Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for
Dynamic Systems and Control

Institut für Dynamische Systeme
und Regelungstechnik

Philippe Petit

Integrating fixed wing UAVs in the FMA

Semester Thesis

Institute for Dynamic Systems and Control
Swiss Federal Institute of Technology (ETH) Zurich

Supervision

Mark Müller
Prof. Dr. D'Andrea

March 2012

IDSC-RD-MM-03

Acknowledgments

Firstly I want to thank Mark Müller for his hosting this thesis. His continuous support as well as his unlimited patience for all my questions and request, solving my problems or helping me with another test where invaluable. Without him, this thesis would literally not have been possible!
I want to also thank Marc-Andre Corzillius for helping me with the electronics and hardware of the testbed as well as for his advice. Last but not least my thanks goes to Prof. Raffaello D'Andrea.

Contents

Abstract

This thesis treats the development of a controller for stabilizing a fixed wing shockflyer in vertical hover position inside the Flying Machine Arena.
A set of ordinary differential equations is acquired by using a six degrees of freedom rigid body assumption together with an aerodynamic model describing the forces of a flat plate in laminar and turbulent condition, taken from [1].
In a C++ simulation, a cascaded PD controller was validated for the developed set of ordinary differential equations.
Finally, real life flight test showed the ability of the controller to stabilize the aircraft inside the FMA in hover position, by using Vicon data as position and attitude feedback. Control over attitude and position while hovering where successfully demonstrated.

Chapter 1

Introduction

Shockflyers are small model aircrafts which are characterized by a high trust to weight ratio and a low mass. These characteristics enable this type of fixed wing aircrafts to perform almost surreal aerobatics like extremely tight flown turns, loopings and rolls as well as a maneuver called the prop hang, in which the aircraft hovers vertically, “hanging“ on the propeller in mid air. Many videos can be found in the internet which show the full capability of these aircraft if flown by a skilled pilot.
For a control engineer, controlling such a system in a confined airspace like the Flying Machine Arena, poses a strong challenge due to the narrow room constraints and the fast and agile dynamics of the aircraft. Especially complex is the description of the aerodynamics, as the flow over the wings is detached and turbulent for many flight conditions. As this thesis is a first step in the integration of fixed wing aircrafts into the FMA, it was decided to start off with the least complex maneuver such an aircraft can fly: The prop hang maneuver or vertical hover. However it should be noted that ultimately only more complex maneuvers can show the stunning aerobatic capabilities of the system!

Figure 1.1: A typical shockflyer, with widespread use in hobby applications

Chapter 2

3D modeling

For simulation and control of the aircraft in three dimension we need the equations of motion in three dimensions. These equations will be derived in the following:

2.1 Coordinate System

First a sketch of the airplane with the coordinate system:

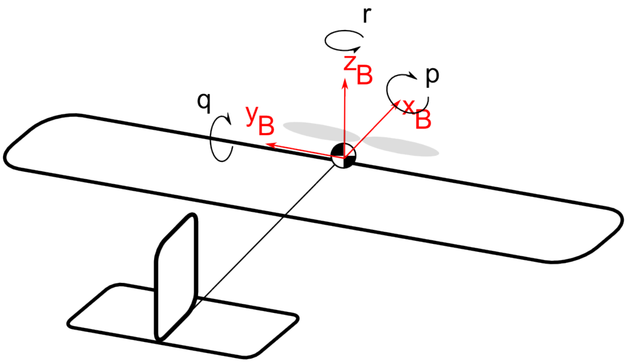


Figure 2.1: Sketch of the 3D coordinate System

The sketch for the x-z plane:

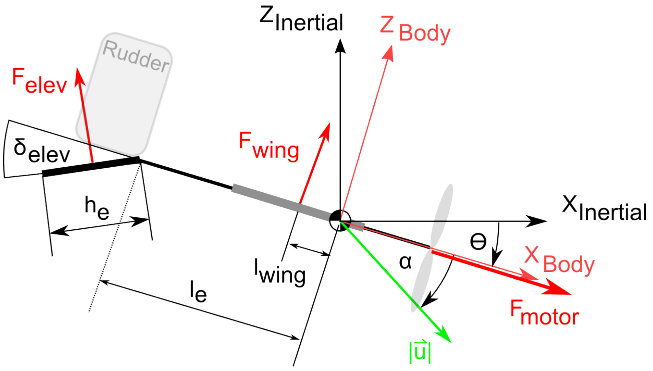


Figure 2.2: Forces and moments in the x-z plane

The x-y plane:

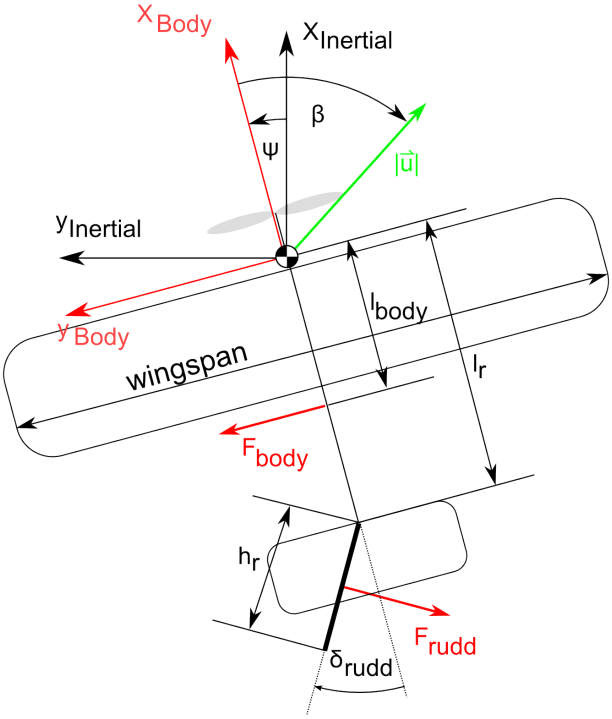


Figure 2.3: Forces and moments in the x-y plane

And finally the y-z plane:

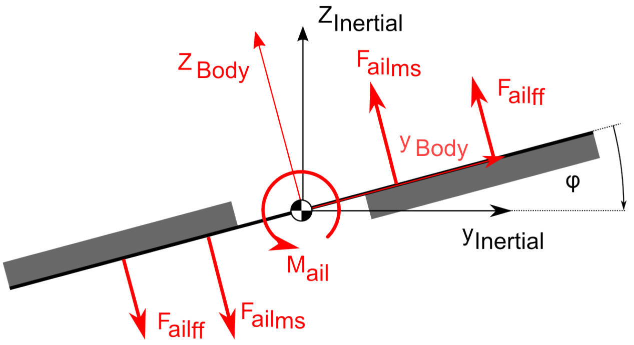


Figure 2.4: Forces and moments in the y-z plane

2.2 Equations of Motion

In the following, I will assume that the aircraft can be described by a 6 degrees of freedom, rigid body assumption with a diagonal inertia matrix. However it should be noted that the diagonal inertia matrix assumption is purely made, because it simplifies the equations of motion though the aircraft is not symetric.

$$I_B = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \tag{2.1}$$

Thus the equations of motion for a rigid 6 degrees of freedom body [4, 6]:

$$\ddot{\vec{x}}_I = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \frac{1}{m} \cdot \begin{pmatrix} F_{x_I} \\ F_{y_I} \\ F_{z_I} \end{pmatrix} + \vec{G} = \frac{1}{m} \cdot T \cdot \begin{pmatrix} F_{x_B} \\ F_{y_B} \\ F_{z_B} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \tag{2.2}$$

$$\frac{\delta \vec{\omega}}{\delta t} = \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = I_B^{-1} \cdot (M_{tot} - \vec{\omega} \times (I_B \cdot \vec{\omega}) = \begin{pmatrix} (M_{x_B} - q \cdot r \cdot (I_{zz} - I_{yy}))/I_{xx} \\ (M_{y_B} - r \cdot p \cdot (I_{xx} - I_{zz}))/I_{yy} \\ (M_{z_B} - p \cdot q \cdot (I_{yy} - I_{xx}))/I_{zz} \end{pmatrix} \tag{2.3}$$

As well as the definitions for the angle of attack α, the sideslip angle β and absolute air velocities $|\vec{u}|$ and $|\vec{v}|$

$$\alpha = -\tan^{-}1(\frac{w}{u}) \tag{2.4}$$

$$\beta = -\tan^{-}1(\frac{v}{u}) \tag{2.5}$$

$$|\vec{u}| = \sqrt{u^2 + w^2} \tag{2.6}$$

$$|\vec{v}| = \sqrt{u^2 + v^2} \tag{2.7}$$

2.2.1 Forces

All forces necessary to describe the aircraft will be derived in this section. The forces acting upon the airplane in x-, y- and z-direction, as indicated in figures

2.2, 2.3 and 2.4:

$$\begin{aligned} F_{x_B} =& F_{motor} - F_{elev} \cdot \sin(\delta_{elev}) - F_{rudd} \cdot sin(\delta_{rudd}) - \\ & 2 \cdot F_{ail} \cdot \sin(\frac{\delta_{ail}}{2}) \end{aligned} \tag{2.8}$$

$$F_{y_B} = F_{body} - F_{rudd} \cdot \cos(\delta_{rudd}) \tag{2.9}$$

$$F_{z_B} = F_{wing} + F_{elev} \cdot \cos(\delta_{elev}) \tag{2.10}$$

Note that a list of acronyms can be found at the end of this chapter in table 2.1.

2.2.2 Motor force

The motor is providing the energy needed to propel the system.
In my thesis I assume that the force is a linear function of the motor input:

$$F_{motor} = u_M \cdot F_{max} \qquad u_M \in [0;1] \tag{2.11}$$

However, the motor does not only provide a force, but also creates a streamtube which is incidenting upon the different control surfaces:

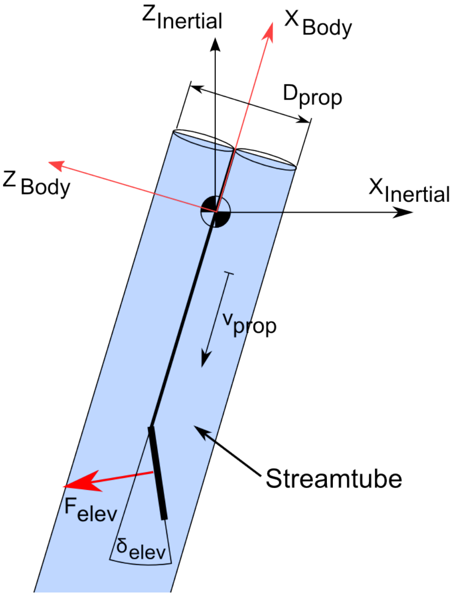


Figure 2.5: Illustration for the streamtube generated by the motor

This streamtube in combination with the deflected control surfaces, will create a force normal to the flat plate of the actuator, as shown exemplary in figure 2.5 for the elevator surface.
For the following calculations, I assume that the diameter of the streamtube stays constant over its length. Also it is assume that the velocity profile of the streamtube is constant throughout its diameter.
The velocity of the streamtube generated by the motor can be calculated over

the force generated by the motor and the conservation of momentum law from fluid dynamics [5, S 87, S 95].

$$F_{thrust} = \rho \cdot v_{prop}^2 \cdot A_{prop} \tag{2.12}$$

$$v_{prop} = \sqrt{\frac{u_M \cdot F_{max}}{\rho \cdot A_{prop}}} \tag{2.13}$$

Note that this streamtube is not sketched in figures 2.2 to 2.4, though it acts upon every control surface.

