, the human driver, and imperfectly tuned controllers resulted in the vehicles overshooting in position. An example of this is illustrated in Figure 9.1. Here it is evident that relative position alone does not provide enough information. Position difference reaches zero at a point when the relative velocity is still high, causing position difference to overshoot before returning to zero.

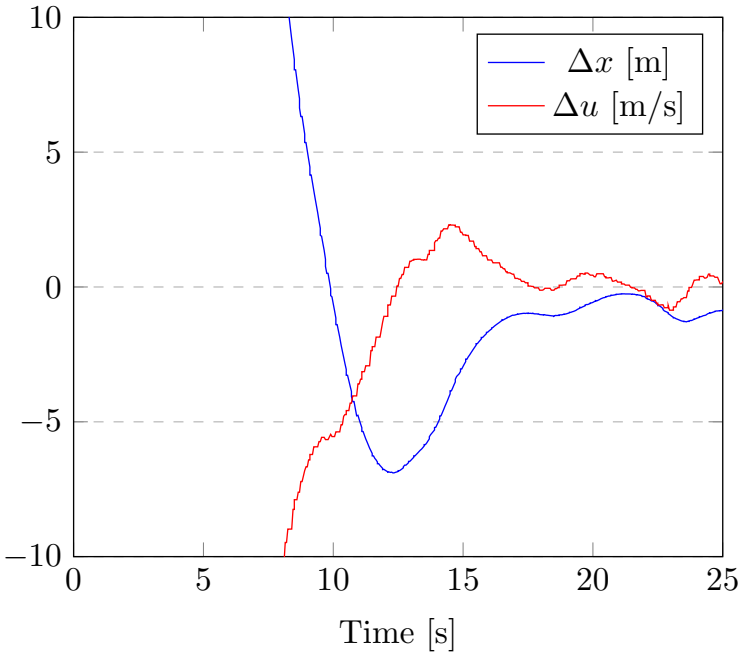


Figure 9.1: If the velocity is not taken into account when estimating when to land, then situations such as this with significant overshoots will give misleading information of the systems capability to land.

A robust controller must be able to handle disturbances and measurement errors, with possibly large overshoots and oscillations around the alignment point. To do this, the error must better reflect when landing is feasible.

Noting that both a high relative velocity and a high relative position are undesirable for landing, another error metric could be chosen as

$$e = \sqrt{\Delta x^2 + \Delta y^2 + \Delta u^2 + \Delta v^2}. \tag{9.1}$$

As noted above, a large difference in relative velocity is undesirable during landing. However, when there is a position difference and a velocity difference at the same time, this indicates that the vehicles are approaching each other and will eventually be aligned (Table 9.1).

	$x_{UAV} > x_{UGV}$	$x_{UAV} < x_{UGV}$
$u_{UAV} < u_{UGV}$	Approaching	Diverging
$u_{UAV} > u_{UGV}$	Diverging	Approaching

Table 9.1: Cases for Combinations of Position and Velocity Differences. u is the velocity in the forward direction.

The $\Delta x^2 + \Delta u^2$ term in Equation (9.1) is very large if the relative distance is large. By including a cross term $\Delta x \cdot \Delta u$, whether or not the vehicles are approaching each other also has an influence on the error. If the vehicles are approaching each other, this term will be negative, and if the distance is increasing this term will be positive. A quadratic cost

$$e^T Q e = \begin{bmatrix} \Delta x & \Delta y & \Delta u & \Delta v \end{bmatrix} \begin{bmatrix} k_x & 0 & k_{xu} & 0 \\ 0 & k_y & 0 & k_{yv} \\ k_{xu} & 0 & k_u & 0 \\ 0 & k_{yv} & 0 & k_v \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta u \\ \Delta v \end{bmatrix} \tag{9.2}$$

captures this relationship. The coefficients are then used to weigh the influence of the different factors.

Figure 9.2 shows how four different types of errors evolves during a flight. The plotted error measures are

$$\begin{aligned} e_1 &= \Delta x \\ e_2 &= \sqrt{\Delta x^2 + \Delta y^2} \\ e_3 &= \sqrt{\Delta x^2 + \Delta y^2 + \Delta u^2 + \Delta v^2} \\ e_4 &= (e^T Q e)^{1/2} \end{aligned}$$

Figure 9.2a shows these metrics in a flight when there is a relatively large overshoot. Far away, the error measures are all large. As the relative position gets closer to zero, the error metrics start to diverge. While e_1 and e_2 more or less misses to account for the overshoot, e_3 and e_4 are more smoothed out and reaches a small value only when it is actually safe to land.

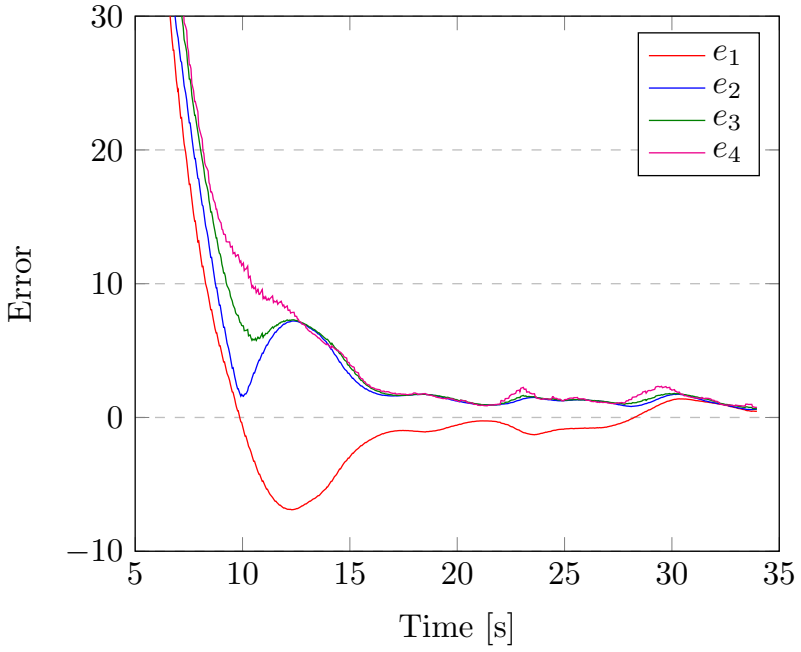
Figure 9.2b demonstrates the performance of the error measures during a flight with no overshoots. Here, the behavior is similar between the errors.

With inspiration from the flare laws in Chapter 7.2 a new flare law that takes the error into account could be chosen as

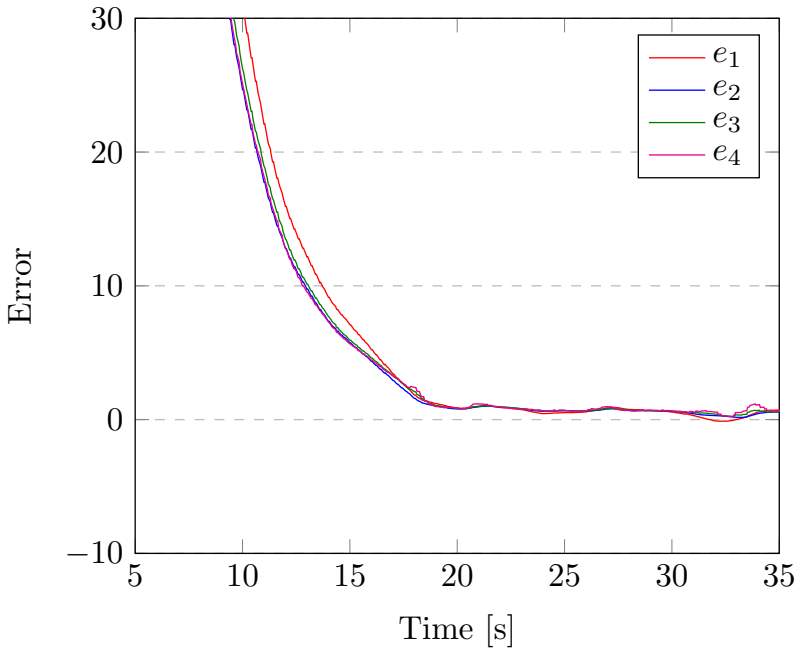
$$\dot{h} = -\frac{1}{(e+1)T}(h + h_B). \tag{9.3}$$

so that the time constant is scaled with the error.

Then, as the error e goes toward zero, the sink rate will go toward the altitude dependent flare law (7.1). This means that the sink rate is dependent on the error and the altitude. The sink rate increases with altitude and decreases with error. One problem with this is that it only slows the descent down since $\dot{h}$ only asymptotically goes toward zero as the error goes to infinity. This means that landing will happen even if the error is relatively large during the entire descent.



(a) Flight with large overshoot in position



(b) Flight with no overshoot in position

Figure 9.2: Comparison of different error measures. By including more information in the error term, a more accurate measure of the actual distance to a feasible landing is acquired.

