Technische Universität München

Institute of Flight System Dynamics

Model Filtering for an INDI Controlled UAV

–

Modellfilterung für ein INDI Geregeltes UAV

Masterarbeit

Autor: Maximilian Söpper

Immatrikulationsnummer: 03637356

Betreuer: M.Sc. Stefan Raab

Juni 2018

Eidesstaatliche Erklärung

Hiermit erkläre ich, Maximilian Söpper, gegenüber dem Lehrstuhl für Flugsystemdynamik der Technischen Universität München, dass ich die vorliegende Masterarbeit selbstständig und ausschließlich unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen angefertigt habe.

Die Arbeit wurde in gleicher oder ähnlicher Form an keiner anderen Hochschule oder Universität vorgelegt.

Garching, 06.06.2018

Maximilian Söpper

Kurzfassung

Diese Masterarbeit

Abstract

This master thesis blablabla

Contents

List of Figures vi

List of Tables viii

List of Acronyms x

List of Symbols xii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 State of the Art . . . 1
- 1.3 Contribution of the Thesis . . . 1
- 1.4 Structure of the Thesis . . . 1

2 Theoretical Background 2
- 2.1 State Estimation with Kalman Filters . . . 2
  - 2.1.1 Stochastic Noise Processes . . . 2
  - 2.1.2 Linear Kalman Filtering . . . 3
  - 2.1.3 Nonlinear Kalman Filtering . . . 6
- 2.2 Dynamic Inversion . . . 7
  - 2.2.1 Relative Degrees . . . 8
  - 2.2.2 Nonlinear Dynamic Inversion . . . 8
  - 2.2.3 Incremental Nonlinear Dynamic Inversion . . . 9

3 Closed Loop Model of the UAV 11
- 3.1 Nomenclature and Coordinate Frames . . . 11
  - 3.1.1 Coordinate Frames and Angles . . . 11
  - 3.1.2 Kinematic Notation . . . 12
  - 3.1.3 Mathematical Preliminaries . . . 14
- 3.2 Plant Model . . . 14
  - 3.2.1 Equations of Motion . . . 14
  - 3.2.2 Actuator Model . . . 20
  - 3.2.3 Sensor Model . . . 20
  - 3.2.4 Wind Modeling . . . 21

3.2.5 Numeric Plant Data . . . 21
3.3 INDI Controller Model . . . 21
3.3.1 General Overview and Concepts . . . 22
3.3.2 Reference Models and Error Controller . . . 24
3.3.3 Control Allocation . . . 26
3.3.4 Onboard Plant Estimation . . . 28
3.3.5 Numeric Controller Data . . . 31

4 Requirements Capture **32**
4.1 Linear Closed Loop Performance Measures . . . 32
4.1.1 Linearization of the Closed Loop . . . 32
4.1.2 Trimming of the Closed Loop . . . 36
4.1.3 Uncertainty Modelling for the Linear Closed Loop . . . 36
4.1.4 Robustness Analysis . . . 39
4.2 Nonlinear Performance Measures . . . 40
4.2.1 Simulation of the Nonlinear Closed Loop . . . 40
4.2.2 Filter Covariance Consistency . . . 43
4.2.3 Nonlinear Closed Loop Performance Measure . . . 44
4.2.4 Nonlinear Robustness Assessment . . . 45
4.3 Online Parameter Estimation . . . 47
4.3.1 Impact of Parameters on Controller Performance . . . 47
4.3.2 Observability of Parameters . . . 49
4.4 Practical Considerations . . . 51
4.4.1 Filter Logics . . . 51
4.4.2 Filter Diagnostics . . . 51
4.4.3 Computational Cost . . . 52

5 Implementation of the Model Filter **53**
5.1 Overall Strategy and Ideas . . . 53
5.2 Kalman Filter Equations . . . 54
5.2.1 Propagation . . . 54
5.2.2 Correction . . . 55
5.3 Measurement and Process Noise Covariance . . . 55
5.3.1 Measurement Noise Covariance $\boldsymbol{R}$ . . . 56
5.3.2 Measurement Noise Covariance $\boldsymbol{Q}$ . . . 56
5.4 Filter Switches and Logics . . . 66
5.5 Filter Initialization . . . 68
5.6 Filter Diagnostics . . . 68
5.7 Reducing Computational Cost . . . 69
5.7.1 Sequential Correction . . . 69
5.7.2 Neglecting State Covariance Propagation . . . 70