2.2.3 Aerodynamic forces

The corner stone of the aerodynamic description, is the assumption that the force generated by the various flat plate surfaces can be described over the equation [1]:

$$F_N = \rho \cdot \frac{v^2}{2} \cdot A \cdot c_N \tag{2.14}$$

$$c_N = 2 \cdot \sin(\delta) \tag{2.15}$$

With the following variables:

F_N	Force generated by flat plate in Normal direction	$[N]$
ρ	Density of air	$[\frac{Kg}{m^2}]$
v	Velocity of air	$[\frac{m}{s}]$
A	Flat plate surface area	$[m^2]$
c_N	Dimensionless force coefficient	$[]$

This description holds true for flat plate surfaces and for all angles δ, including those beyond stall as shown in [1].
I will adapt this description throughout the thesis, as our aircraft solely consists of flat plates as can be seen in figure 3.1.
Two airstreams are incidenting upon the different control surfaces: The freestream component and the streamtube generated by the motor. For the elevator force this yields together with 2.13, 2.14 and 2.15:

$$\begin{aligned} F_{elev} =& F_{freestream} + F_{MotorStreamtube} \\ =& \rho \cdot A_{elev} \cdot |\vec{u}| \cdot \sin(\delta_{elev} + \alpha) + \\ & \frac{h_e \cdot D_{prop} \cdot F_{max}}{A_{prop}} \cdot u_M \cdot \sin(\delta_{elev}) \end{aligned} \qquad \delta_{elev} \in [-45°, 45°] \tag{2.16}$$

The force in x-direction is maximal at $\delta_{elev} = 45°$, which is the reason for the limits in 2.16. The maximum deflections on the real aircraft where adjusted accordingly. The surface of the elevator for the term $F_{MotorStreamtube}$, was calculated over the chord of the elevator times the diameter of the motor streamtube: $A_{elevStreamtube} = h_e \cdot D_{prop}$
Likewise for the rudder:

$$\begin{aligned} F_{rudd} =& \rho \cdot A_{rudd} \cdot |\vec{v}| \cdot \sin(\delta_{rudd} - \beta) + \\ & \frac{h_r \cdot D_{prop} \cdot F_{max}}{A_{prop}} \cdot u_M \cdot \sin(\delta_{rudd}) \end{aligned} \qquad \delta_{rudd} \in [-45°, 45°] \tag{2.17}$$

To calculate the force generated by the main wing and the body of the aircraft, the motor streamtube can be neglected, as it has zero incident angle upon these surfaces. The emerging forces for the Wing and body and the sidewards body of the aircraft:

$$F_{wing} = \rho \cdot A_{wing} \cdot |\vec{u}| \cdot \sin(\alpha) \tag{2.18}$$

$$F_{body} = \rho \cdot A_{body} \cdot |\vec{v}| \cdot \sin(\beta) \tag{2.19}$$

Aileron Deflection

The deflected ailerons do not have the shape of flat plates, therefore equation 2.15 cannot be applied anymore. However, to describe the forces created by the deflected aileron and due to the lack of better alternatives, I choose the following assumption: the deflected aileron can be described as a flat plate which spans from the leading edge to trailing edge of the aileron. The resulting deflection angle is $\frac{\delta_{ail}}{2}$. This assumption is illustrated in 2.6.

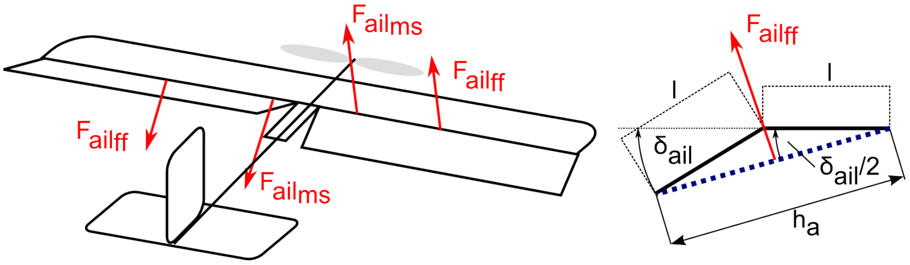


Figure 2.6: Illustration for the deflection of the aileron

The aileron force itself is again split up into the force generated by the motor streamtube, marked with the index "ms", while the force arising from the freestream is marked with "ff". This is necessary due to the different moments generated by the two force components.
Also note that I split up the force of the wing into the force generated by the ailerons and the force of the wing without the ailerons. This might seem cumbersome, however it simplifies some calculations and the debugging process in simulation.

$$F_{ail} = (F_{ail_{ff}} + F_{ail_{ms}}) \cdot \sin(\frac{\delta_{ail}}{2}) = \tag{2.20}$$

$$= \frac{1}{2} \cdot \rho \cdot A_{wing} \cdot u^2 \cdot \sin(\frac{\delta_{ail}}{2}) + \frac{1}{2} \cdot \frac{h_a \cdot D_{prop} \cdot F_{max}}{A_{prop}} \cdot u_M \cdot \sin(\frac{\delta_{ail}}{2}) \qquad \delta_{ail} \in [-45°, 45°] \tag{2.21}$$

Thus inserting into 2.8 to 2.10, we get:

$$
\begin{aligned}
F_{x_B} =& u_M \cdot F_{max} - \\
& (\rho \cdot A_{elev} \cdot |\vec{u}| \cdot \sin(\delta_{elev} + \alpha) + C_1 \cdot u_M \cdot \sin(\delta_{elev})) \cdot \sin(\delta_{elev}) - \\
& (\rho \cdot A_{rudd} \cdot |\vec{v}| \cdot \sin(\delta_{rudd} - \beta) + C_2 \cdot u_M \cdot \sin(\delta_{rudd})) \cdot \sin(\delta_{rudd}) - \\
& (\rho \cdot A_{wing} \cdot u^2 \cdot \sin(\frac{\delta_{ail}}{2}) + C_3 \cdot u_M \cdot \sin(\frac{\delta_{ail}}{2})) \cdot \sin(\frac{\delta_{ail}}{2})
\end{aligned}
\tag{2.22}
$$

$$
\begin{aligned}
F_{y_B} =& \rho \cdot A_{body} \cdot |\vec{v}| \cdot \sin(\beta) - \\
& (\rho \cdot A_{rudd} \cdot |\vec{v}| \cdot \sin(\delta_{rudd} - \beta) + C_2 \cdot u_M \cdot \sin(\delta_{rudd})) \cdot \cos(\delta_{rudd})
\end{aligned}
\tag{2.23}
$$

$$
\begin{aligned}
F_{z_B} =& \rho \cdot A_{wing} \cdot |\vec{u}| \cdot \sin(\alpha) + \\
& (\rho \cdot A_{elev} \cdot |\vec{u}| \cdot \sin(\delta_{elev} + \alpha) + C_1 \cdot u_M \cdot \sin(\delta_{elev})) \cdot \cos(\delta_{elev})
\end{aligned}
\tag{2.24}
$$

With the following constants:

$$C_1 = \frac{h_e \cdot D_{prop} \cdot F_{max}}{A_{prop}} \tag{2.25}$$

$$C_2 = \frac{h_r \cdot D_{prop} \cdot F_{max}}{A_{prop}} \tag{2.26}$$

$$C_3 = \frac{h_a \cdot D_{prop} \cdot F_{max}}{A_{prop}} \tag{2.27}$$

2.2.4 Moments

Together with the geometric properties of the aircraft, the acting moments upon the airplane become:

$$M_{x_B} = 2 \cdot \frac{wingspan}{4} \cdot F_{ail_{ff}} + 2 \cdot \frac{D_{prop}}{4} \cdot F_{ail_{ms}} \tag{2.28}$$

$$M_{y_B} = (l_e + \frac{h_e}{2} \cdot \cos(\delta_{elev})) \cdot F_{elev} \cdot \cos(\delta_{elev}) + l_{wing} \cdot F_{wing} \tag{2.29}$$

$$M_{z_B} = (l_r + \frac{h_r}{2} \cdot \cos(\delta_{rudd})) \cdot F_{rudd} \cdot \cos(\delta_{rudd}) - l_{body} \cdot F_{body} \tag{2.30}$$

As $l_e \gg h_e$ and $l_r \gg h_r$, I assume $(l_e + \frac{h_e}{2} \cdot \cos(\delta_{elev})) \approx l_e$ and $(l_r + \frac{h_r}{2} \cdot \cos(\delta_{rudd})) \approx l_r$.
Thus with inserted forces:

$$M_{x_B} = \frac{1}{4} \cdot wingspan \cdot (\rho \cdot A_{wing} \cdot u^2 \cdot \sin(\frac{\delta_{ail}}{2})) + C_4 \cdot u_M \cdot \sin(\frac{\delta_{ail}}{2}) \tag{2.31}$$

$$
\begin{aligned}
M_{y_B} =& l_e \cdot \rho \cdot A_{elev} \cdot |\vec{u}| \cdot \sin(\delta_{elev} + \alpha) \cdot \cos(\delta_{elev}) + \\
& C_5 \cdot u_M \cdot \sin(\delta_{elev}) \cdot \cos(\delta_{elev}) + l_{wing} \cdot \rho \cdot A_{wing} \cdot |\vec{u}| \cdot \sin(\alpha)
\end{aligned}
\tag{2.32}
$$

$$
\begin{aligned}
M_{z_B} =& l_r \cdot \rho \cdot A_{rudd} \cdot |\vec{v}| \cdot \sin(\delta_{rudd} - \beta) \cdot \cos(\delta_{rudd}) + \\
& C_6 \cdot u_M \cdot \sin(\delta_{rudd}) \cdot \cos(\delta_{rudd}) - l_{body} \cdot \rho \cdot A_{body} \cdot |\vec{v}| \cdot \sin(\beta)
\end{aligned}
\tag{2.33}
$$

With the following constants:

$$C_4 = D_{prop} \cdot \frac{h_a \cdot D_{prop} \cdot F_{max}}{A_{prop}} \tag{2.34}$$

$$C_5 = l_e \cdot \frac{h_e \cdot D_{prop} \cdot F_{max}}{A_{prop}} \tag{2.35}$$

$$C_6 = l_r \cdot \frac{h_r \cdot D_{prop} \cdot F_{max}}{A_{prop}} \tag{2.36}$$

Note, I assumed in the derivation of the moments, that all control surfaces only create moments around one axis. Thus the elevator only generates moments around the y-axis, the rudder only around the z-axis and the aileron only around the x-axis. This is not necessarily true especially for the rudder, however it simplifies the case, as no cross coupling between the rotational axis occurs in this type of model.
This concludes the complete set of forces and moments in the body frame of the system.

A list of the used symbols and acronyms:

Table 2.1: List of acronyms

Acronym	Description	Unit
	Physical Parameters	
ρ	Density of air	$[\frac{Kg}{m^3}]$
g	Earth acceleration	$[\frac{m}{s^2}]$
	Distances	
l_h	Distance COG to hinge Elevator	$[m]$
l_r	Distance COG to hinge rudder	$[m]$
l_{wing}	Distance COG to center of lift of main wing	$[m]$
l_{body}	Distance COG to center of lift of side body	$[m]$
h_e	Cord length of elevator	$[m]$
h_r	Cord length of rudder	$[m]$
h_w	Cord length of main wing	$[m]$
D_{prop}	Diameter of propeller	$[m]$
wingspan	Wingspan of main wing	$[m]$
	Areas	
A_{wing}	Main wing surface	$[m^2]$
A_{body}	Side body surface	$[m^2]$
A_{elev}	Elevator surface	$[m^2]$
A_{rudd}	Rudder surface	$[m^2]$
A_{prop}	Propeller circle surface	
	Masses/Inertias	
m	Mass of airplane	$[Kg]$
I_{xx}	Inertial moment around x-axis	$[Kg \cdot m^2]$
I_{yy}	Inertial moment around y-axis	$[Kg \cdot m^2]$
I_{xx}	Inertial moment around z-axis	$[Kg \cdot m^2]$
	Motor force	
F_{max}	Maximal motor force	$[N]$

Chapter 3

Simulation

This chapter will give an overview over the systems and methods used to simulate the equations of chapter 2.