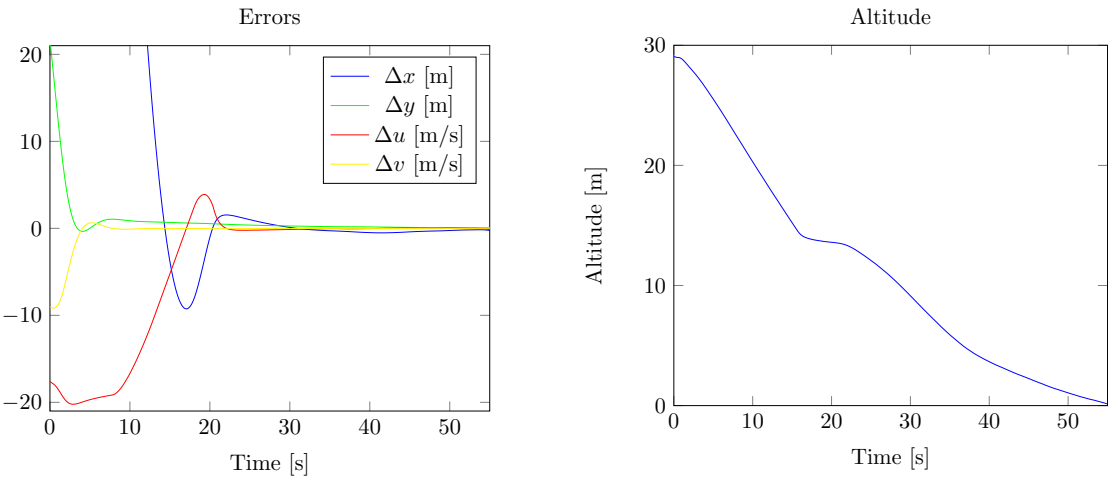


Figure 9.3: When using the adaptive flare law, the sink rate is slowed down due to the poor landing conditions in the beginning

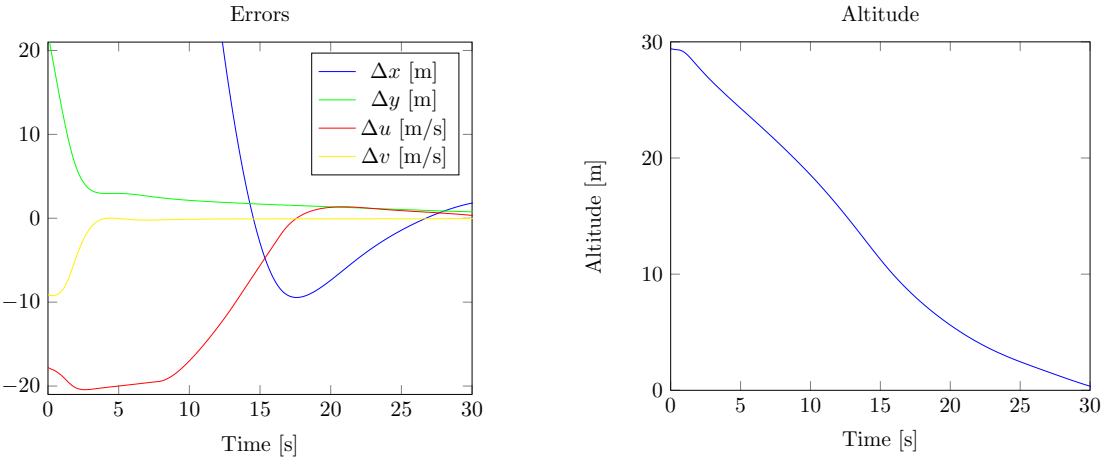


Figure 9.4: When using the normal altitude dependent flare law, the sink rate is independent of landing conditions and this results in a landing before it is possible.

Figures 9.4 and 9.3 show the performance of the error dependent sink rate compared to the sink rate used in flight for a system with bad initial conditions, disturbances and bad controller settings.

In this example, the start of the flare is done at a fixed altitude of 15 m. As can be seen in the altitude profile, the sinking is immediately lowered when 15 m is reached. The small sink rate is a result of the large Δu term. As this error goes toward zero, the sink rate returns to follow the Altitude Dependent Flare Law. Both simulations were done using a fully cooperative system, and so both the UAV and UGV corrects for differences in position. This shows that this method also has potential when a more advanced control structure is used, even when the acceleration phase might be different from what is assumed when calculating the UGV velocity.

9.3 Predictive Control

A well-studied area in the field of autonomous vehicles is Model predictive Control (MPC), a method for solving optimal control problems online. Formulating the problem for autonomous landing of the UAV on top of the UGV would require several different parts

- Trajectory generation
- Prediction model
- Objective function
- State and control constraints

which could all be found in several different ways.

Trajectory generation is used to generate reference values as inputs for the vehicle controllers. In MPC, an a priori known reference trajectory is made use of by taking future references into account in the current control input, thereby reducing delays in the process response [24]. Trajectories could be generated only once or be updated several times during the task. The trajectory should be both feasible and optimal according to some criteria. It also needs to account for the dynamics of both vehicles. Depending on the method of trajectory generation, this task can be computationally demanding, and so it would then be desirable not to do the generation more times than necessary.

Early on in the slope of the project that this thesis is a part of, the possibility of using pseudospectral collocation for trajectory generation was explored. Pseudospectral methods have proven to be useful in aerospace industries [25], and combines the numerical pseudospectral method for solving partial differential equations with an optimal control problem [26]. The collocation pseudospectral method solves the optimal control problem by discretizing it and converting it into a nonlinear program. Although this method presents a relatively simple way of generating optimal trajectories, it was rejected since it was considered more complex than needed at the time. There has also been issues with the onboard

computer having limited computational power, which would make this method difficult to implement.

The prediction model is used when evaluating how different sequences of control inputs affect the cost function. Ideally, the prediction model would completely imitate the real system, however, in reality both model deficiency and computational limitations make this impossible. Instead, a simpler model that captures the most relevant dynamics is used. Among the many variations of MPC there exist those capable of solving problems with nonlinear prediction models. Both vehicles in the landing control problem have nonlinear dynamics, and possible models to use for the prediction are a bicycle model for the ground vehicle movement, and a Dubins vehicle representing the fixed-wing dynamics [27].

Given that the mathematical model of the system is sufficiently precise, it will be possible to predict how the system evolves under different inputs. At each time step, a optimization problem is solved for a given horizon. The optimization consists of a cost function, which could look like

$$J = \sum_{j=1}^{N} \delta(j)[y(t+j) - w(t+j)]^2 + \sum_{j=1}^{M} \lambda(j)[u(t+j-1)]^2$$

where the first term indicates that deviations from a reference trajectory $(y-w)$ are penalized, as well as the control effort. Here the coefficients δ and λ are sequences that reflect how the future behavior is penalized. In practice this cost function can be defined in many other ways. The optimization is subject to a set of constraints such as

$$\begin{aligned} u_{min} &\leq u(t) &&\leq u_{max} \\ \dot{u}_{min} &\leq \dot{u}(t) &&\leq \dot{u}_{max} \\ y_{min} &\leq y(t) &&\leq y_{max} \end{aligned}$$

The possibility to add constraints is helpful in the landing control problem, since the limitations are taken into account already in the calculation of the control input.

The level of the model complexity and the objective function and constraints needs to be chosen with computational complexity in mind. A high fidelity model with a long prediction horizon will demand a lot of resources when it comes to each calculation. Since we are dealing with relatively fast dynamics, the next control decision needs to be available fast and so each computation cannot demand too much time.

As stated above, it was previously suggested and rejected for trajectory planning to be used during the early experimental phase of the project. The method does require a higher computational effort but nevertheless it presents a good alternative for further developments of the landing control. Trajectory generation is likely to give improved results since the paths can be planned in an optimal way.

9.4 Landing Prediction Adjusted γ

With the variable time constant flare in Equation (7.6), we were able to predict the length of the flare. It is possible to change the predicted flare length by choosing the time constant T_0 to fulfill

$$\begin{aligned} x_{des} &= T_0 V_{G0} \ln\left(\frac{h}{h_B} + 1\right) \\ \implies T_0 &= x_{des} \cdot \left(V_{G0} \ln\left(h/h_B + 1\right)\right)^{-1}. \end{aligned} \tag{9.4}$$

where x_{des} is some desired flare distance. This leads to the following descent law:

$$\dot{h} = -\frac{V_G \cdot \ln\left(h/h_B + 1\right)}{x_{des}}. \tag{9.5}$$

By adjusting the desired flare length x_{des} to be equal to the distance needed by the UAV before the vehicles are close enough to land, the UAV could land at the correct time and place.