6 Results **72**
6.1 Nominal Performance 72
6.2 Robustness 72
6.2.1 Parametric Uncertainty Robustness 72
6.2.2 Input Robustness 72
6.2.3 Output Robustness 72
6.3 Covariance Consistency 72
6.4 Computational Cost Reduction 72

7 Resumee **73**

8 Future Works **74**

Appendix **75**

List of Figures

3.1 Vehicle configuration and coordinate systems 12
3.2 Lift Coefficient over Angle of Attack 17
3.3 Control Effectors 19
3.4 Onboard plant estimation structure 28
3.5 Model filter structure 30

4.1 Uncertainty Modeling with the ΔM analysis model 37
4.2 Translational States of the Baseline Simulation 41
4.3 Pitch Channel of the Baseline Simulation 42
4.4 Main Propellers 43
4.5 Observability of different parameter estimation schemes 49
4.6 Observability of a combined parameter estimation scheme 50

5.1 Distribution of parametric uncertainty for the state time derivatives during the transition 62
5.2 Distribution of parametric uncertainty for the state time derivatives during braking 63
5.3 Distribution of parametric uncertainty for the state time derivatives during the transition. The uncertainty due to the estimated parameters (wind and effectiveness of the main propellers) is neglected here. 64
5.4 Distribution of parametric uncertainty for the state time derivatives during braking. The uncertainty due to the estimated parameters (wind and effectiveness of the main propellers) is neglected here. 65

List of Tables

3.1 Sensor and INS Emulation Data 21
3.2 Numeric Weight Balance and Geometry Data 22
3.3 Numeric Environment Data 22
3.4 Numeric Aerodynamic Data 22
3.5 Numeric Control Effector Dynamics Data 23
3.6 OBPM Perturbation Data 30
3.7 Numeric Reference Model and Error Controller Data 31
3.8 Numeric Control Effector Estimation Dynamics Data 31
3.9 Numeric Complementary Filter Data 31

4.1 Weights for Performance Measure 45
4.2 Expected Standard Deviation of Parameters 46
4.3 Parameter Impact Analysis 48

List of Acronyms

Acronym	Description
A	blablabla

List of Symbols

Latin Letters		
Symbol	**Unit**	**Description**
a	-	General matrix element
Greek Letters		
Symbol	**Unit**	**Description**
a	-	General matrix element
Indices		
Symbol	**Unit**	**Description**
a	-	General matrix element

Chapter 1

Introduction

1.1 Motivation

1.2 State of the Art

1.3 Contribution of the Thesis

1.4 Structure of the Thesis

Chapter two introduces the main theoretical backgrounds for this thesis, namely state estimation with Kalman filers and incremental nonlinear dynamic inversion. In chapter three the closed loop model of the controlled vehicle is the focus. Therefore, the equations of motion are derived and the corresponding INDI controller is introduced. Building up on chapter three, chapter four gives a requirements analysis for the included model filter. The fifth chapter gives the implementation of the model filter equations also concerning reduction of computational cost. Chapter six gives the results of the introduced requirements and performance analysis in chapter three. The main conclusions of the findings are summarized in chapter seven. Chapter eight proposes open points for future works. The appendix contains additional mathematical derivations or formulas which were not included in the man chapters for the sake of space consumption.

Chapter 2

Theoretical Background

This chapter focusses on the two main theoretical backgrounds of this thesis. The first one is state and parameter estimation with Kalman fitlers. The second one is the control strategy of incremental nonlinear dynamic inversion.

2.1 State Estimation with Kalman Filters

This section foucusses on the basics of Kalman filtering. As the stochastic process of white noise is crucial for the idea of Kalman filtering it is introduced first. Afterwards this section introduces linear Kalman filtering and finally explains extended Kalman filtering.

2.1.1 Stochastic Noise Processes

There exist several stochastic noise processes. Since gaussian distributed white noise plays an important role in Kalman filtering it shall be introduced here.