3.1 Hardware Setup

The shockflyer which is simulated is based on the hobby model “Extra 300“ by “Shillouette Art“. The flight test where also conducted with this model.

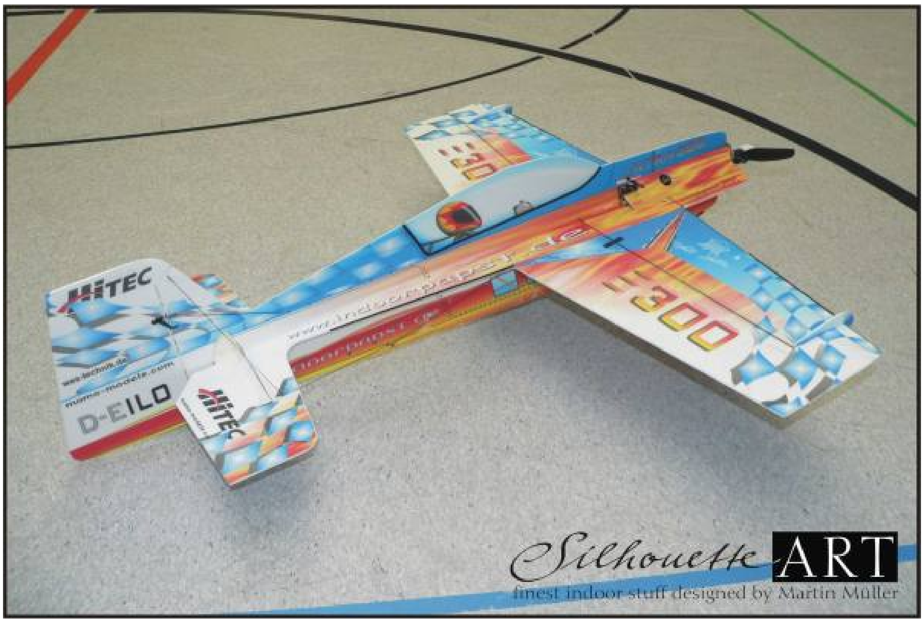


Figure 3.1: The shockflyer used for flight tests

The shockflyer itself is constructed out of 3mm Depron together with carbon fiber rods for structural support. Please note that this implies that the complete aircraft is constructed out of flat plate surfaces. The aircraft was equipped with the following electronics:

Item	Product
Brushless Motor	Hacker A10-9L
Brushless Controller	YGE-8s
Propeller	GWS 8x4.3
Lipo-Akku	SLS-APL 500mAh 2s1p 7,4V 30C
Elevator/Rudder-Servo	Torcester - Micro Servo NR-50 5g
Aileron Servo	Torcester - Micro Servo NR-62 8g

The rudder respectively the elevator surface are powered using two NR-50 5g servos, while the aileron is deflected using one NR-62 8g servo. For propulsive energy a brushless Hacker A 10-9L motor is used in combination with a 8x4.3, 2-bladed propeller. Together with the 500 mAh lithium polymer battery, flight times of up to 10 minutes can be achieved, depending on the flown maneuvers.

3.1.1 Parameters

Four sets of parameters are necessary to simulate the aircraft.

- Geometric properties
- Mass
- Moments of Inertia
- Full thrust of motor

The mass as well as the geometric properties can be measured from the airplane itself. To identify the moments of inertia, the airplane was modelled in a CAD program which automatically calculated the moments of inertia:

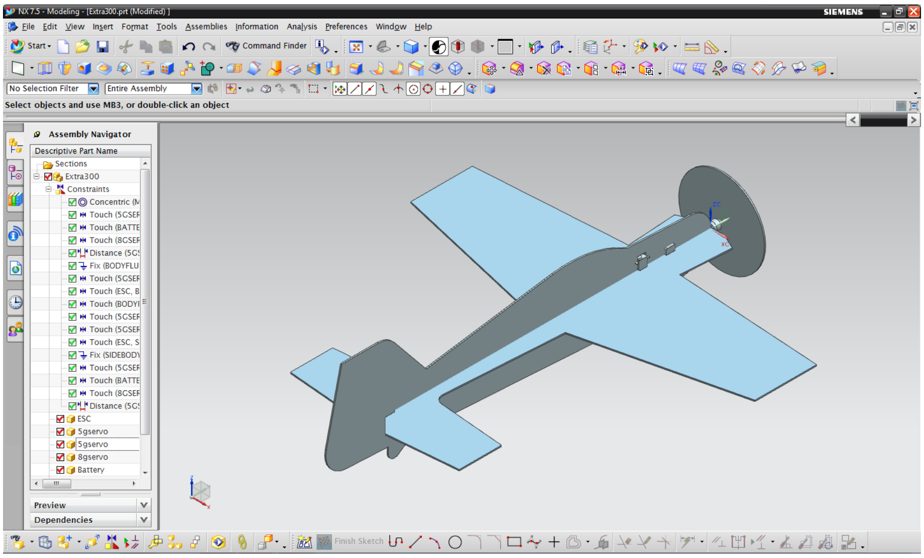


Figure 3.2: The airplane modeled in CAD NX 7.5, with all electronic components attached

The thrust at full throttle of the motor in combination with the propeller listed in table 3.1, can be looked up in a document provided by the manufacturer [7]. All identified parameters can be found in appendix A.
The CAD model is also used to visualize the outputs of the simulations.

3.1.2 Modeling of Servos

The governing equations for the system dynamics are derived in chapter 2. However the actuator dynamics where not derived in this chapter as these are not an integral part of the flight dynamics. As we will be using servo motors for actuating the control surfaces in reality, we have to assume that the deflection of the control surfaces do not follow the input signal instantly, but rather will show a certain delay in their response.
In the following the dynamics of the servos are simplified as a first order system with an approximated time constant τ [6].
To estimate the time constant the step response behavior given by the manufacturer of the servo is used. Together with the step response solution of a first order system, the time constant τ is then readily obtained.
The step response solution of the following first order system [3]:

$$\tau \frac{\partial y(t)}{\partial t} = -y(t) + u(t) \tag{3.1}$$

$$\frac{X(s)}{U(s)} = \frac{1}{\tau s + 1} \tag{3.2}$$

Is given by the equation

$$y(t) = k \cdot (1 - e^{-\frac{t}{\tau}}) \tag{3.3}$$

For a step input with height $u(t) = k$ for $t > 0$.
As this solution is an exponential function it will not reach the value k in finite time. Therefore it is defined that the first order system reaches its endpoint at $y(t) = 0.95 \cdot k$.
The step response of the real servo motor is obtained from the specifications.
"Torcester - Micro Servo NR-50 5g": "Velocity at 6V: 0,10 sec/60°" [8]
Because the supply voltage for the servo is only 5 V, it is estimated that the servo will take $\frac{6V}{5V} \cdot 100ms = 120ms$ to turn over these 60°.
Together with equation 3.3:

$$\tau = -\frac{t_{end}}{ln(1 - 0.95)} = 0.03825s \tag{3.4}$$

$$\frac{1}{\tau} = 26.1438\frac{1}{s} \tag{3.5}$$

The specification of the "Torcester - Micro Servo NR-62 8g" states [8]: "Velocity 4,8V (sec/60°): 0,12". As this is roughly the same value as used in the calculation above, it is assumed that the two servo types have the same time constant τ.

3.2 Flight envelope

In the following a analysis for a sustained free flight based on the model derived in chapter 2 and the identified parameters of section 3.1.1 is shown. The most simple case of sustained free flight in a closed area is that of a circular flight-path with constant height and constant radius. To characterize this flight state consider the following sketch of the situation.

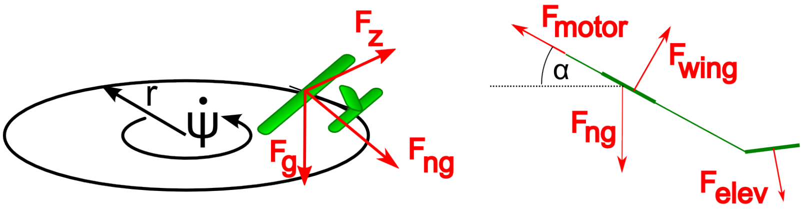


Figure 3.3: Sketch of continuous free flight case in a circle

Out of the sketch:

$$F_{ng} = \sqrt{F_z^2 + (m \cdot g)^2} \tag{3.6}$$

$$F_{ng} = m \cdot n \cdot g \tag{3.7}$$

$$F_z = m \cdot \dot{\psi}^2 \cdot r \tag{3.8}$$

$$v = r \cdot \dot{\psi} \tag{3.9}$$

n	Multiple of earth acceleration g	[]
r	Curve radius	$[m]$
$\dot{\psi}$	rotational velocity	$[\frac{rad}{s}]$

Therefore:

$$n = \sqrt{\left(\frac{\dot{\psi} \cdot v}{g}\right)^2 + 1} \tag{3.10}$$

The force F_{ng} has to be counteracted by the aerodynamic wing forces, the motor force and the elevator force. Thus:

$$F_{ng} \cdot \sin(\alpha) = F_{motor} \tag{3.11}$$

$$F_{ng} \cdot \cos(\alpha) = F_{wing} \tag{3.12}$$

The term F_{elev} is neglected, as it is relatively small compared to the other factors.

The angle of attack α can be computed out of the equations 3.10, 3.12 together with the equation 2.18.

$$\alpha = \tan^{-1}\left(\frac{m \cdot g \cdot \sqrt{\left(\frac{\dot{\psi} \cdot v}{g}\right)^2 + 1}}{\rho \cdot v^2 \cdot A_{wing}}\right) \tag{3.13}$$

Thus we can state α now for every flat, steady state flight condition. The current flight condition is characterized by two variables: v and $\dot{\psi}$.

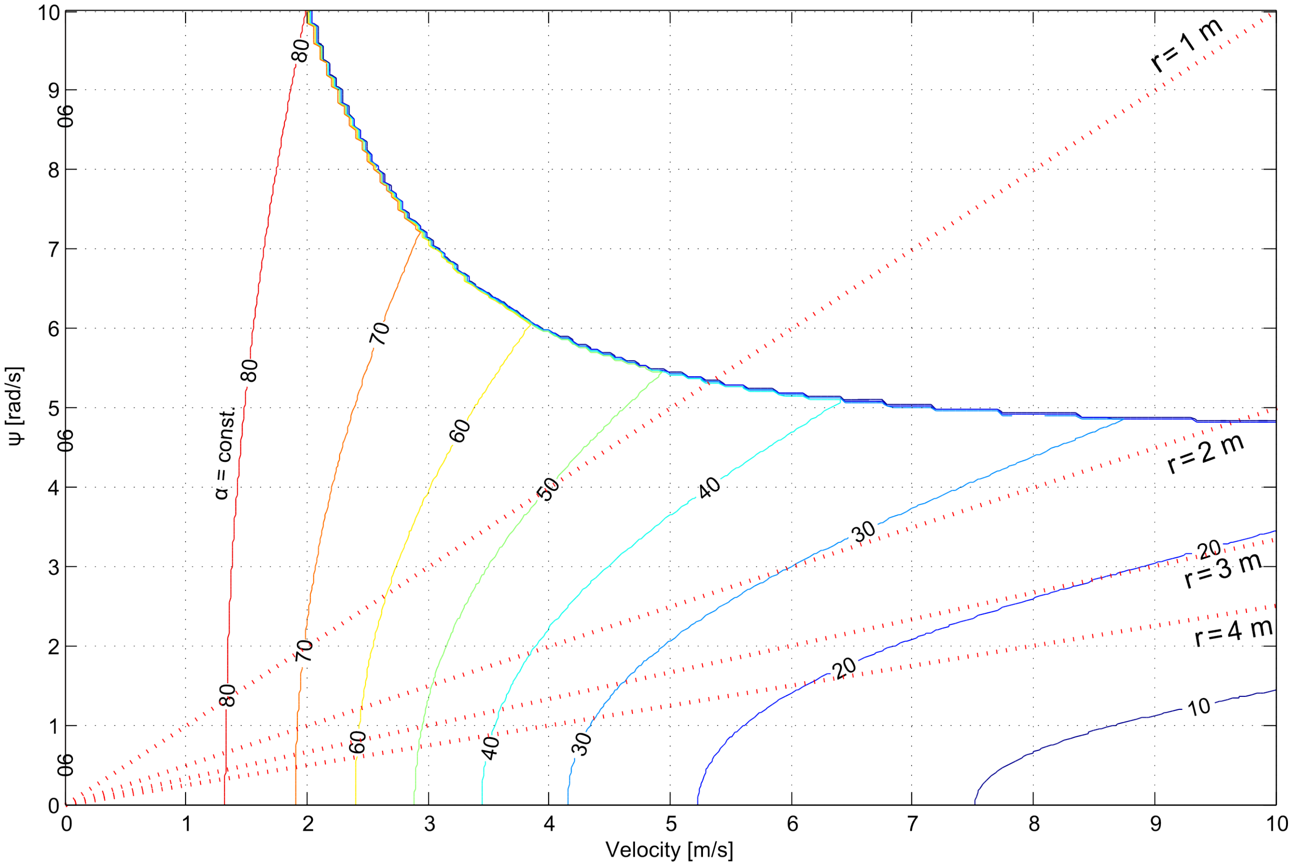


Figure 3.4: The flight envelope for our model.