This raises the question as to how x_{des} should be chosen. The prediction step described in Section 9.3 could be used to calculate an estimated distance. This could be done repeatedly throughout the landing as to determine if the landing is still going according to this prediction. In that sense, this method is a simplified version of the MPC approach where the optimization has been exchanged for the descent law of Equation (9.5). This make s this law less flexible in terms of optimization and constraints, however, it would be straightforward to implement it with the current system.

This approach is very close an automatic version of the way that the system was adjusted manually in between flights. If the system would not have enough time to converge one flight, then a fast way to improve this was to give it more time by decreasing the glide slope angle γ_0 or by changing one of the constants in the flare law. This approach would therefore be a very intuitive solution, aiming to automate the methodology used by the operators.

9.5 Future Control Efforts

Three out of many possible paths toward improving the control have been suggested in this chapter. Out of the three, the MPC one is the most complex but also the one that could be used in the most flexible way, by adjusting the objective function or the constraints to get control laws fulfilling certain objectives. Such an objective could for example be to land using minimum amount of runway distance.

The adaptive flare law in Equation (9.3) and the landing prediction flare law in Equation (9.5) are both modifications of the flare laws in Section 7.3. The difference to the old control laws is that these take the relative states of the vehicles into account, adjusting the descent to give the system more time to align themselves in the case of unexpected events. The reason why this is practical is that these laws in a simple way capture the essential constraints that are needed

for the final part of the landing - a decrease in sink rate and a decrease in the pitch angle. Both laws would be relatively straightforward to implement since they are simply extensions of what is already in use, and the computations that they require do not add any substantial load to what is already done.

The MPC approach would on the other hand require more effort for development and implementation into the current system. The computational load would increase considerably and the flight computer is already running on the limit of what it can handle, meaning that it probably would need to be updated first. Another reason is that it differs so much from the old system that it would first need to be tested extensively in order to prove that it is safe for flight. Nevertheless, this approach seems to be the most promising in terms of being a more intelligent and flexible solution.

Another possibility would be to adjust the state machine in order to change the behavior of the overall system. This could be done e.g. by changing the transition conditions to include also the relative velocities of the vehicles. This might improve the performance in that it could prevent the vehicles from going into undesirable states leading up to the retry mode, but it would not be possible to use it in order to for example optimize the trajectory as is possible with MPC.

Chapter 10

Conclusions

Cooperating controllers have been developed and tested for the UAV/UGV system. A PID approach was chosen as an initial method for several reasons. One such reason was due to constraints in time and computing power, which limited the complexity of what could be implemented. Another reason was that the main objective with these tests was to prove that the concept would work, and for that a controller of this type was sufficient. The successful landings during Flight Experiment 13 verify that it was indeed possible to land the fixed-wing UAV on top of a UGV autonomously, making the project a success in that its main objective was fulfilled.

The control design efforts were to a high degree focused on making the descent work in a proper way such that the UAV landed at a safe position. The controller for positioning the vehicles was tuned in flight, but not to a particularly high precision since other issues such as testing out the different flight modes were prioritized. There is therefore still room for improvement within this setup, though it might be of larger interest to investigate other control approaches now that a successful landing has been performed.

10.1 Landing Performance

The next step in developing the cooperative landing system is to consider how it needs to be extended in order to make it work within the final application. Landing of HALE UAVs is a very safety critical task, and their lightweight structures will be sensitive to wind disturbances. Furthermore, the landing should be feasible within some limited runway distance. This leads to three criteria to evaluate the performance by:

1. How well it avoids dangerous situations
2. How robust it is against external conditions
3. How effective it is in terms of required landing distance

These criteria are related in that external conditions such as heavy wind conditions cause the system into a dangerous state. To prevent this, the system takes some action, which is likely to increase the required landing distance.

In the end, landing was done with the state machine introduced in Section 7.3. This approach worked in some cases, but it could be argued that the overall performance and smoothness of the landing would improve if the system did not switch between different modes as much during each landing attempt. The development of the state machine was highly safety focused. This could be seen in the system behavior in that the safety mechanisms retry and go-around mode were reached on multiple occasions in each flight experiment. There was a choice to give a much higher priority to avoiding dangerous situations over making the controller optimized for robustness or a short landing distance in the criteria mentioned above. This was a natural priority to make in initial tests due to the complexity of the system. As the system is understood more deeply, focus should move so that all criteria get attention.

The switching behavior is desirable as long as it is used as a safety mechanism, e.g, against sudden winds or data connection losses. However, in many flights the retry was not caused by an unexpected change in the environment, but rather by something predictable such as an overshoot in position when there is a large velocity component, which was discussed in section 9.2. This could without a doubt be improved to reduce the frequency of the retry modes.

As it has now been established that this safety mechanism works well, focus can be shifted to develop more advanced control strategies to work together with this switching strategy.

10.2 Landing Strategy

The current landing strategy is performed in several separate parts, each with very little to no consideration to the performance of the other

- Longitudinal control
- Lateral control
- Sink rate
- Start of the acceleration phase
- Waypoint selection

In reality, the performance of these separate parts are all bound together in that the failure or bad performance of one part will affect the others. Examples of this were given in section 9.2, where it was shown that a slow convergence in the lateral control (e.g. due to wind or bad initial conditions) could make the decent benefit from also slowing down, to make the vehicle alignment catch up. Making the sink rate and acceleration phase more dependent on the starting distances and the performance of the longitudinal and lateral control could possibly improve the total performance of the system. Another type of coupling between these parts is that between the lateral and longitudinal control, which are decoupled only for small course angles, and that between velocity and sink

rate that is still coupled even though using TECS removes this dependence to a larger degree.

This coupling between the subtasks could be utilized to improve the controller, by letting all tasks to a higher degree be dependent on the state and performance of the rest of the system. For many cases these coupling effects will remain small and therefore it is acceptable to treat them separately. This is based on the assumption that the UAV is flying with a constant velocity, starting close to the runway centerline, and that the velocity of both vehicles remains close to the direction of the runway. When this is not the case, the effect for the landing might be that the UAV goes into go-around or retry modes, or that it requires a longer time or distance before it can land.

A worse performance might of course be expected when the external conditions are very bad, but the controller should still be able to handle these circumstances to some extent. This could be achieved either by improving the efficiency of the positioning control, or by letting the descent be more dependent on the states of the vehicles, as discussed in Section 9. A control strategy that makes decisions based on the entire state of the system instead of dividing the control problem up into smaller parts could improve the control and make the landing faster and more robust. It would also increase the system complexity and the computational workload. This type of method would also be more dependent on an accurate and frequent information exchange between the vehicles.

Of the control strategies investigated in Chapter 9, the MPC control strategy has a lot of potential for improving the control. Its natural way to set constraints on inputs and states could be used to limit the system to act safely. The objective function can be chosen in a way such that the landing is optimized for landing distance. This approach would however require a quite large effort in order to try it out on the real system.

Other methods of improving the performance include extending the state machine to include more situations, changing the flare law to depend on the entire state of the system, and adding features such as autonomous waypoint navigation and gain scheduling to decrease the influence of wind. These could all be incorporated relatively simply into the current system.

10.3 Future Work

The work with the autonomous landing of a UAV on top of a UGV will continue, and other suggested strategies for the control will be developed and tested. If more advanced controllers are going to be used, then parts of the system might need to be changed. There are already plans to change the flight computer, and this will facilitate more computationally expensive controllers to be used if needed.

Another future project for the group might be to further develop the landing platform. If the platform was made controllable, it might be possible for the aircraft to land in more angles, which could perhaps change the way that the final phase of the landing is done.

Finally, since the intention is to have a completely autonomous UGV, one further development will be to build this completely autonomous system. Such a system could potentially handle more advanced control instructions than a human could, and would have a faster response time and less irrational behavior since a human might unintentionally add inputs that are not supposed to be there. With the current controller, complex maneuvers were intentionally avoided in order to simplify the landing attempts and decrease the stress level for the driver. More complex driving might be added in later experiments. A completely autonomous system could also add more possibilities in terms of what controls can be tried out successfully.

Appendix A

Flight Data

A.1 Acceleration Phase

Attempt	Start x [m]	Start y [m]	End x [m]	End y [m]	Length [m]
13.1	118.20	2.94	-6.54	0.73	124.74
13.2	115.82	2.37	5.59	0.83	110.23
13.3	115.89	1.49	3.04	1.23	112.86
13.4	128.63	0.85	3.35	0.25	125.27
13.5	106.76	-0.72	-3.99	-1.59	110.76
13.6	138.45	32.52	-12.04	-2.18	150.49
13.7	131.31	42.01	6.56	-3.60	124.76
13.8	142.19	42.25	24.41	-3.56	117.78
13.9	121.90	-3.05	-0.13	0.17	122.02
13.10	111.80	0.10	2.77	0.14	109.03
13.11	90.99	-1.05	1.87	0.04	89.11
13.12	84.30	0.33	4.63	0.52	79.67
13.13	82.54	-0.70	4.50	0.46	78.04
13.14	80.41	0.10	6.88	0.65	73.52

Table A.1: Acceleration data from Flight Experiment 13. Attempt 1-10 had a velocity of around 21 m/s, and 11-14 had a velocity of 19 m/s.