A continuous stochastic process is the generalization of a random variable to a time dependent random variable given by a random experiment [**?**]. For the special case of a white noise stochastic process there are two conditions that must be satisfied. Form now on, let $\boldsymbol{w}(t)$ denote variable following the white noise stochastic process. First of all, the mean of $\boldsymbol{w}(t)$ has to equal zero. The second condition for a continuous white noise stochastic process is that the stochastic process is uncorrelated. This means that the autocovariance is only nonzero for a zero time shift:

$$\operatorname{cov}(t,\tau) = \mathbb{E}\Big[\left(\boldsymbol{w}(t) - \mathbb{E}\left[\boldsymbol{w}(t)\right]\right)\left(\boldsymbol{w}(t+\tau) - \mathbb{E}\left[\boldsymbol{w}(t+\tau)\right]\right)^T\Big] = \boldsymbol{Q}_c(t)\delta(\tau) \tag{2.1}$$

where $\mathbb{E}$ denotes the expectation, $\boldsymbol{Q}_c(t)$ the process covariance and $\delta(\tau)$ the Dirac Delta function defined by:

$$\int_a^b \delta(\tau)d\tau = \begin{cases} 1, & a \leq 0 \leq b \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

In order to be gaussian white noise the random variable at a single point in time has to follow a gaussian normal distribution:

$$\boldsymbol{w}(t) = \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{Q}_c(t)\right) \tag{2.3}$$

2.1.2 Linear Kalman Filtering

The linear Kalman filter or also called best linear unbiased estimator [**?**] estimates the states of a stochastic linear system in an optimal manner. The optimality referes to reducing the trace of the covariance of the estimation error for a linear system.

Linear Time Invariant Model Dynamics

Let the linear system be a continuous time invariant linear system [**?** , p. 233-234] of the form

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\dot{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{w} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v} \end{aligned} \tag{2.4}$$

where the process noise $\boldsymbol{w}$ and measurement noise $\boldsymbol{v}$ follow white gaussian process noise with the continuous time covariances $\boldsymbol{Q}_c$ and $\boldsymbol{R}_c$ respectively. The process noise $\boldsymbol{w}$ represents uncertainties in the assumed dynamics of the system whereas the measurement noise $\boldsymbol{v}$ represents uncertainties in the measurements, for example sensor white noise.

Since Kalman filters operate onboard with a sampletime Δt the following derivations are carried out in a discrete time domain. This requires a transformation of the continuous model dynamics to a discrete time model dynamics as well. The discrete time model dynamics of equation (2.4) is

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{G}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} \\ \boldsymbol{y}_k &= \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \end{aligned} \tag{2.5}$$

with k denoting the discrete time index. The matrices $\boldsymbol{F}$, $\boldsymbol{G}$ and $\boldsymbol{H}$ in equation 2.5 follow

$$\begin{aligned} \boldsymbol{F} &= e^{\boldsymbol{A}\Delta t} \\ \boldsymbol{G} &= \left(e^{\boldsymbol{A}\Delta t} - \boldsymbol{I}\right)\boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{H} &= \boldsymbol{C} \end{aligned} \tag{2.6}$$

The discrete time process and measurement noise still follow gaussian white noise charcteristics. The covariance of the white noise processes of the discrete time system transforms according to

$$\begin{aligned} \boldsymbol{Q} &\approx \boldsymbol{Q}_c \Delta t \\ \boldsymbol{R} &\approx \frac{\boldsymbol{R}_c}{\Delta t} \end{aligned} \tag{2.7}$$

Prediction

A Kalman filter uses two steps to estimate the states of a dynamic system. In the first step the Kalman filter calculates a model based estimate of the current state. In the second step the Kalman filter performs a data fusion of the model based estimate and a current measurement. The data fusion is based on uncertainty measures of the estimated states which are given by state estimation error covariances. For this reason the Kalman filter also estimates corresponding covariances.

The first step is called prediction or propagation step and propagates the estimated state and the estimated state covariance to the time point of the next measurement [**?** , p. 124-129].