To calculate this graph the parameters introduced in section 3.1.1 where used. The colored height lines represent the angle of attack α while the red dashed lines represents the curve radius at the state indicated by the axes. Also note that alpha was set to 0 if the motor force was beyond the force limits of the motor which is the case in the upper right corner of the graph. In addition every point was checked for balance of moment. Thus the graph was searched for a case in which the moment generated by the elevator would not be sufficient to compensate the moment generated by the wing. However no points in which the elevator force could not balance the main wing moment where found. This leaves us with the conclusion that we should be able to transit between hover and free flight without crossing uncontrollable states.
Also note that the graph characterizes the flight envelope of the system in a very short and neat way, and shows that according to our equations, every curve radius should be feasible to fly.

3.3 C++ Simulation

This section will describe the process of implementing a working C++ simulation into the existing framework of the FMA. It will also describe the outline of the controller and its implementation. The actual code can be found among the deliverables of this thesis.

3.3.1 The Simulator class

Structure

The fixed wing simulator class is structured similarly to the quadrocopter simulator class, as both fulfill the same task, which is to simulate the dynamics of the respective vehicle. However the underlying governing equations in these two cases differ. The programmed code performs the following tasks in the given order:

1. Receive commands from controller
2. Compute deflection of control surfaces
3. Compute forces acting upon the system
4. Integration of acceleration
5. Integration of velocity

These tasks where completed as follows:

1. The reception of commands from the controller was handled using a public method which is called from the main loop. This enabled the interface between the controller and the simulator.

2. Each time the simulation function was called, the actuator dynamics where calculated over the analytical solution of the first order system given in chapter 3.1.2. The reason to use the analytical solution instead of the differential equations for a step function is the fact that the controller can send commands with an update rate of about 50Hz, while the simulation is propagated with a much higher frequency. Also note that the analytical solution to a step response represents the exact discretization of the first order system in question.

3. After this step, the calculation of the forces is straightforward by using the equations derived in chapter 2.

4. The integration of angular acceleration and translational acceleration is performed over a Runge-Kutta solver of 4th grade. A Runge-Kutta integration for angular acceleration as well as for translational acceleration was chosen, because of the fact that the aerodynamic forces depend on translational velocities. Therefore a simpler integrational step like the Euler-integration would yield excessive integration errors which in turn would lead to unrealistic behavior of the airplane.

5. The last step of the simulation is the integration of velocity to gain the attitude and position of the aircraft. This step is automated by the framework.

3.3.2 Controller

All equations derived in chapter 2 where not verified or tested upon their correctness and therefore include some uncertainty. Thus the controller is desired to perform its task of stabilizing the aircraft robustly, to compensate for the uncertainties of the modeling. In the following a control strategy based on a PD controller is introduced.

Control strategy

I state that the equations of motion in hover condition can be approximated as four decoupled SISO systems. To justify this statement, the equations of motion for rotation and translation will be reviewed:

Rotation

The states in hover condition is given by:

$$\begin{pmatrix} x \\ y \\ z \\ u \\ v \\ w \\ \psi \\ \theta \\ \phi \\ p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{\pi}{2} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.14}$$

The hover point 3.14 inserted in the equation of rotational movement given by 2.3 together with the moments from 2.31 to 2.33 yield:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{C_4 \cdot u_M \cdot \sin(\frac{\delta_{ail}}{2})}{4 \cdot I_{xx}} \\ \frac{C_5 \cdot u_M \cdot \sin(\delta_{elev}) \cdot \cos(\delta_{elev})}{I_{yy}} \\ \frac{C_6 \cdot u_M \cdot \sin(\delta_{rudd}) \cdot \cos(\delta_{rudd})}{I_{zz}} \end{pmatrix} \tag{3.15}$$

Assuming small rotational speeds and hover condition we can also state the integrational step for the euler angles [6]:

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} \approx \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} \tag{3.16}$$

Together with the relationship $\cos(x) \cdot \sin(x) = \frac{1}{2} \cdot \sin(2x)$ and 3.15, equation 3.16 becomes:

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{C_4 \cdot u_M \cdot \sin(\frac{\delta_{ail}}{2})}{4 \cdot I_{xx}} \\ \frac{C_5 \cdot u_M \cdot \sin(2 \cdot \delta_{elev})}{2 \cdot I_{yy}} \\ \frac{C_6 \cdot u_M \cdot \sin(2 \cdot \delta_{rudd})}{2 \cdot I_{zz}} \end{pmatrix} \tag{3.17}$$

Therefore every input is fed through a double integrator before acting upon one euler angle, which can be viewed as three decoupled second order systems.

Translation

For the hover point 3.14, the rotation matrix T from 2.2 becomes:

$$T = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{3.18}$$

This inserted into equations 2.2, yields:

$$\ddot{\vec{x}}_I = \frac{1}{m} \cdot \begin{pmatrix} F_{z_B} \\ F_{y_B} \\ -F_{x_B} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \tag{3.19}$$

$$\tag{3.20}$$

Together with the force equations given by 2.22 to 2.24.

$$\ddot{\vec{x}}_I = \frac{1}{m} \cdot \begin{pmatrix} -C_1 \cdot u_M \cdot 0.5 \cdot \sin(2 \cdot \delta_{elev}) \\ C_2 \cdot u_M \cdot 0.5 \cdot \sin(2 \cdot \delta_{rudd}) \\ u_M \cdot F_{max} - C_1 \cdot u_M \cdot \sin^2(\delta_{elev}) \\ -C_2 \cdot u_M \cdot \sin^2(\delta_{rudd}) - C_3 \cdot u_M \cdot \sin^2(\frac{\delta_{ail}}{2}) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \tag{3.21}$$

Neglecting all terms of higher order as well as the entries for x and y as:
$C_1 \cdot 0.5 \cdot \sin(2 \cdot \delta_{elev}) \ll F_{max}$ and $C_2 \cdot 0.5 \cdot \sin(2 \cdot \delta_{rudd}) \ll F_{max}$.

$$\ddot{\vec{x}}_I = \frac{1}{m} \cdot \begin{pmatrix} 0 \\ 0 \\ u_M \cdot F_{max} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \tag{3.22}$$

Thus solving equation 3.22 and 3.17 for steady state condition we get the inputs in equilibrium at working point:

$$\begin{pmatrix} u_M \\ \delta_{elev} \\ \delta_{ail} \\ \delta_{rudd} \end{pmatrix} = \begin{pmatrix} \frac{g \cdot m}{F_{max}} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.23}$$

Note that although equation 3.17 does not show any coupling between the different control surfaces, it displays a strong coupling between the motor input

and the moments generated.
This shows that at hover condition, the euler attitude angles can be viewed as three uncoupled second order SISO systems but without any damping or spring behavior. An asymptotical stable system can be achieved by choosing any negative feedback on position and velocity [3].

Servo State

As shown in 3.1.2, the inputs cannot be accessed directly, but are rather a low-pass filtered signal of the commanded positions. Thus for any of the three angular dynamic SISO systems shown in 3.17 the system can be depicted as seen in figure 3.5.

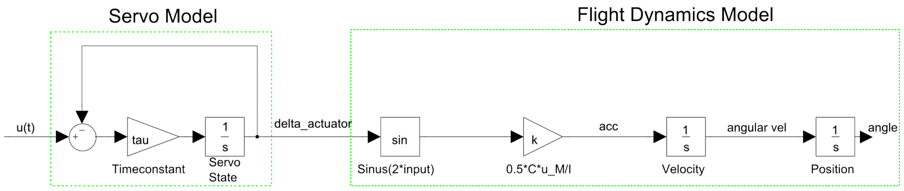


Figure 3.5: Schemata of the angular second order system together with the servo dynamics

This SISO system, representing angular dynamics, now possesses 3 states: Angle, angular velocity and the servo position.
To control this system a feedback on position and velocity of the aircraft is introduced:

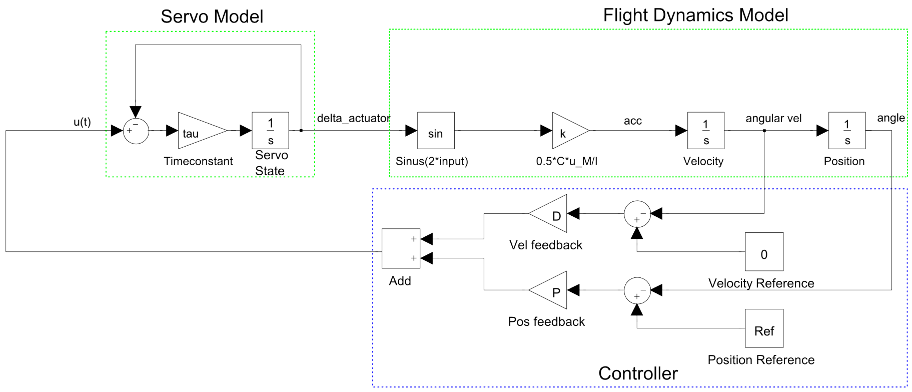


Figure 3.6: Schemata of the angular second order system together with the servo dynamics and feedback

The poles of the system linearized around hover condition can be written as:

$$\begin{pmatrix} \dot{\delta} \\ \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} -\tau & -P \cdot \tau & -D \cdot \tau \\ 0 & 0 & 1 \\ k & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \delta \\ \theta \\ \dot{\theta} \end{pmatrix} \tag{3.24}$$

Plotting these eigenvalues for a empirically chosen P-Value of 3 and a D-Value ranging from 0 to 1 yields the following plot:

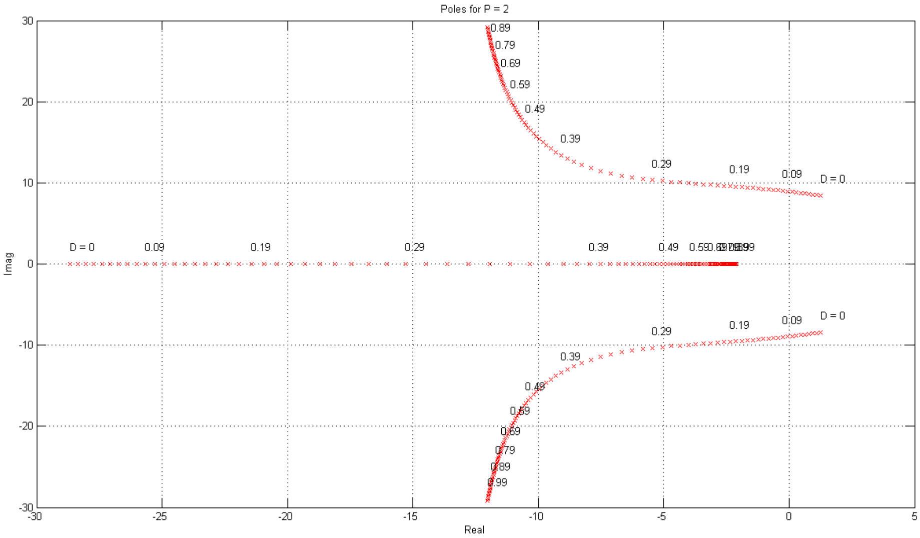


Figure 3.7: Eigenvalues of the 3rd order Feedback system for fixed P value of 3

To be able to apply 2^{nd} order system theory for this problem, the third pole introduced by the servo state must be neglectable. After [2] we can do so if this pole is at least 10 times faster than the remaining two poles. Thus after figure 3.7, this is true for a D-value of approximately 0.15.
The P-value for the simulation was chosen empirically such that the overall response was fast enough but maintained reasonable robustness margins.
This shows that the 3 angular dynamics systems can be stabilized by using a PD controller, with values chosen such that the servos state is neglectable.