A.2 Initial Values

Attempt	h [m]	Δx [m]	Δy [m]	V_k [m/s]	χ [deg]
9.1	17.71	172.41	−66.19	25.59	−6.6
9.2	18.59	306.16	−54.51	23.78	−3.0
9.3	16.45	332.31	−55.17	24.65	−0.6
9.4	17.08	171.46	−54.99	25.97	−5.9
12.1	28.82	225.50	2.59	24.32	1.5
12.2	28.55	152.84	27.31	22.76	3.3
12.3	28.17	217.74	13.40	24.17	−3.8
12.4	27.84	173.78	5.14	23.06	1.6
12.8	25.90	237.34	−0.92	21.49	4.1
12.10	25.42	201.74	4.67	23.96	2.9
13.1	24.17	303.39	−22.10	23.09	−14.4
13.2	24.10	301.39	−19.59	23.62	−9.8
13.3	24.50	275.18	−17.56	23.45	−15.3
13.4	24.19	309.02	−11.68	24.48	−4.3
13.5	24.19	181.81	14.95	23.70	−9.4
13.6	24.21	144.30	32.56	23.47	−1.9
13.7	24.29	113.59	42.48	23.58	1.5
13.8	24.17	127.90	42.49	23.78	1.7
13.9	24.17	180.71	−3.67	23.34	4.0
13.10	24.21	255.09	−8.02	22.97	−18.7
13.11	24.21	224.80	−0.69	20.70	−3.5
13.12	24.34	273.35	−8.19	20.16	−10.6
13.13	24.13	245.54	−3.07	20.09	−7.4
13.14	24.30	293.33	−10.47	20.04	−9.4

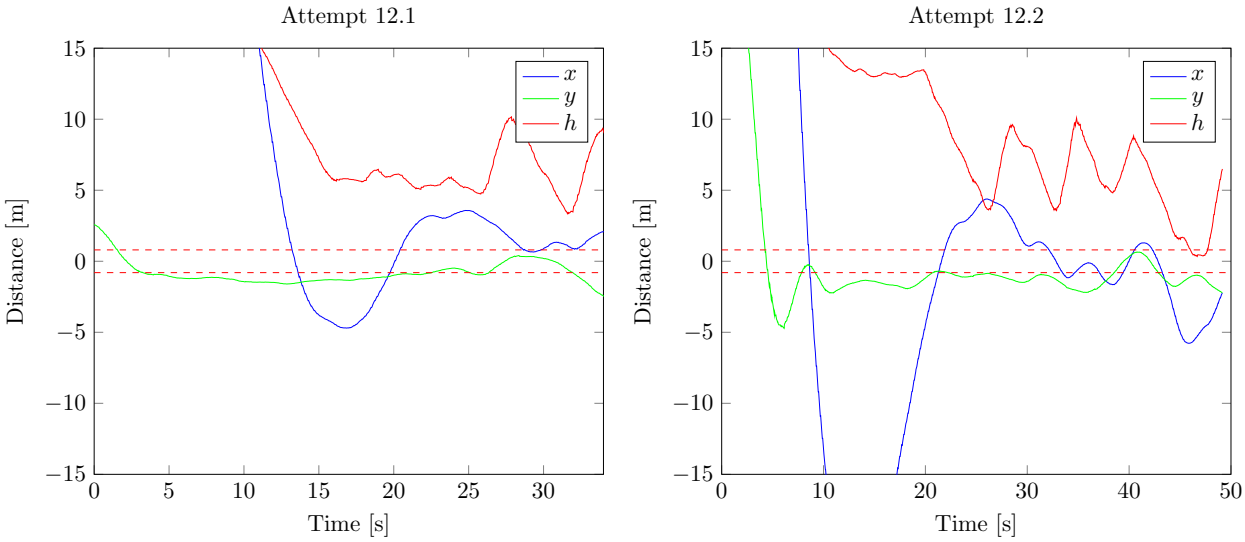
Table A.2: The initial values are dependent on the waypoints, the environment and when the operator chooses to switch to landing mode.

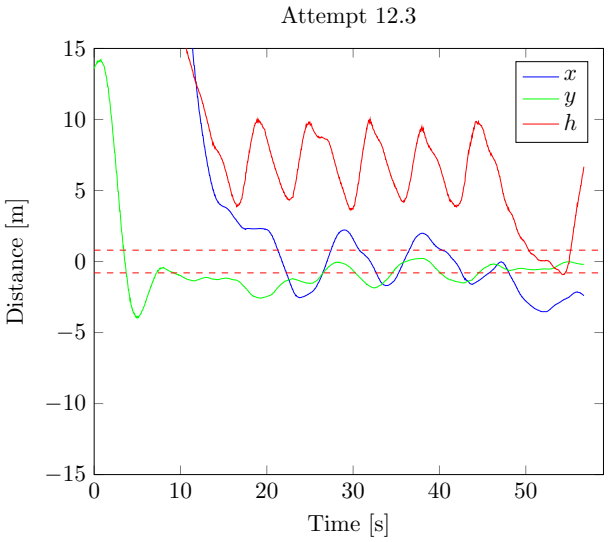
Flight Experiment: 12 Date: 8/12-2015

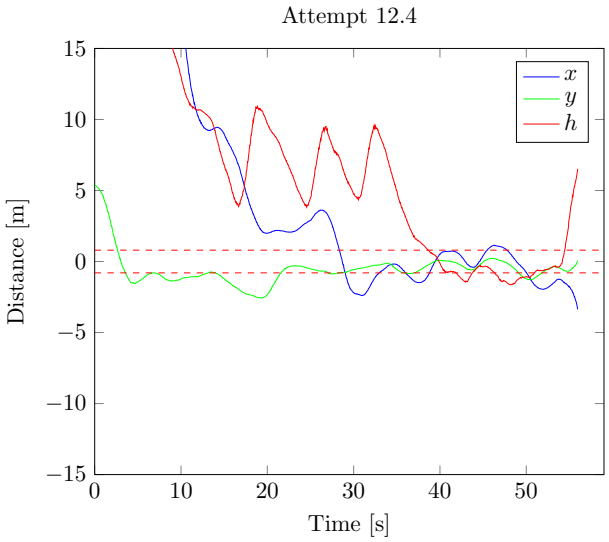
A.3 Position Control Performance

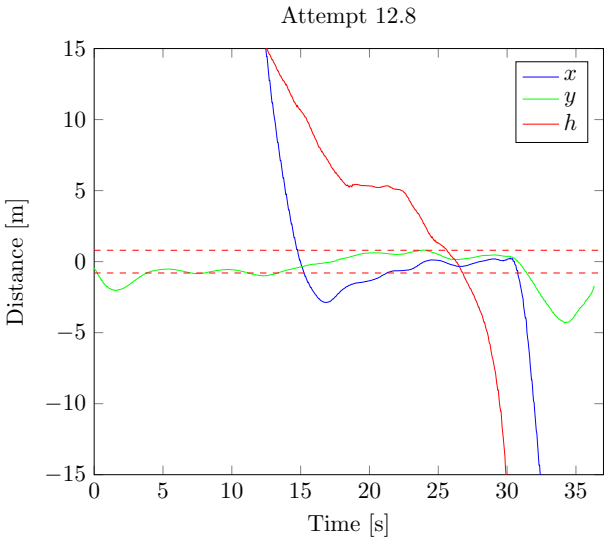
Notes: When in waypoint mode, the UAV lost connection when far away, which is why many of the later flights failed. In landing attempt 2-4, oscillations in x and h were observed as the system went in and out of go around. In attempt 12.8, retard and ground lock modes were successfully tested.