Let $\boldsymbol{x}_{k|k}$ denote the estimated state $\boldsymbol{x}_k$ and $\boldsymbol{P}_{k|k}$ the estimated state covariance $\boldsymbol{P}_k$ given all measurements up to time step k. The prediction consequently yields the estimates for the next time step $\boldsymbol{x}_{k+1|k}$ and $\boldsymbol{P}_{k+1|k}$ given all measurements up to time step k. The prediction follows the propagation of the stochastic estimated state through the model dynamics in equation (2.5).

$$\begin{aligned} \boldsymbol{x}_{k+1|k} &= \boldsymbol{F}\boldsymbol{x}_{k|k} + \boldsymbol{G}\boldsymbol{u}_k \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{F}\boldsymbol{P}_{k|k}\boldsymbol{F}^T + \boldsymbol{Q} \end{aligned} \tag{2.8}$$

Correction

The second step of the Kalman filter is called correction or measurement update. The correction combines the information of the prediction $\boldsymbol{x}_{k+1|k}$ and the measurement $\boldsymbol{y}_{k+1}$. The state correciton follows the same structure as a simple non stochastic Lüneberger observer:

$$\boldsymbol{x}_{k+1|k+1} = \boldsymbol{x}_{k+1|k} + \boldsymbol{K}_{k+1}\left(\boldsymbol{y}_{k+1} - \boldsymbol{H}\boldsymbol{x}_{k+1|k}\right) \tag{2.9}$$

where $\boldsymbol{K}_{k+1}$ denotes the observer gain, which is called Kalman gain. The Kalman gain determines the weight of the measurement compared to the prediction. In order to perform an uncertainty based data fusion of model prediction and sensor measurements, the Kalman gain depends on corresponding state error covariances. The Kalman gain is chosen to

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{P}_{k+1|k+1}\boldsymbol{H}^T + \boldsymbol{R}\right)^{-1} \tag{2.10}$$

This choice of the Kalman gain minimizes the trace of the predicted covariance $\boldsymbol{P}_{k+1|k+1}$. For this reason the Kalman filter is often referred to as an optimal filter [**? ?**]. As the correction step updates the state and includes measurement information, this also affects the error covariance of the corrected state. The state error covariance after the update using the Kalman gain from equation (2.10) is

$$\boldsymbol{P}_{k+1|k+1} = \left(\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H}\right)\boldsymbol{P}_{k+1|k} \tag{2.11}$$

Sequential Kalman Filtering

Since the online inversion in equation (2.10) is computationally expensive, sequential kalman filtering can reduce computational workload. For the sake of simplification, assume that the measurment covariance matrix is diagonal. In that case all measurements are independent of each other. The idea of the sequential correction is to use this independency to apply the correction sequentially for each measurement component individually [**?** , p. 150-154]. Let m denote the number of available measurements. For the sequential Kalman filter each correction step consists of m measurement updates with each update using just a single component of the measurement vector. For this procedure the sequential state and covariance estimates initialize as follows:

$$\begin{aligned} \boldsymbol{x}_{0,k+1} &= \boldsymbol{x}_{k+1|k} \\ \boldsymbol{P}_{0,k+1} &= \boldsymbol{P}_{k+1|k} \end{aligned} \tag{2.12}$$

Then, for $i = 1...m$ the componentwise update yields

$$\begin{aligned} \boldsymbol{x}_{i+1,k+1} &= \boldsymbol{x}_{i,k+1} + \boldsymbol{K}_{i,k+1}\left(\boldsymbol{y}_i - \boldsymbol{H}_i \boldsymbol{x}_{i,k+1}\right) \\ \boldsymbol{P}_{i+1} &= \left(\boldsymbol{I} - \boldsymbol{K}_{i,k+1}\boldsymbol{H}_i\right)\boldsymbol{P}_{i,k+1} \end{aligned} \tag{2.13}$$

with $\boldsymbol{H}_i$ denoting the row corresponding to the ith measurement, $\boldsymbol{x}_{i+1,k+1}$ and $\boldsymbol{P}_{i+1,k+1}$ the estimates after the ith sequential update and the Kalman gain $\boldsymbol{K}_{i,k+1}$ given by

$$\boldsymbol{K}_{i,k+1} = \boldsymbol{P}_{i,k+1|k}\boldsymbol{H}_i^T\left(\boldsymbol{H}_i\boldsymbol{P}_{i,k+1|k+1}\boldsymbol{H}_i^T + \boldsymbol{R}_i\right)^{-1} \tag{2.14}$$

with $\boldsymbol{R}_i$ being the ith component of the diagonal measurement covariance matrix.