Nonlinearity handling

As indicated in the equations 3.17 the output of the controller is fed through a sinus function before reaching the integrators. To eliminate the effects of this nonlinearity, the outputs of the state-feedback controller are mapped onto the inverse of the sinusoid function as follows:

$$\delta_{requested} = \frac{\sin^{-1}(2 \cdot u(t))}{2} \tag{3.25}$$

For every euler angle and the x_B -axis, the controller becomes:

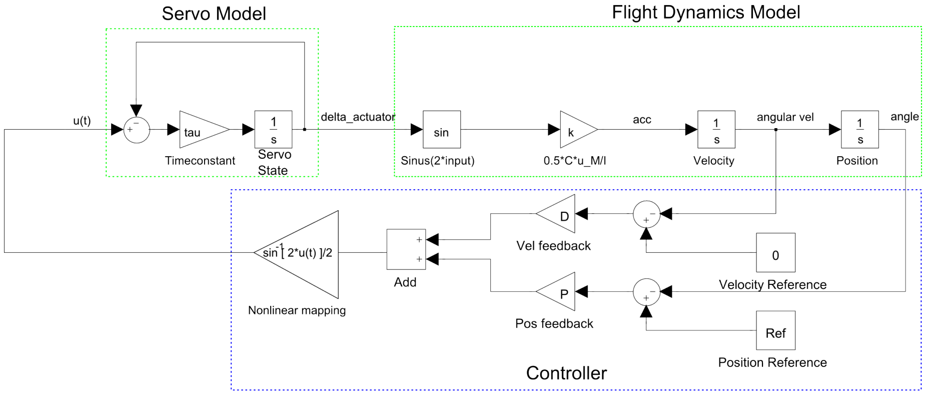


Figure 3.8: Schemata of the angular second order system together with the servo dynamics with feedback and nonlinearity handling

Thus the system behaves linear to the inputs of the controller until the deflection boundaries $\delta = \pm 45°$ are reached.
This control loop which stabilizes the attitude around the hover condition will be called the “Attitude Control loop“ in the following.

Navigational loop

However this first loop does not compute any feedback on the position of the aircraft and therefore does not stabilize the position of the aircraft. To overcome this drawback, a second loop was installed based on the following two correlations: Firstly, the force in x- and y-direction can be steered over tilting the system in either one direction. Therefore the following matrix, which rotates the system from hover conditions to small deviations of the hover condition based on the euler angles, emerges:

$$T_{ID} = \begin{pmatrix} \cos(\Delta\theta) & \sin(\Delta\psi)\sin(\Delta\theta) & -\cos(\Delta\psi)\sin(\Delta\theta) \\ 0 & \cos(\Delta\psi) & -\sin(\Delta\psi) \\ \sin(\Delta\theta) & \cos(\Delta\theta)\sin(\Delta\psi) & \cos(\Delta\psi)\cos(\Delta\theta) \end{pmatrix} \tag{3.26}$$

And therefore equation 2.2 becomes:

$$\ddot{\vec{x}}_I = \frac{F_{max} \cdot u_M}{m} \cdot \begin{pmatrix} \cos(\Delta\psi) \cdot \sin(\Delta\theta) \\ \sin(\psi) \\ \cos(\psi) \cdot \cos(\theta) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \tag{3.27}$$

With this correlation, the position integrators can be accessed without having to sacrifice control over the attitude angle. Thus a PD controller is installed which computes feedback on position to set the setpoints of the attitude loop, which will tilt the aircraft to reach the desired position. This second loop is named navigational loop, somewhat taken from the pilots view of the problem. A sketch of the two loops together is presented below:

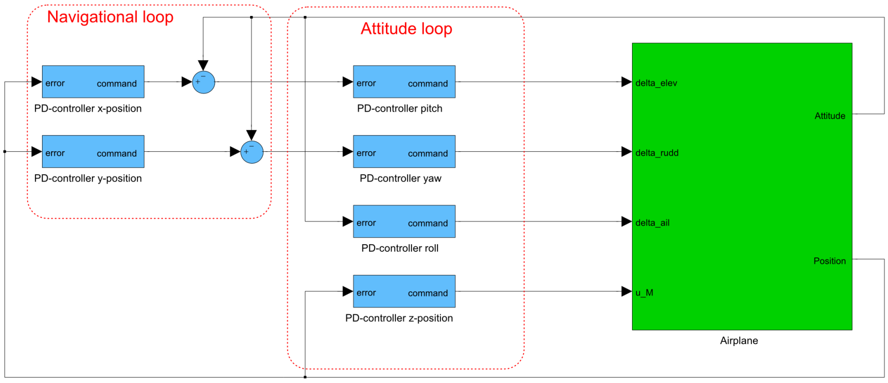


Figure 3.9: Schematics of the complete controller with attitude and navigational loop

It is very important to notice that the navigational loop has to be tuned such that its poles are much slower then those of the attitude loop. Otherwise unwanted interferences between the two control loops will eventually yield an unstable overall controller. This separation in time was performed and confirmed empirically, thus no analysis concerning the actual pole placement of the two pole pairs where conducted.

Motor

Likewise to the servos, the motor also shows a delay from its input to the force it develops. However unlike the servos, the manufacturer does not provide a reference for this time constant. The time constant of the motor is therefore chosen to have the same value as that of the servos. As the dynamics of the x_B - axis of the aircraft can be described as a second order system, this axis displays the same structure as the rotational axes. Out of this reason the x_B - axis is controlled with a PD-controller, just as the angular dynamic.
All PD values are listed in appendix B.

3.3.3 C++ Simulation results

Together with the simulation and controller described above the following tests where successfully demonstrated on the C++ simulator with all 6 degrees of freedom enabled:

- Stable Hover without position feedback
- Stable Hover with position feedback
- Control over position in space while hover
- Stable free flight

One particular simulation will be shown in the following.
The task of this simulation was to demonstrate the controllers ability to stabilize the aircraft in upright position and to show its behavior to jumps in commanded

position. Figures 3.10 and 3.12 display the data recorded during this simulation run. This particular test initialized the aircraft in hover position, but yawed by 90 degrees.

$$\begin{pmatrix} \theta_{init} \\ \phi_{init} \\ \psi_{init} \end{pmatrix} = \begin{pmatrix} \frac{\pi}{2} \\ 0 \\ \frac{\pi}{2} \end{pmatrix} \tag{3.28}$$

This forced the controller to reorient the aircraft at the beginning of the simulation to assume the correct hover position. At time stamp $t \approx 0s$ until $t \approx 3.5s$ this roll reorientation is visible on the roll plot of figure 3.12. The commanded position in x- and y-axis are indicated in 3.11 and correspond to a diagonal translation in the x-y plane.

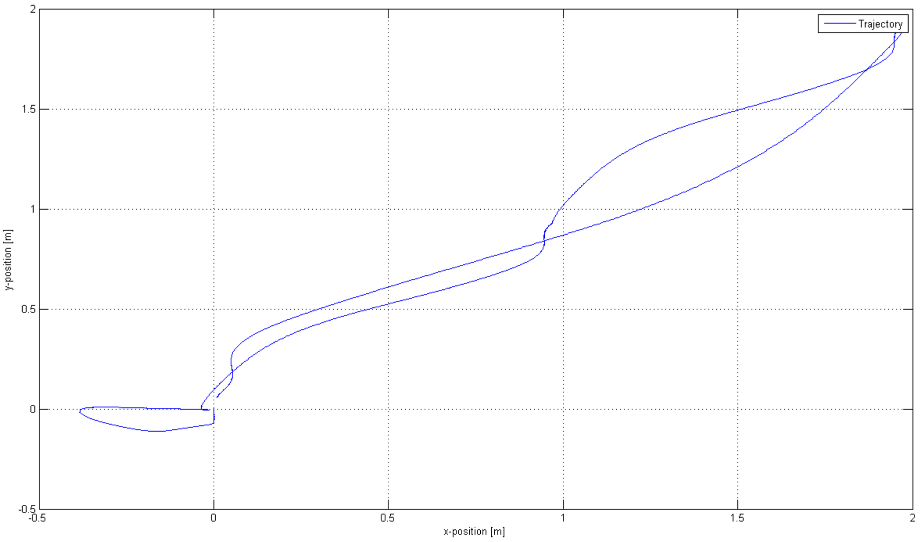


Figure 3.10: Trajectory during the simulation.

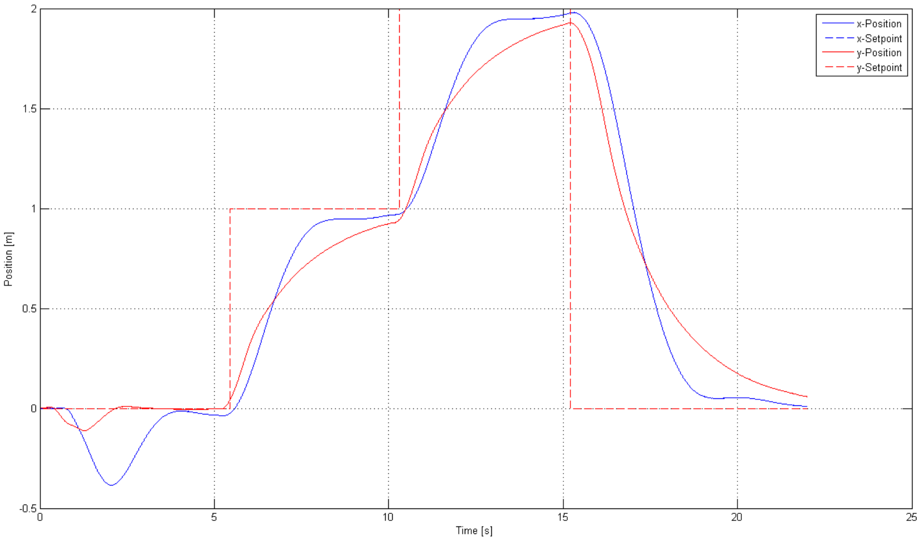


Figure 3.11: The tracking performance of the x and y position controller, together with the commanded positions.