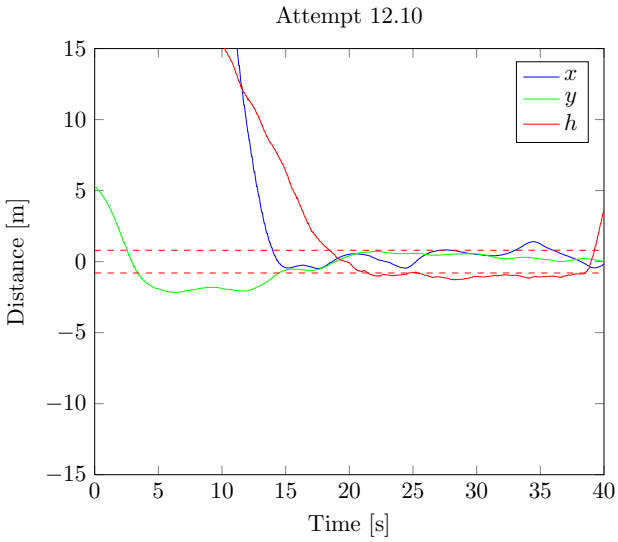
Attempt	Virtual runway	Lateral	Longitudinal	V_{land}	γ_0
12.1	50 m	UAV: PID	UGV: P, UAV: P	21 m/s	5°
12.2	50 m	UAV: PID	UGV: P	21 m/s	5°
12.3	50 m	UAV: PID	UGV: P, UAV: P	21 m/s	5°
12.4	50 m	UAV: PID	UGV: P, UAV: P	21 m/s	5°
12.8	80 m	UAV: PID	UGV: P	21 m/s	5°
12.10	10 m	UAV: PID	UGV: P	21 m/s	5°




Attempt 12.3
Distance [m]
Time [s]
x
y
h


Attempt 12.4
Distance [m]
Time [s]
x
y
h

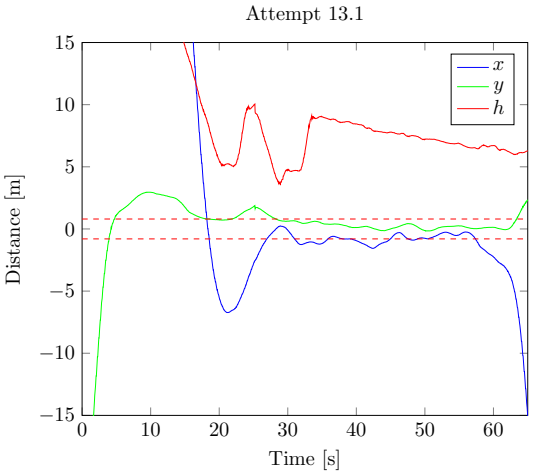

Attempt 12.8
Distance [m]
Time [s]
x
y
h

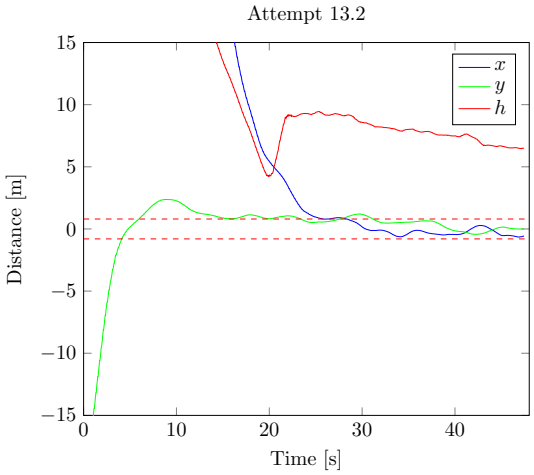

Attempt 12.10
Distance [m]
Time [s]
x
y
h

Flight Experiment: 13 Date: 14/12-2015

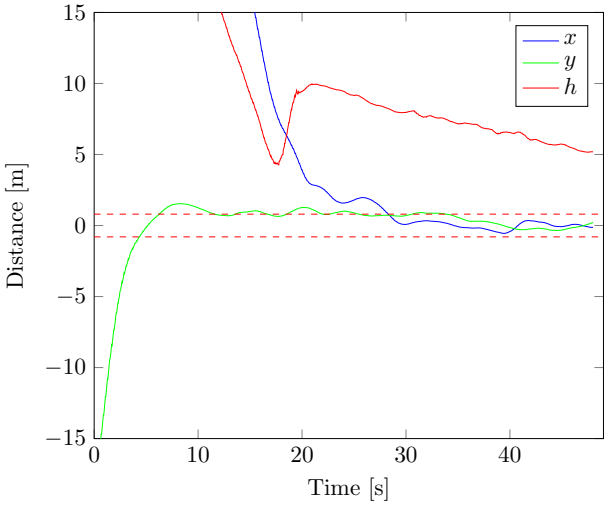
Notes: First landing at attempt 13.10, second succesfull landing at attempt 13.14. The first three attempts had an error in the flare, making it go very slow. The first four flights had an initial descent angle of 5 degrees, which was then lowered to 3 degrees to give the vehicles more time to align. Go-around was initiated both because of the vehicles being too far away and because of time lag.

Attempt	Virtual runway	Lateral	Longitudinal	V_{land}	γ_0
13.1	80 m	UAV: PI	UGV: PD	Groundspeed	5°
13.2	80 m	UAV: PI	UGV: PD	Groundspeed	5°
13.3	80 m	UAV: PI	UGV: PD	Groundspeed	5°
13.4	80 m	UAV: PI	UGV: PD	Groundspeed	5°
13.5	5 m	UAV: PI	UGV: PD	Groundspeed	3°
13.6	2 m	UAV: PI	UGV: PD	Groundspeed	3°
13.7	1.5 m	UAV: PI	UGV: PD	Groundspeed	3°
13.8	0.75 m	UAV: PI	UGV: PD	Groundspeed	3°
13.9	0.75 m	UAV: PI	UGV: PD	Groundspeed	3°
13.10	0 m	UAV: PI	UGV: PD	Groundspeed	3°
13.11	0 m	UAV: PI	UGV: PD	Groundspeed	3°
13.12	0 m	UAV: PI	UGV: PD	Groundspeed	3°
13.13	0 m	UAV: PI	UGV: PD	Groundspeed	3°
13.14	0 m	UAV: PI	UGV: PD	Groundspeed	3°