After performing all m sequential updates the corrected state and state covariance are

$$\begin{aligned} \boldsymbol{x}_{k+1|k+1} &= \boldsymbol{x}_{m,k+1} \\ \boldsymbol{P}_{k+1|k+1} &= \boldsymbol{P}_{m,k+1} \end{aligned} \tag{2.15}$$

Latter formulation of sequential Kalman filtering relies on a diagonal shape of the measurement covariance matrix. For the case that the measurement covariance is not diagonal, diagonalization of $\boldsymbol{R}$ by transforming the measurement vector may be an option. It is important to notice that the diagonalization of $\boldsymbol{R}$ is only feasible for a constant measurement covariance. Otherwise an online diagonalization is required which is contradictory to the purpose of sequential Kalman filtering.

Steady State Kalman Gain

This subsection introduces the concept of the steady state kalman gain which will be used later for the linearization of the controller. The steady state Kalman gain is the gain a continuous Kalman filter reaches as it converges. The corresponding linear approximation of the estimator then equals a simple state observer structure:

$$\dot{\hat{\boldsymbol{x}}} = \hat{\boldsymbol{A}}\hat{\boldsymbol{x}} + \boldsymbol{K}\left(\boldsymbol{x} - \hat{\boldsymbol{x}}\right) + \hat{\boldsymbol{B}}\hat{\boldsymbol{u}} \tag{2.16}$$

where the $\hat{.}$ quantities denote the estimated nature of the quantities. In order to calculate the steady state Kalman gain the continuous algebraic ricati equation has to be solved for the steady state error covariance $\hat{\boldsymbol{P}}$:

$$\hat{\boldsymbol{A}}^T\hat{\boldsymbol{P}} + \hat{\boldsymbol{P}}\hat{\boldsymbol{A}} - \hat{\boldsymbol{P}}\boldsymbol{C}^T\boldsymbol{R}_c^{-1}\boldsymbol{C}\hat{\boldsymbol{P}} + \boldsymbol{Q}_c = \boldsymbol{0} \tag{2.17}$$

The continuous Kalman gain then is

$$\boldsymbol{K} = \hat{\boldsymbol{P}}\boldsymbol{C}^T\boldsymbol{R}_c^{-1} \tag{2.18}$$

Combined with equation (2.16) the Kalman filter can be linearized as a continuous system. The assumption here is that the filter has reached the steady state. Furthermore it is important to mention that the continuous representation of the filter does not completely match the dynamics of the real discrete filter. This simplification is only used in order to assess the linear robustness of the closed loop system.

2.1.3 Nonlinear Kalman Filtering

The challenge of nonlinear Kalman filtering is to cope with nonlinear system dynamics. Let the nonlinear continuous system be

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, t\right) \\ \boldsymbol{y} &= \boldsymbol{h}\left(\boldsymbol{x}, \boldsymbol{v}, t\right) \end{aligned} \tag{2.19}$$

Like for the linear case the process noise $\boldsymbol{w}$ and the measurement noise $\boldsymbol{v}$ follow zero mean normal distributions with covariance matrices $\boldsymbol{Q}_c$ and $\boldsymbol{R}_c$ respectively.

The Extended Kalman Filter

The idea of the Extended Kalman Filter (EKF) is to linearize the nonlinear system dynamics in equation (2.20) around the the current estimated state $\hat{\boldsymbol{x}}_k$ [**?** , p.400-409]:

$$\begin{aligned} \boldsymbol{A}_k &= \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_k} \\ \boldsymbol{L}_k &= \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}\right|_{\hat{\boldsymbol{x}}_k} \\ \boldsymbol{C}_k &= \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_k} \\ \boldsymbol{M}_k &= \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{v}}\right|_{\hat{\boldsymbol{x}}_k} \end{aligned} \tag{2.20}$$

The covariance propagation of the extended Kalman filter now uses latter linearization in order to apply the same covariance propagation as for the linear system. Therefore, the same discretization is applied in order to obtain the matrices $\boldsymbol{F}_k$ and $\boldsymbol{H}_k$.

$$\begin{aligned} \boldsymbol{F}_k &= e^{\boldsymbol{A}_k \Delta t} \\ \boldsymbol{H}_k &= \boldsymbol{C}_k \end{aligned} \tag{2.21}$$