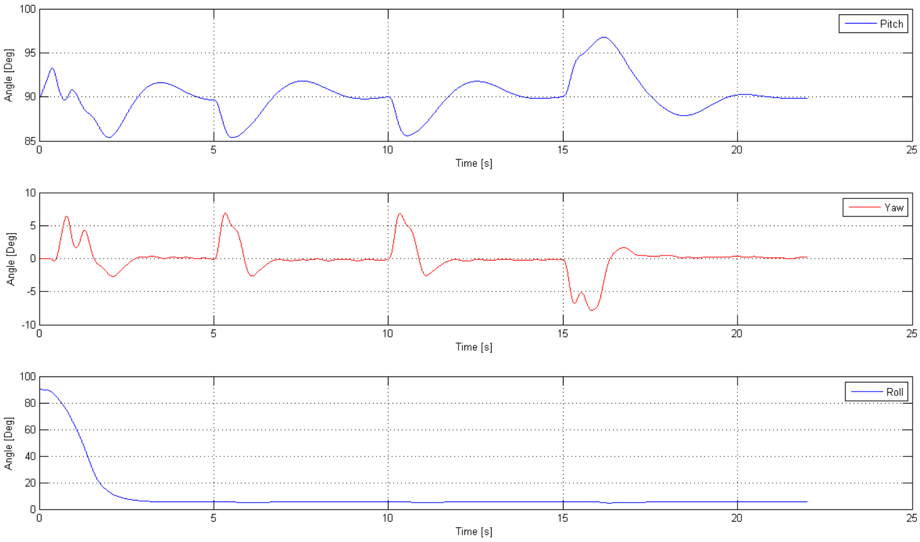


Figure 3.12: Yaw, Pitch and Roll angle during the simulation.

Note that the two axis are not able to track the position jump immediately but require roughly 7 seconds to assume the commanded position. A faster response could be achieved by tuning the PD values until a faster and even more stable response is achieved, however for demonstration purposes the found PD values are sufficient.

3.4 Technical Implementation

This section will describe some implementation aspects which are conceptually not relevant, however which play a major role for the simulation environment. A description of the used angle calculations as well as considerations of roll rate noise and a utilized coordinate shift will be described in the following.

3.4.1 Gimbal Lock and coordinate change

The hover condition described by equation 3.14 also implies that the aircraft is flying in the gimbal lock attitude [6]. The limits of the euler angle description are given by equations 3.29 to 3.31 [6].

$$\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}] \tag{3.29}$$

$$\psi \in [-\pi, \pi] \tag{3.30}$$

$$\phi \in [-\pi, \pi] \tag{3.31}$$

Note that the θ angle of equation 3.14 crosses the boundary given by 3.29 which leads to the gimbal lock phenomena.
However the roll axis is constrained from $[-\pi, \pi]$ thus offering a larger valid angle range then the pitch axis, which is constrained by $[-\frac{\pi}{2}, \frac{\pi}{2}]$.
This fact is exploited by turning the descriptive frame of the aircraft by 90° to circumvent the gimbal lock problem for hover condition.
The emerging coordinate shift is illustrated in 3.13

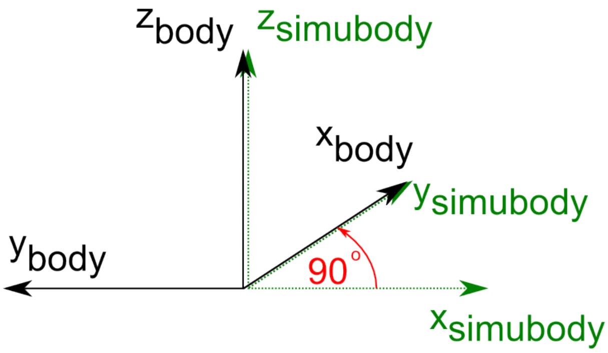


Figure 3.13: The coordinate shift performed to avoid gimbal lock.
Dashed green: Original coordinate system,
Solid black: Shifted coordinate system

Mathematically this coordinate shift can be described by the following rotation matrix, which turns the coordinate system from the original “Simubody“-frame to the shifted “Body“-frame.

$$R_{BSb} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.32}$$

Important to notice is that the simulation only “sees“ the “Simubody“-frame, however all implemented force and moments equations are turned as described above such that they only “see“ the shifted “Body“ frame. Also the gimbal lock problem is not erased by this measure, but was only moved to another more favourable point.
In the next chapter it will be explained that the Euler angles where not used for calculating the control inputs. The coordinate shift is still important, as the simulation environment uses the Euler angles for the calculation of attitude and position.

3.4.2 Angle calculation

To make the controller indepent of the euler angles, and to be able to better control the calculation of the attitude angles, a different method for the calculation of the attitude angles was implemented. These calculated angles where exclusively used for the control algorithm.
The idea of this algorithm, was to use the rotation matrix given by the simulator framework, to calculate three angles by means of the $atan2()$ function. In the following the rotation matrix which transform a vector in body coordinates into a vector in inertial coordinates will be called R_{IB}. Three attitude angles called a, b and c can be calculated out of the R_{IB} by means of the $atan2()$ function as follows:

1. Calculate $y_B = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}_B$ in inertial coordinates:
   $y_{BI} = R_{IB} \cdot y_B = \begin{pmatrix} y_{1_{BI}} \\ y_{2_{BI}} \\ y_{3_{BI}} \end{pmatrix}_I$

2. Calculate the angle a by $a = atan2(y_{1_{BI}}, y_{2_{BI}})$

3. Calculate $x_B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}_B$ in inertial coordinates:
   $x_{BI} = R_{IB} \cdot x_B = \begin{pmatrix} x_{1_{BI}} \\ x_{2_{BI}} \\ x_{3_{BI}} \end{pmatrix}_I$

4. Turn this vector by the angle a:
   $x_{Ba} = R_{aI} \cdot x_{BI} = \begin{pmatrix} \cos(a) & \sin(a) & 0 \\ -\sin(a) & \cos(a) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot x_{BI} = \begin{pmatrix} x_{1_{Ba}} \\ x_{2_{Ba}} \\ x_{3_{Ba}} \end{pmatrix}_a$

5. Calculate angles b and c: $b = atan2(x_{3_{Ba}}, x_{1_{Ba}})$ and $c = atan2(x_{3_{Ba}}, -x_{2_{Ba}})$

Figure 3.14 illustrates how these angles define the orientation of an vector.

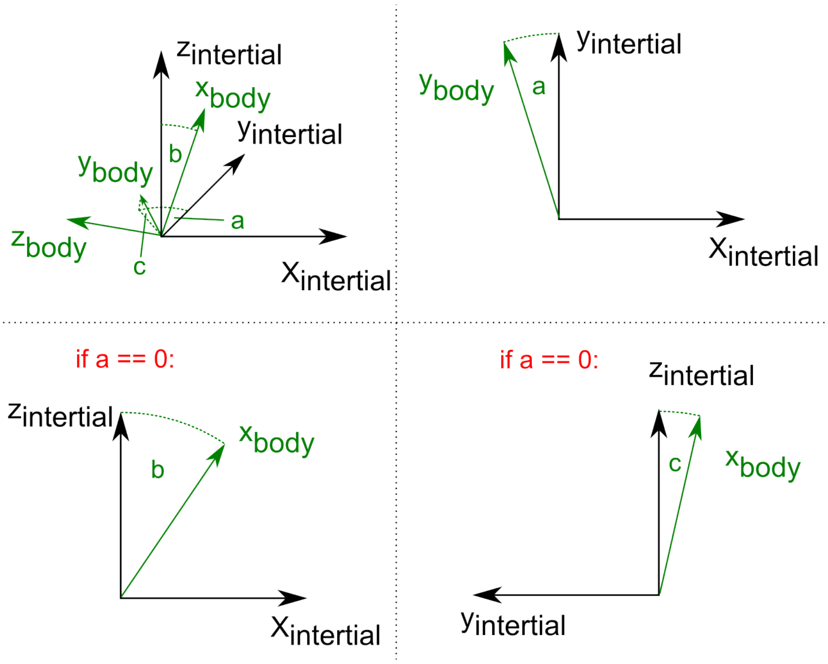


Figure 3.14: Illustration of the calculation of the angles a, b and c

Note that the code for the calculation has to be executed such that it handles the coordinate change discussed in 3.4.1.

3.4.3 Roll rate noise

The FMA estimator which supplies all information about the current state of the simulation to the controller normally includes realistic measurement noise. In the plots shown in 3.3.3, all measurement noise for ω, had to be switched off due to the inability of the controller to stabilize the aircraft with noise enabled.

The Vicon system does not deliver information about the current body rates but rather only position and attitude of the airplane. To calculate the body rates the controller has to numerically differentiate the attitude data of the airplane by applying a finite difference scheme. As the Vicon data is fed with 200 Hz, this process yields a very noisy signal. The PD-controller was not able to stabilize the airplane in simulation given the body rates corrupted with noise, which is why the noise was removed from the signal for the simulation.
In reality of course noise cannot be switched of, however we justified the disabled noise by the following two assumptions about the real life tests:

1. High aerodynamic damping because of body rates, which is not simulated
2. Lower Vicon data noise because of bigger distance between markers

The tests showed, that the inherent natural damping caused by body rates is sufficient to replace the D-controller. This fact will also be discussed in 4.2.2.

Chapter 4

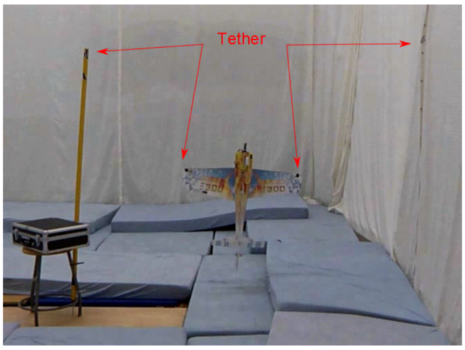
Flight Tests and Results

This section will give an overview over the infrastructure used in the flight tests, the problems encountered during these and the data of one typical flight test. All flight tests where conducted inside the FMA where a Vicon system was employed for feedback of the attitude and position of the aircraft. The description of the used aircraft can be found in section 3.1.

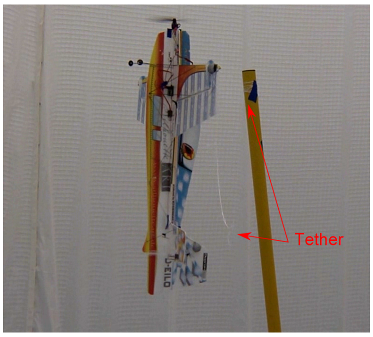
4.1 Setup

4.1.1 Teststand

For the first few flight tests, a teststand was designed to experiment with the controller and test setup without fear of harming the aircraft. A picture of the static teststand is given below:



(a) The aircraft suspended in the teststand



(b) Sideview of the Teststand, while testing

The figure 4.1(a) shows the aircraft hanging inside the static teststand with switched of motor, ready to begin the hover tests. Red arrows mark the beginning and the end of the tethers, which run from the pole respectively the net to the wingtips. Figure 4.1(b) shows the same setup while a hover test is performed. The tether which is hanging loosely, is marked with red arrows.
For later hover tests which included a jump in x- or y-position, the aircraft was bound to a hand held pole, which allowed greater freedom in movement.

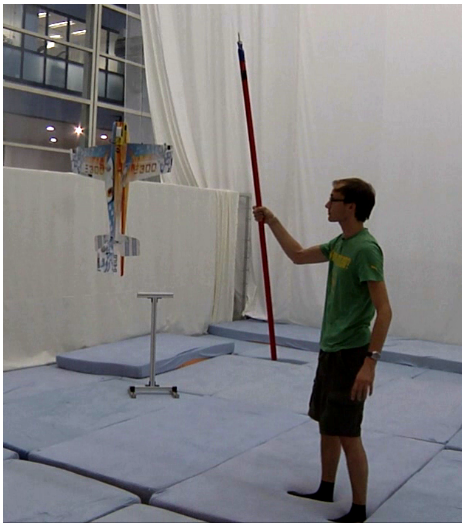


Figure 4.1: Manually tethered flight

Figure 4.1 shows the hand held pole which runs a wire from its top to the aircrafts wing tip, to prevent the aircraft from damage if the engine was shut off.