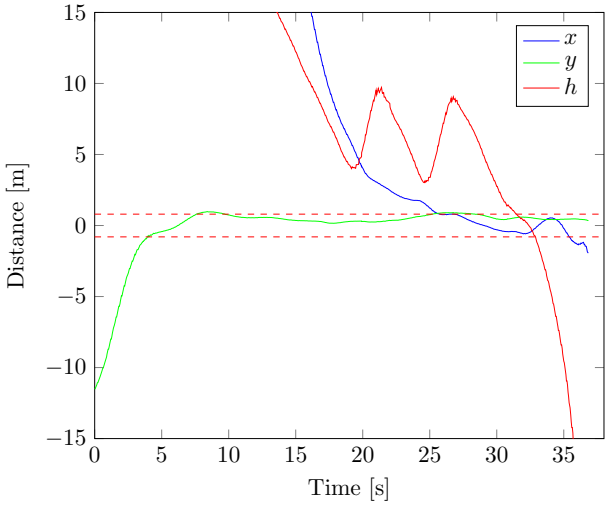




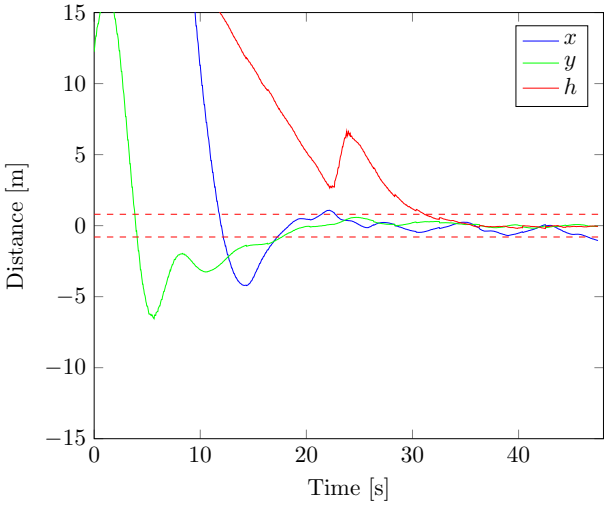
Attempt 13.3



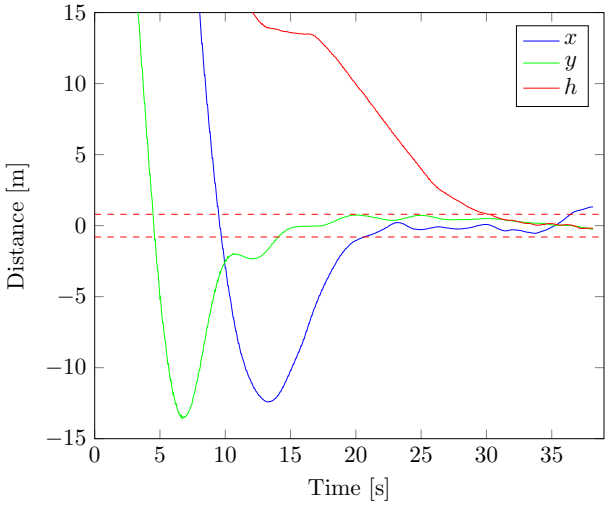
Attempt 13.4



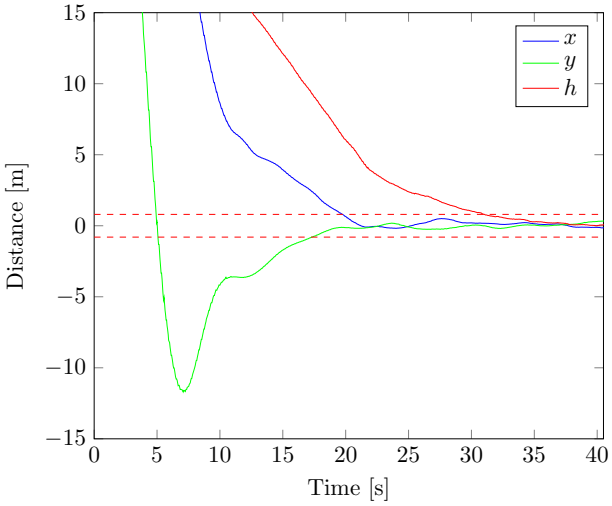
Attempt 13.5



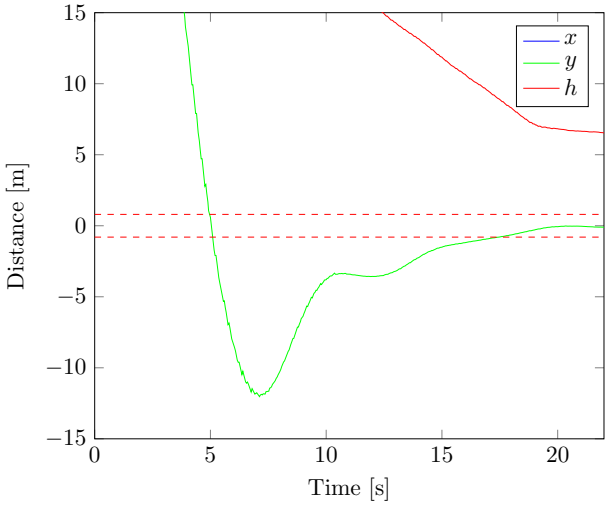
Attempt 13.6



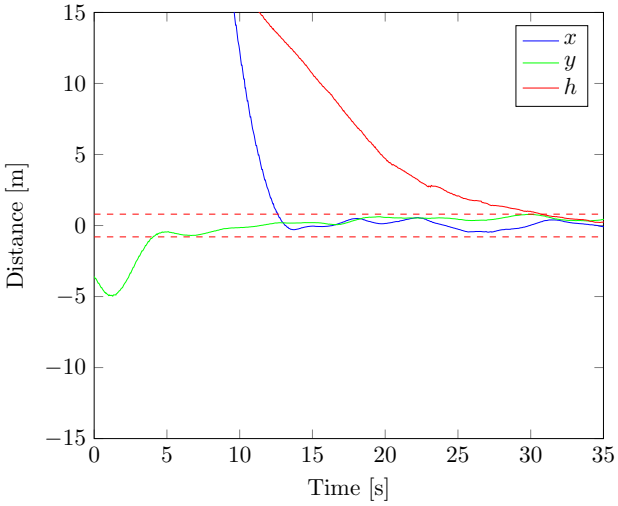
Attempt 13.7



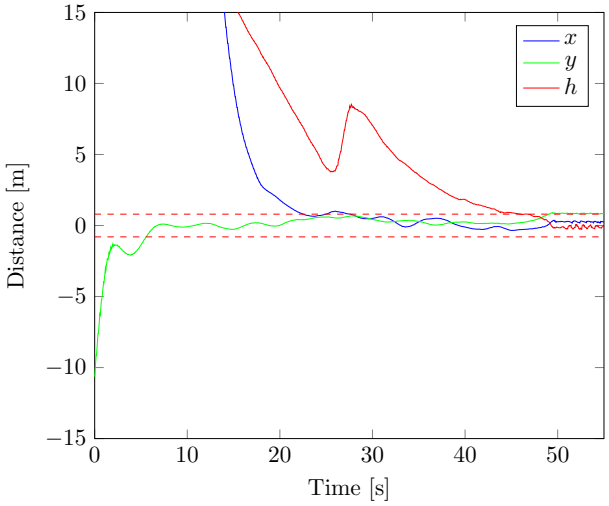
Attempt 13.8



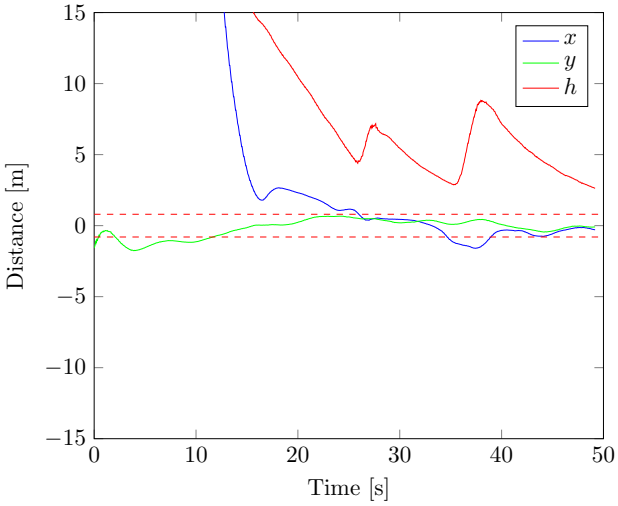
Attempt 13.9



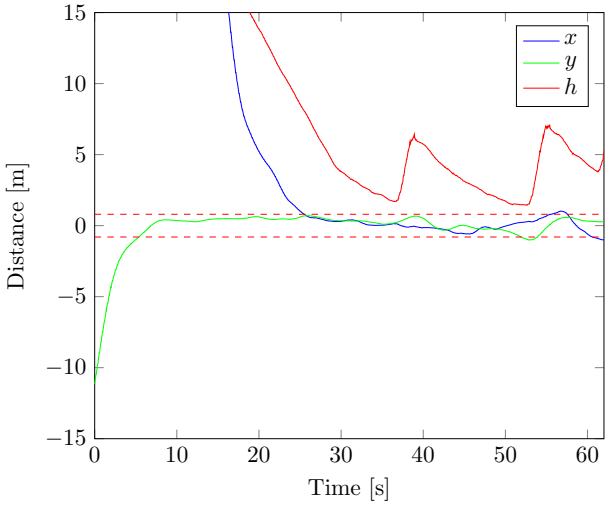
Attempt 13.10



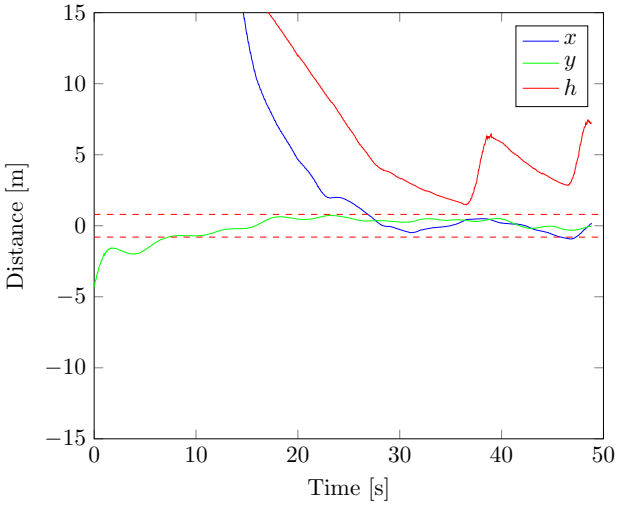
Attempt 13.11



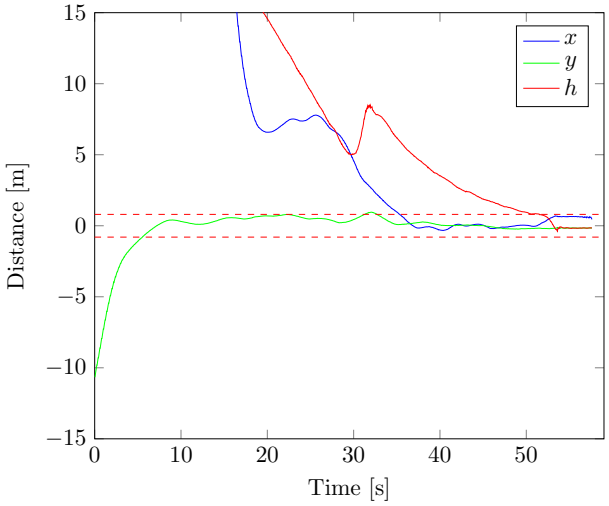
Attempt 13.12



Attempt 13.13



Attempt 13.14



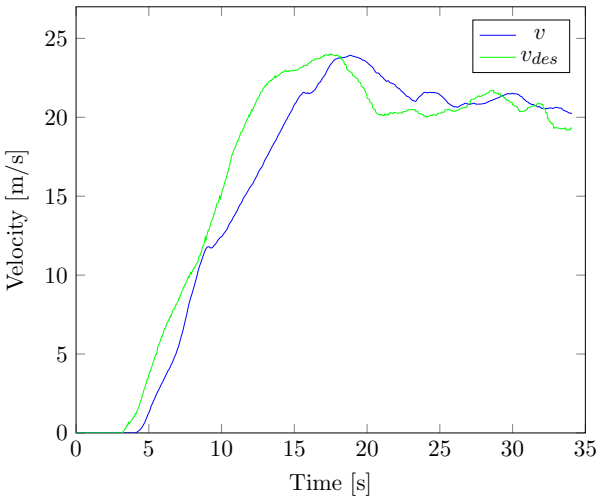
Appendix B

Performance of Human Actuator

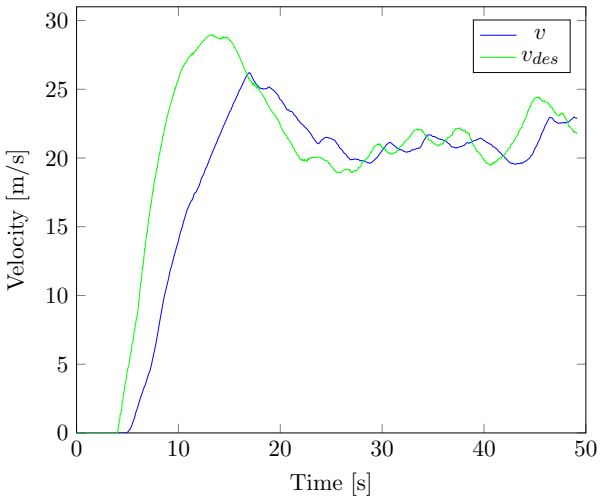
The reaction time seems to be 0.5-1 seconds. This gives a delay in addition to the delay in information exchange.

The performance of the driver was often satisfactory, only occasionally were there any large errors. The desired vehicle dynamics was smooth in order to make it simpler to follow the commands. Sometimes this failed due to reason mentioned in Section 8.2.