Notice that the involved matrices now all depend on the time step which is denoted by the index k. Since the process and measurement noise have a nonlinear influence in equation (2.20), the corresponding linearizations have to be considered in the corresponding covariances in order to achieve the same system structure as in equation (2.5). The covariances of the discrete time system then are

$$\begin{aligned} \boldsymbol{Q}_k &= \boldsymbol{L}_k \boldsymbol{Q}_c \boldsymbol{L}_k^T \Delta t \\ \boldsymbol{R}_k &= \frac{\boldsymbol{M}_k \boldsymbol{R}_c \boldsymbol{M}_k^T}{\Delta t} \end{aligned} \tag{2.22}$$

Above equations allow for the same covariance propagation and a state and covariance correction as for the linear Kalman filter. The main difference is the time step dependency of the required matrices. The mentioned linearizations and discretizations only determine the covariance propagation, not the state propagation. The state is propagated with an arbitrary numeric integration scheme. This thesis uses the Euler forward integration scheme:

$$\boldsymbol{x}_{k+1|k} = \boldsymbol{x}_{k|k} + \boldsymbol{f}\left(\boldsymbol{x}_{k|k}, \boldsymbol{u}_k, t\right) \Delta t \tag{2.23}$$

Parameter Estimation with the EKF

The ability to handle nonlinear system dynamics allows the EKF to perform online parameter estimation. The idea is to extend the states with the parameters which leads to a new state vector

$$\boldsymbol{x}^{extended} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{p} \end{bmatrix} \tag{2.24}$$

Since parameters do not follow any modelled dynamics, the extended dynamics for this case is

$$\dot{\boldsymbol{x}}^{extended} = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{p}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}, \boldsymbol{w}, t\right) \\ \boldsymbol{w}^p \end{bmatrix}}_{\boldsymbol{f}^{extended}} \tag{2.25}$$

where $\boldsymbol{w}^p$ is a process noise for the parameters which helps avoiding convergence to a wrong parameter value. The extended process noise consequently is

$$\boldsymbol{w}^{extended} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{w}^p \end{bmatrix} \tag{2.26}$$

The extended process noise follows a normal distribution with an extended covariance

$$\boldsymbol{Q}^{extended} = \begin{bmatrix} \boldsymbol{Q}^{xx} & \boldsymbol{Q}^{xp} \\ \boldsymbol{Q}^{px} & \boldsymbol{Q}^{pp} \end{bmatrix} \tag{2.27}$$

where $\boldsymbol{Q}^{xx}$ denotes the covariance of the not extended process noise $\boldsymbol{w}$, $\boldsymbol{Q}^{pp}$ the covariance of the parameter process noise $\boldsymbol{w}^p$ and $\boldsymbol{Q}^{xp}$ the cross covariance between original and parameter process noise.

Linearizing the nonlinear system dynamics in equation (2.25) according to equation (2.20) leads to

$$\begin{aligned} \boldsymbol{A}_k^{extended} &= \begin{bmatrix} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\big|_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{p}}_k} & \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\big|_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{p}}_k} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{C}_k^{extended} &= \begin{bmatrix} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\big|_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{p}}_k} & \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{p}}\big|_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{p}}_k} \end{bmatrix} \\ \boldsymbol{L}_k^{extended} &= \begin{bmatrix} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}\Big|_{\hat{\boldsymbol{x}}_k} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \end{aligned} \tag{2.28}$$

Since the measurement noise is not affected by the expansion the matrix $\boldsymbol{M}_k$ of equation (2.20) remains the same. The extended propagation and correction for parameter estimation follow as for the normal extended Kalman filter.

2.2 Dynamic Inversion

Incremental nonlinear dynamic inversion is a development of nonlinear dynamic inversion. Therefore this section first introduces nonlinear dynamic inversion and the underlying concept

of relative degrees. Then this idea is extended to the incremental nonlinear dynamic inversion (INDI) approach. The considered dynamic system to control is of the form

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) \end{aligned} \tag{2.29}$$

2.2.1 Relative Degrees

The relative degree of an output y_i is the minumum required number of time derivatives applied to the output y_i that yields a direct dependency of the output derivative on the input $\boldsymbol{u}$ [**?**]. The structure of the plant and controller considered in this thesis (see chapter 3) result in a relative degree of one. In order to simplify the equations a relative degree of one is used in the rest of this chapter. For a more general NDI and INDI controller definition see [**?**].