4.1.2 Communication

For information about the attitude and the position of the aircraft, markers where attached to the airplane, which then where tracked via a Vicon system. This information was processed and sent to the network using the software framework provided by the FMA. After sending the data to the network, a fixed computer or laptop received the Vicon data, and processed the information using the controller, developed during the simulation. The commands were then sent over a commercial RC-transmitter to the airplane. To access the RC-transmitter a "Buddy-box" system was supplied by the FMA, which provided the link between the computer and the RC-transmitter. The full situation is illustrated in figure4.2.

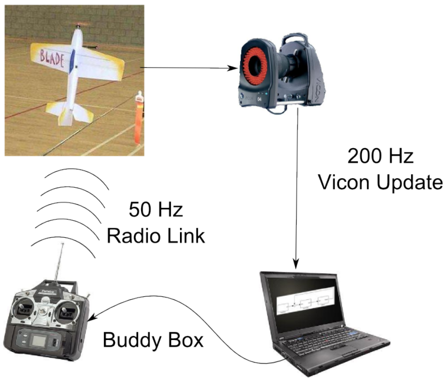


Figure 4.2: The communication link

4.1.3 Security

To minimize the risk of dangerous situations due to the rotating propeller, a dead mans switch was implemented in the controller. The commanded motor speed was automatically set to 0 if a distinguished button on the computer running the controller was not pressed. If this switch was released during testing, the motor stopped to prevent damage to operators or material.

4.2 Flight Tests

The flight test were scheduled to test the following controller setups:

- Open loop hover
- Closed loop hover
- Closed loop hover with step inputs

The P and D values found in the simulation where not optimal. Therefore these parameters where hand tuned during the flight test. The P and D values can be found in appendix B

4.2.1 Final flight test

In the following I want to show one successful flight log. The trajectory is displayed in 4.3, the yaw roll and pitch angle are shown in figure 4.4 while the z-position is plotted in figure 4.5.

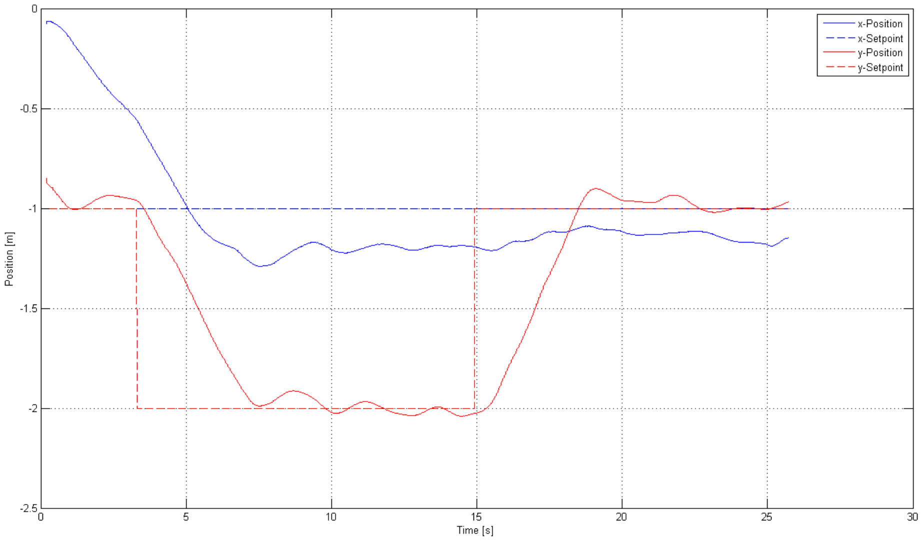


Figure 4.3: The position in x-y direction of the test flight

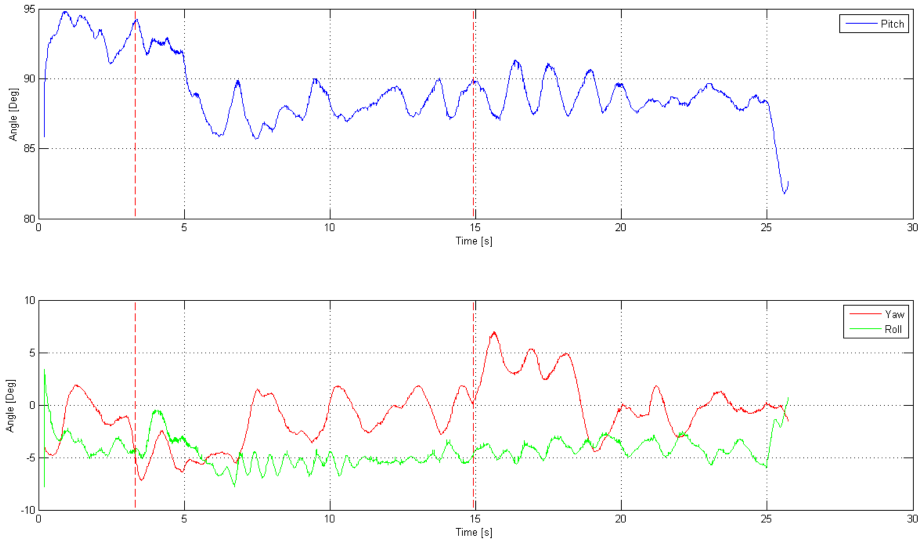


Figure 4.4: Pitch, roll and yaw for the given test flight

The test was performed by using the hand held pole to grant the needed freedom in movement. After the beginning of the test a series of jumps in commanded position was performed as listed in table 4.1. Also note that the test ended at $t = 25s$, at which the motor was switched off, and the aircraft was retaken by hand.

In figure 4.3 the reaction of the system to these inputs is shown. The y-axis tracked these jumps well, while the x-axis shows a constant deviation from its commanded position of approximately $x_{command} - x \approx 0.2m$ after it reaches steady state at $t \approx 7.5$. Also note that as the x-axis reaches its equilibrium

Table 4.1: Commanded position profile

	$t = 0$	$t \approx 3$	$t \approx 15$	Units
$x_{command}$:		-1		[m]
$y_{command}$:	-1	-2	-1	[m]
$z_{command}$:		2		[m]

state, the mean of the pitch angle is not $\theta = 90°$ as would be expected, but rather shows a constant offset of approximately $1.0°$, as visible in figure 4.4.
These two observations lead to the conclusion that the axes of the Vicon model do not agree with the physical axes of the actual aircraft. Thus the x_B - axis assumed by the Vicon system shows a constant offset from the x_B - axis of the real aircraft leading to a constant pitch offset. This results in a constant drift of the aircraft until the pitch error will be compensated by the navigational loop because of the large error in x-position. This statement can be justified over calculating the controller steady state inputs of the aircraft:
The steady state error is $e_x \approx 0.2m$ with an angle error of $e_\theta \approx 1.0°$, whereby $e_x = x_{command} - x$ and $e_\theta = \theta_{command} - \theta$. If recalling that the position error is multiplied with a constant factor to obtain the command for the pitch axis, we can state the following together with the PD values given in appendix B:

$$\begin{aligned} e_\theta &= e_x \cdot k_{P_{x,Navigation}} \\ 1.0° &= 0.2 \cdot 0.1 \ rad \\ 1.0° &\approx 1.1459° \end{aligned}$$

This short calculation shows that this constant error can be explained by a faulty axis orientation assumption made by the Vicon system. We can also state that the misalignment of the pitch axis is approximately $1°$. The yaw axis however seems to be better aligned with the physical yaw axis as the y-axis does not show a constant deviation.
Also note that the constant offset of the roll axis is a product of the moment generated by the motor, which is counteracted by the aileron with a constant offset around x_B.
Figure 4.5, shows the z-tracking performance of the controller throughout the test. Because of the problems discussed in 4.2.3, the tracking performance in the z-axis is rather poor.

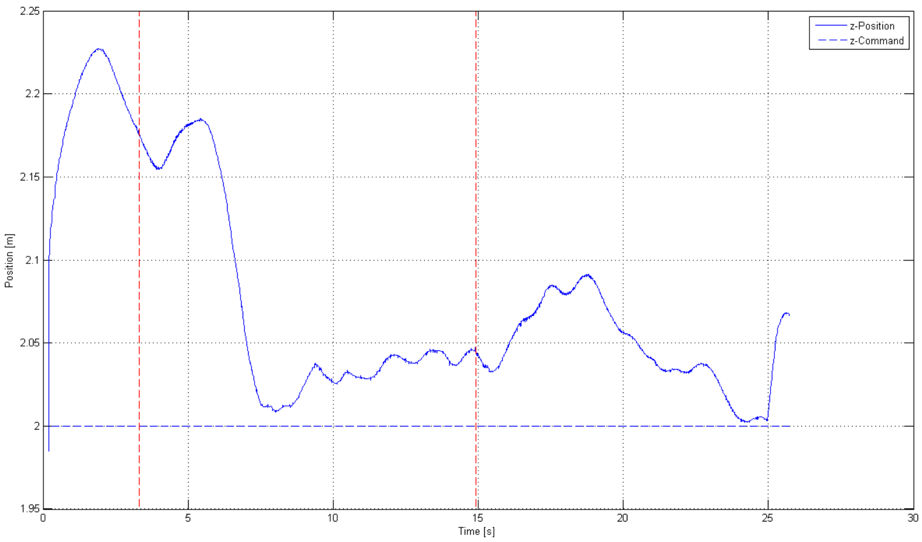


Figure 4.5: The z-position over the recorded test flight

4.2.2 Roll rate noise

As discussed in section 3.4.3, the roll rate noise was too noisy in simulation to be of use for the controller. This was also confirmed by the real flight tests: the D-value of the controller had to be set to zero as it was too noisy for the controller. However the physical aerodynamic damping provided by the aircraft was big enough to replace the D-value of the controller, thus still yielding a well damped overall system. This also shows that the aerodynamic damping should be considered during modeling and simulation.

Also I want to emphasize that this problem can be addressed by using an appropriate filter e.g. a Kalman filter.

4.2.3 Unstable Oscillation in z-Axis

The initial flight tests unexpectedly showed a unstable behavior in height/z-axis as shown in figure 4.7:

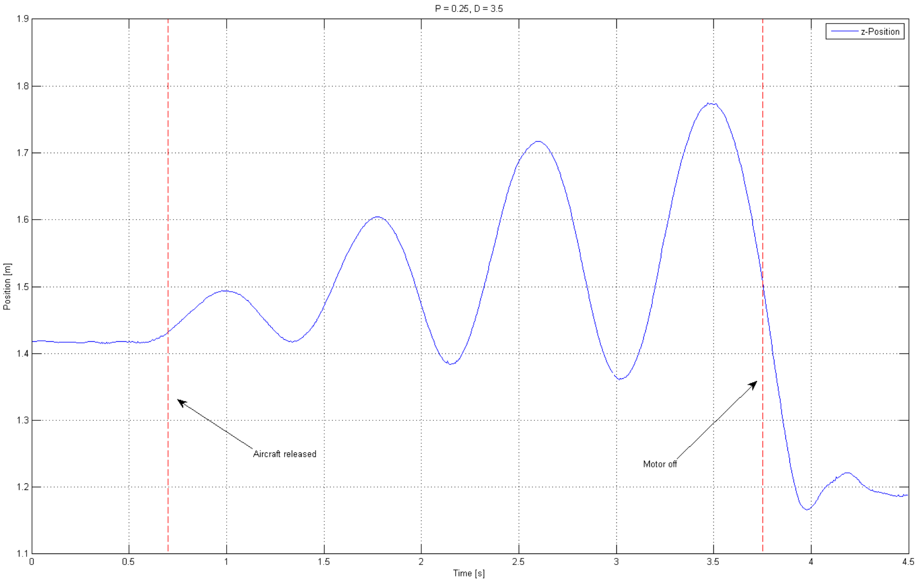


Figure 4.6: The instability in z-position, with a kP-gain of 0.25, and kD-gain 0.7.