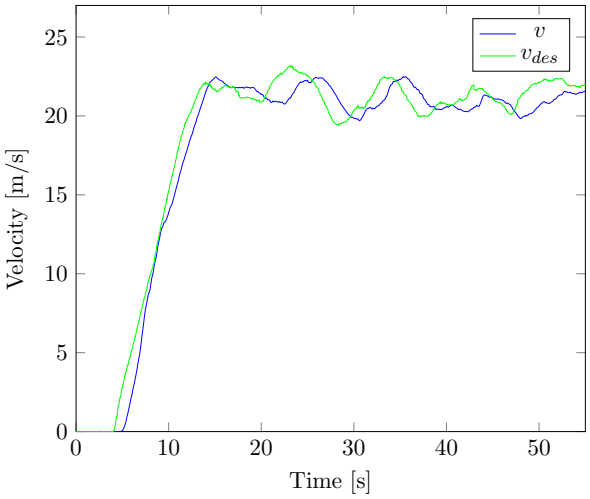
Attempt 12.1



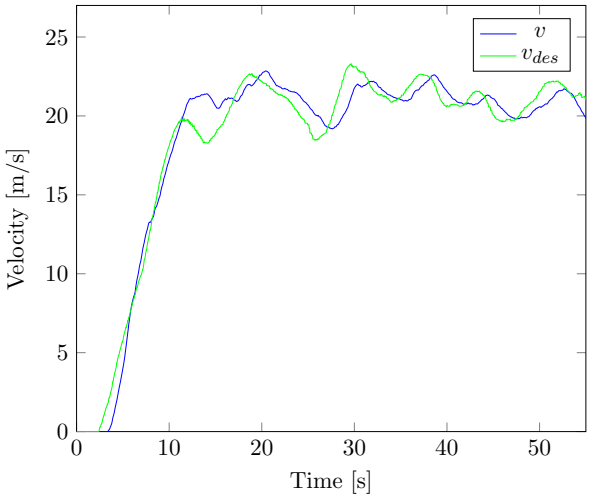
Attempt 12.2



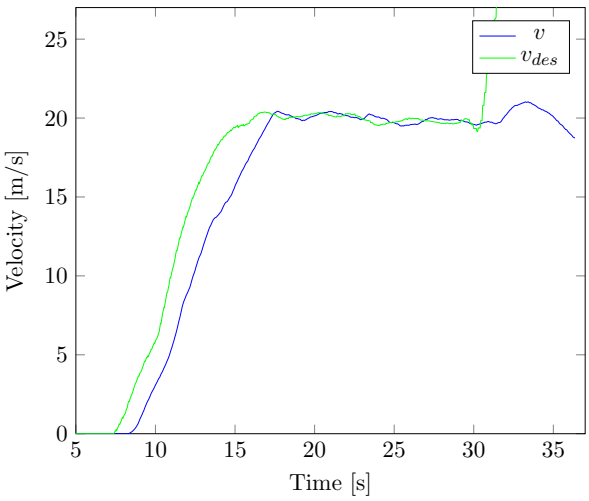
Attempt 12.3



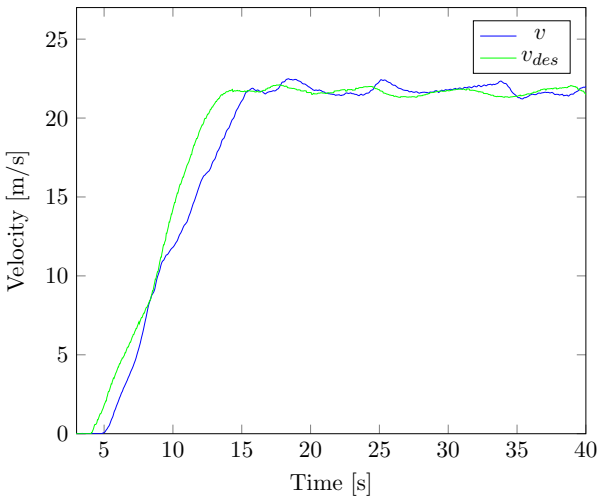
Attempt 12.4



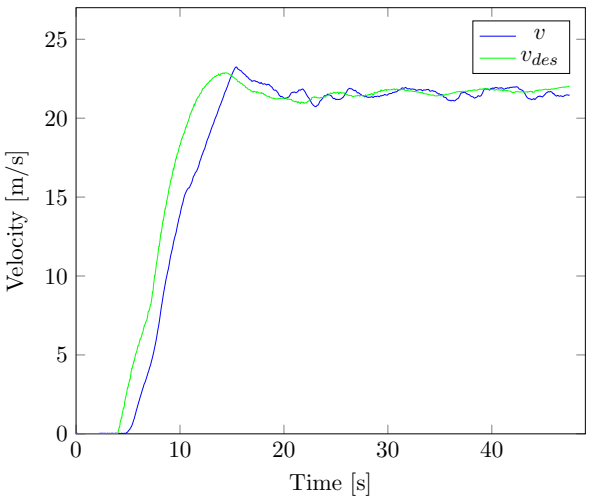
Attempt 12.8



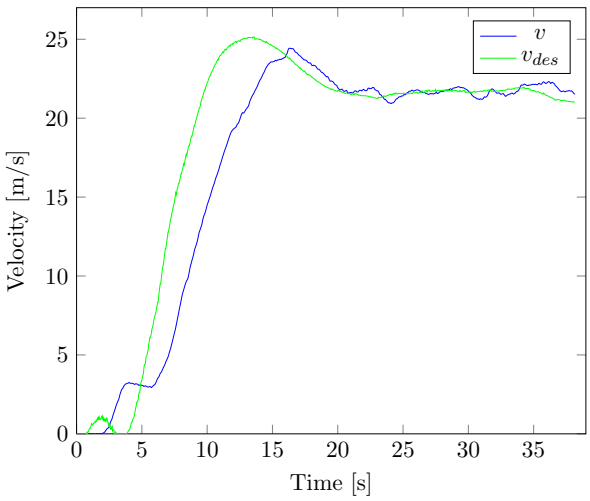
Attempt 12.10



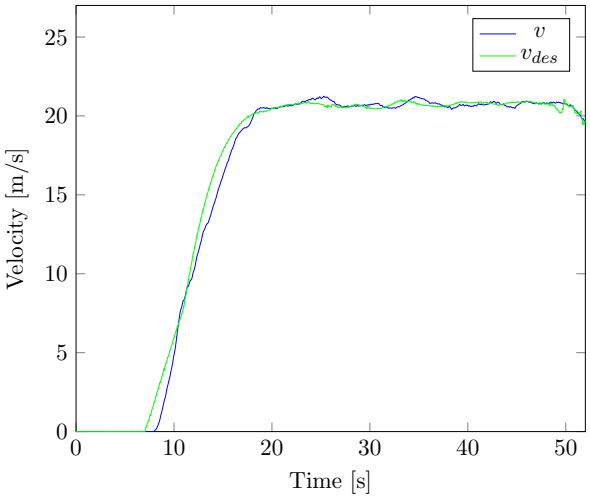
Attempt 13.5



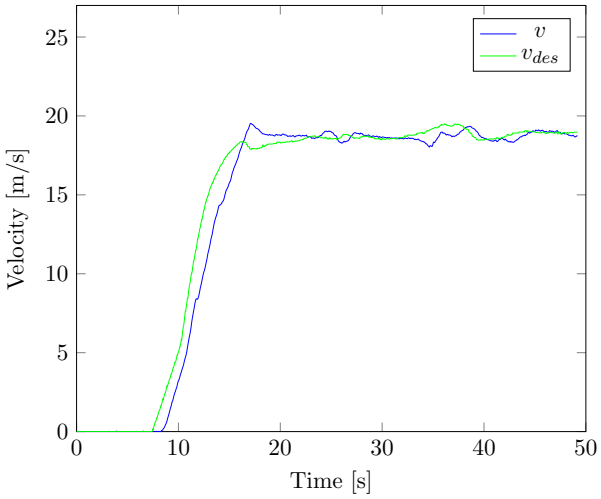
Attempt 13.6



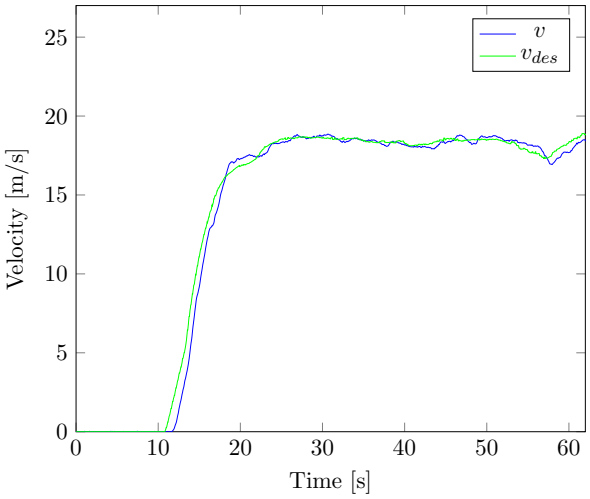
Attempt 13.10



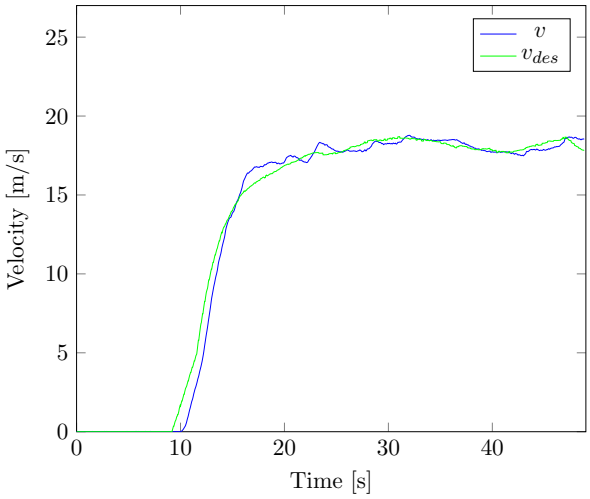
Attempt 13.11



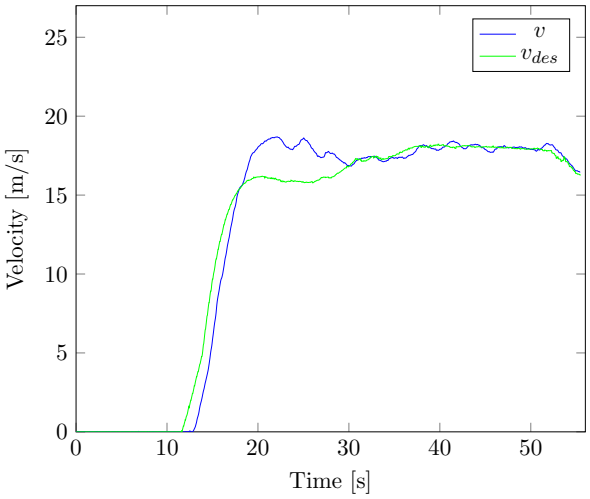
Attempt 13.12



Attempt 13.13



Attempt 13.14



Appendix C

UGV Model

The model for the UGV used in the feedback analysis was

$$\frac{d}{dt}\begin{bmatrix} u \\ x \\ y \\ a \\ \theta \end{bmatrix} = \begin{bmatrix} -0.1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 \\ -2.5 & 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} u \\ x \\ y \\ a \\ \theta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 2.5 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} v^{cmd} \\ \theta^{cmd} \end{bmatrix}$$

Bibliography

[1] A. Noth. (2008) History of solar flight. [Online]. Available: http://www.sky-sailor.ethz.ch/docs/History_of_Solar_Flight_v1.2-A.Noth_2006.pdf

[2] (2015) Nasa armstrong fact sheet: Helios prototype. [Online]. Available: http://www.nasa.gov/centers/armstrong/news/FactSheets/FS-068-DFRC.html

[3] Zephyr solar-powered hale uav. [Online]. Available: http://www.airforce-technology.com/projects/zephyr/

[4] Solar powered high altitude and long endurance platforms. [Online]. Available: http://www.dlr.de/rmc/rm/desktopdefault.aspx/tabid-7660/13033_read-32819/

[5] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 3, 2001, pp. 2968–2973 vol.3.

[6] S. McCamish, M. Romano, and X. Yun, "Autonomous distributed control of simultaneous multiple spacecraft proximity maneuvers," *Automation Science and Engineering, IEEE Transactions on*, vol. 7, no. 3, pp. 630–644, July 2010.

[7] L. Sun and W. Huo, "Robust adaptive control for spacecraft cooperative rendezvous and docking," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec 2013, pp. 5516–5521.

[8] L. Sandino, M. Bejar, and A. Ollero, "On the applicability of linear control techniques for autonomous landing of helicopters on the deck of a ship," in *Mechatronics (ICM), 2011 IEEE International Conference on*, April 2011, pp. 363–368.

[9] J. Hervas, M. Reyhanoglu, and H. Tang, "Automatic landing control of unmanned aerial vehicles on moving platforms," in *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, June 2014, pp. 69–74.

[10] D. Rohacs, M. Voskuijl, and N. Siepenktter, "Evaluation of landing characteristics achieved by simulations and flight tests on a small-scaled model