After release of the airplane at time $t = 0.6s$, the airplane began a unstable oscillation in height with a estimated frequency of $f \approx 8.8Hz$. Eventually the test was aborted after 2 seconds, as the oscillations became to large. After the motor was switched off, the aircraft fell into the tether of the teststand, which is reason why the z-position would not got beneath 1.1 meters. The attitude stabilization however seemed to work as expected, and managed to keep the attitude stable throughout the entire test. Despite our efforts to manually tune the PD gain to acquire a stable system, the aircraft still displayed the same z-instability.

I conducted several simulations to proof the theory that these oscillations are induced by the system delays introduced by the finite computational speed of the Vicon system and controller calculations. However the simulations showed that the oscillations cannot be explained by adding a delay in signal transmission.

Simulation of unstable z-oscillation

As the attitude stabilization seemed to work, I only simulated the z-axis with the model derived in chapter 2 with an added delay. The design of the simulation also accounts for the discrete update rate of the controller of 50 Hz, with a sample and hold device. Also a variable delay was added to vary over different delay times.

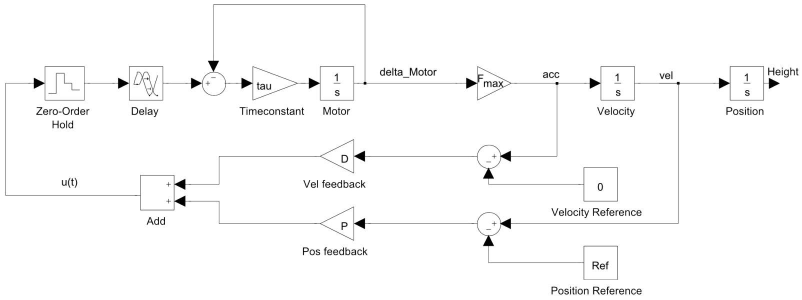


Figure 4.7: Schematic drawing of the simulation to understand the unstable z behavior

Two specific simulation results are shown in figure 4.8 and 4.9. The PD settings are the same as used during the flight tests shown in figure 4.7, however the first simulation uses a delay of 0.04 seconds, while the second one uses a delay of 0.06 seconds. While the first simulation is capable of stabilizing the aircraft despite the command delay, the second simulation yields a critically stable behavior of the plant. Setting the delay to values as high as 0.2 seconds does not significantly change the situation from the simulation shown in 4.9, as the system still displays a critically stable behavior. Therefore, this simulation shows that the instability in z cannot be explained with the addition of a pure delay.

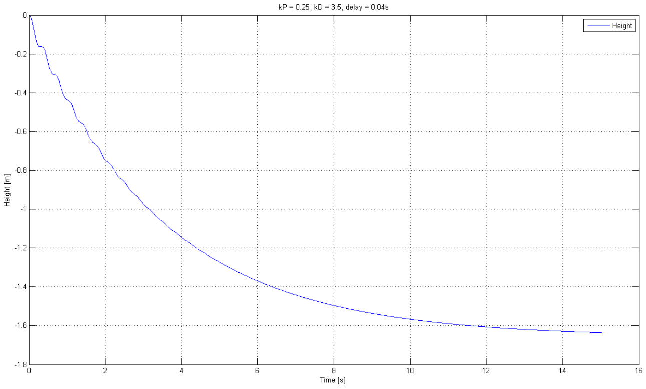


Figure 4.8: Simulation result for a delay of 0.04 seconds

For the flight tests, we finally managed to stabilize the z-position by installing a feedforward term, which sets the motor input to the equilibrium input, while the controller was tuned to very low PD-values. This approach worked well, as it managed to stabilize the z-position, however the feedforward term had to be constantly adjusted throughout the flight tests due to the decrease in battery voltage during operation.

I speculate that the instability in z, did not show up in simulation due to our inaccurate model of the motor, which simply neglected all motor dynamics. Also it is not accurate that the force generated by the motor is proportional to the input to it. The motor controller however does incorporate a feature which is called the “Governing Mode“, which in turn employs a PI controller, to keep the RPM of the motor proportional to the motor. It might be that this mode

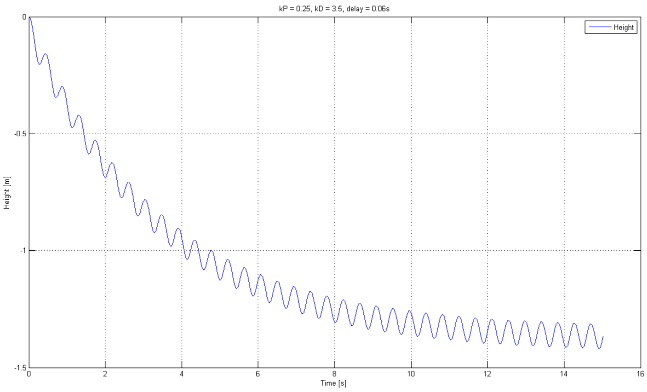


Figure 4.9: Simulation result for a delay of 0.06 seconds

is the solution to the instability in z, however because of time constraints I was not able to test this possibility.

4.2.4 Results and Outlook

Though the real system worked as planned at the end of the thesis, a couple of things should be addressed to improve current performance and to allow further steps towards aerobatic flight. A summary of identified problems together with a possible solutions will be presented below:

- All attitude axes depend on the correct initialization of the Vicon system. However the static hover data does reveal initialization errors, like shown in the previous section. An algorithm to estimate these off-initializations can be implemented, to gain a more exact initialization of the Vicon system and therefore to improve overall performance of the system.

- The coupling effects between motor and deflection of various control surfaces is very well visible in data and in real flight. A feedforward term based on the nonlinear model could improve the tracking behavior of the z-controller.

- One of the most significant nonlinearities of the model is the coupling between motor input and force generated by control surfaces. As seen in section 2 the force generated by the control surfaces is multiplied by the motor input. In hover however the PD gains where tuned for the equilibrium case, in which the motor input is constant. When the motor input differed, because of smaller or larger disturbances or steps in the reference, the gains where mistuned. From the equations we know that the force produced by the actuators is a linear function of the motor input. Therefore a linear gain scheduling based on the motor input would very likely improve the attitude performance in hover for different motor inputs.

- The aerodynamic model of the system was not confirmed for the aircraft used. Therefore a thorough system identification should be conducted. If the motor forces and moments are correctly identified, the equations presented in section 3.2 can be used to verify the rest of the model.

- The simulator model does not include inherent aerodynamic damping due to body rates. This however, is a major contributor to the angular dynamics as seen in section 4.2. This should be modeled and included into the simulator.
- Simulation and reality test showed that the roll rate estimation based on Vicon data cannot be used for control. To be able to better predict the roll rate, several possibilities exist: For example onboard electronics with gyroscopes could be used for this task, though it would add a lot of weight to the system. Also the implementation of a kalman filter would be feasible.

4.2.5 Concluding Remarks

Although some basic work was performed on the overall task of controlling the system at hand, a lot more work remains to be done to fully show the aircrafts capabilities. Very dynamic maneuvers are possible by exploiting the coupling between motor input and force generated by control surfaces. This makes it possible to fly extreme fast maneuvers and sharp turns which seem almost surreal as seen on some internet videos. Also conventional aerobatics are still to be shown in the FMA. Future work towards aerobatics can begin now, based on the fundamentals laid out in this thesis.

Appendix A

Parameters

Table A.1: Estimated and measured Parameters

Parameter	Estimation	Measurement	Unit
	Physical Parameters		
ρ	1.225		$[\frac{Kg}{m^3}]$
g	9.81		$[\frac{m}{s^2}]$
	Distances		
l_h	0.7	0.5	$[m]$
l_r	0.7	0.5	$[m]$
l_{wing}	0.05	0.05	$[m]$
l_{body}	0.05	0.18	$[m]$
h_e	0.1	0.09	$[m]$
h_r	0.1	0.11	$[m]$
h_w	0.13	0.2	$[m]$
D_{prop}	8*0.0254		$[m]$
$wingspan$	0.84	0.84	$[m]$
	Areas		
A_{wing}	0.11	0.14	$[m^2]$
A_{body}	0.11	0.0945	$[m^2]$
A_{elev}	0.04	0.04	$[m^2]$
A_{rudd}	0.04	0.0314	$[m^2]$
A_{prop}	0.0324		$[m^2]$
	Masses/Inertias		
m	0.15	0.174	$[Kg]$
I_{xx}	0.0162	0.001028859	$[Kg \cdot m^2]$
I_{yy}	0.0162	0.009943845	$[Kg \cdot m^2]$
I_{xx}	0.0162	0.010756274	$[Kg \cdot m^2]$
	Motor force		
F_{max}	3,8259		$[N]$

Appendix B

PD Values

Table B.1: PID values used in 3D simulation and flight tests

	Navigational loop			
	Simulation		**Flight Test**	
	kP	kD	kP	kD
Motor	1	0.7	0.1	0.7
Elevator	0.1	2	0.1	0
Aileron	-	-	-	-
Rudder	0.1	2	0.1	0
	Attitude loop			
	Simulation		**Flight Test**	
	P	D	P	D
Motor	-	-	-	-
Elevator	2	0.4	2.5	0
Aileron	4	2	4	0
Rudder	2	0.4	2.5	0

I used two different sets of PD values for the Navigational loop and the Attitude loop. Both sets of variables perform the same task, however the computation of the command is performed differently. Both sets can be transformed into the other variable set by the following transformation:

$$u(t) = P \cdot e_x + D \cdot e_v = k_P \cdot (e_x + k_D \cdot e_v) \tag{B.1}$$

$$P = k_P \tag{B.2}$$

$$D = k_P \cdot k_D \tag{B.3}$$

Bibliography

[1] Rick E. Cory. Supermaneuverable perching. *PhD Thesis*, 2010.

[2] Richard Dorf and Robert Bishop. *Modern Control Systems, Eleventh Edition.* Prentice-Hall, 2008.

[3] L. Guzzella. *Modeling and Analysis of Dynamic Systems.* ETH Zurich, 2008.

[4] Barnes W. McCormick. *Aerodynamics, Aeronautics and Flight Mechanics.* John Wiley and & sons, 1979.

[5] Ira M. Cohen Pijush K. Kundu. *Fluid Mechanics, Fourth Edition.* Academic Press, 2008.

[6] Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation.* John Wiley and & sons, 2003.

[7] Hacker motor page, `http://www.hacker-motor.com/daten/katalog/2011/Hacker_Antriebstabellen.pdf`.

[8] Torcster servos page, `http://www.torcster.de/Servos:::4.html`.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
Prof. Dr. R. D'Andrea, Prof. Dr. L. Guzzella

Title of work:

Integrating fixed wing UAVs in the FMA

Thesis type and date:

Semester Thesis, March 2012

Supervision:

Mark Müller
Prof. Dr. D'Andrea

Student:

Name:	Philippe Petit
E-mail:	ppetit@student.ethz.ch
Legi-Nr.:	08-935-199
Semester:	2

Statement regarding plagiarism:

By signing this statement, I affirm that I have read the information notice on plagiarism, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Information notice on plagiarism:

`http://www.ethz.ch/students/semester/plagiarism_s_en.pdf`

Zurich, 19. 8. 2012: ______________________________