related to magnetically levitated advanced take-off and landing operations," Tech. Rep., 2014.

[11] M. Fravolini, A. Ficola, M. Napolitano, G. Campa, and M. Perhinschi, "Development of modelling and control tools for aerial refueling for uavs," in *AIAA Guidance, Navigation, and Control Conference and Exhibit.*

[12] G. Campa, M. Fravolini, A. Ficola, M. Napolitano, B. Seanor, and M. Perhinschi, "Autonomous aerial refueling for uavs using a combined gps-machine vision guidance," in *AIAA Guidance, Navigation, and Control Conference and Exhibit.*

[13] J. D. M. Pachter and J. L. Dargan, "Automatic formation flight control," *Journal of Guidance, Control, and Dynamics*, 1994.

[14] E. Lavretsky, "F/a-18 autonomous formation flight control system design," in *AIAA Guidance, Navigation, and Control Conference and Exhibit.*

[15] L. Buzogany, M. Pachter, and J. D'Azzo, "Automated control of aircraft in formation flight," in *Proceedings of the AIAA Guidance, Navigation and Control Conference.*

[16] G. Balmer, "Modelling and control of a fixed-wing uav for landings on a mobile landing platform," Master's thesis, KTH Royal Institute of Technology, 2015.

[17] A. A. Lambregts, "Vertical flight path and speed control autopilot using total energy principles," *Boeing*, 1983.

[18] W. Dunbar and R. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, 2006.

[19] J. Tisdale, Z. Kim, and J. Hedrick, "Autonomous uav path planning and estimation," *Robotics Automation Magazine, IEEE*, vol. 16, no. 2, pp. 35–42, June 2009.

[20] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of uav 3d path planning," in *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, June 2014, pp. 2376–2381.

[21] P. Sujit, S. Saripalli, and J. Borges Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *Control Systems, IEEE*, vol. 34, no. 1, pp. 42–59, Feb 2014.

[22] A. A. Lambregts and J. F. Creedon, "Development and flight evaluation of automatic flare laws with improved touchdown dispersion," *Guidance and Control Conference*, July 1980.

[23] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: A short survey of recent results," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, June 2005, pp. 24–29.

[24] *Model Predictive Control*. Springer-Verlag London, 2007.

[25] W. Kang and N. Bedrossian, "Pseudospectral optimal control theory makes debut flight, saves nasa $1m in under three hours," 2007.

[26] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: From theory to flight," *Annual Reviews in Control*, vol. 36, no. 2, pp. 182 – 197, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578812000375

[27] M. Hwangbo, J. Kuffner, and T. Kanade, "Efficient two-phase 3d motion planning for small fixed-wing uavs," in *Robotics and Automation, 2007 IEEE International Conference on*, April 2007, pp. 1035–1041.