2.2.2 Nonlinear Dynamic Inversion

Assuming a relitve degree of one, the output time derivative of the system in equation (2.35) follows

$$\dot{\boldsymbol{y}} = \underbrace{\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})}_{\boldsymbol{a}(\boldsymbol{x})} + \underbrace{\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{G}(\boldsymbol{x})}_{\boldsymbol{B}(\boldsymbol{x})} \boldsymbol{u} \tag{2.30}$$

It is common to call the time derivative of the output a pseudo control γ which leads to

$$\boldsymbol{\gamma} = \boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \tag{2.31}$$

The idea of nonlinear dynamic inversion (NDI) is to chose the control $\boldsymbol{u}$ in a way that the plant achieves a desired pseudo control $\boldsymbol{\nu}$. Solving equation (2.31) for $\boldsymbol{u}$ leads to

$$\boldsymbol{u} = \boldsymbol{B}^{-1}(\boldsymbol{x}) \left[\boldsymbol{\nu} - \boldsymbol{a}\left(\boldsymbol{x}\right)\right] \tag{2.32}$$

Equation (2.32) is the dynamic inversion. The control $\boldsymbol{u}$ in above equation achieves a real pseudo control γ that equals a given desired pseudo control $\boldsymbol{\nu}$. Assume there exists a desired linear output dynamics of the form

$$\underbrace{\dot{\boldsymbol{y}}_{ref}}_{\nu} = \boldsymbol{A}_{ref} \left(\boldsymbol{y}_{cmd} - \boldsymbol{y}\right) \tag{2.33}$$

where $\boldsymbol{y}_{cmd}$ is a command to be tracked and $\boldsymbol{A}_{ref}$ specifies the linear desired reference dynamics. Applying equation (2.32) with a desired pseudo control following the desired linear reference dynamics in equation (2.33) imposes the linear reference dynamics on the output of the nonlinear system $\boldsymbol{y}$.

It is important to mention that the real dynamic system defined by $\boldsymbol{a}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ is not known exactly. There is only an approximate model of reality described by $\hat{\boldsymbol{a}}(\boldsymbol{x})$ and $\hat{\boldsymbol{B}}(\boldsymbol{x})$. With this assumption the real pseudo control follows

$$\boldsymbol{\gamma} = \boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\hat{\boldsymbol{B}}^{-1}(\boldsymbol{x}) \left[\boldsymbol{\nu} - \hat{\boldsymbol{a}}\left(\boldsymbol{x}\right)\right] \tag{2.34}$$

Errors in the assumed system behavior cause mismatches in the inversion and the real pseudo controll γ will not follow the desired pseudo control $\boldsymbol{\nu}$ anymore. For this reason nonlinear dynamic inversion strongly relies on accurate model knowledge of $\boldsymbol{a}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$. In constrast to NDI, incremental nonlinear dynamic inversion (INDI) does not rely on $\boldsymbol{a}(\boldsymbol{x})$. This advantage will be shown in the next subsection.

2.2.3 Incremental Nonlinear Dynamic Inversion

While NDI requires special structures of equation (2.35) in order to be able to invert the equation, incremental dynamic inversion (INDI) is able to handle systems which are not affine in the control. For this reason, let the more general nonlinear system dynamics be

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) \end{aligned} \tag{2.35}$$

Solving above equation for the pseudo control like in equation (2.30) yields

$$\boldsymbol{\gamma} = \underbrace{\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}_{\boldsymbol{F}(\boldsymbol{x},\boldsymbol{u})} \tag{2.36}$$

INDI does not invert above equation directly but uses a Taylor inversion with respect to time first. The achieved pseudo control in the next time step of the controller equals

$$\boldsymbol{\gamma}(t + \Delta t) = \boldsymbol{\gamma}(t) + \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial t} \Delta t + h.o.t. \tag{2.37}$$

where Δt denotes the sample time of the controller. From here on the sample time is assumed to be small so that the higher order terms vanish and only the linear terms remain. In this case the partial derivative can be rewritten

$$\frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial t} \Delta t = \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} \Delta t + \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}} \dot{\boldsymbol{u}} \Delta t = \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} + \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}} \Delta \boldsymbol{u} \tag{2.38}$$

where $\Delta \boldsymbol{x}$ denotes the state change in the next time step $\boldsymbol{x}(t + \Delta t) - \boldsymbol{x}(t)$ and $\Delta \boldsymbol{u}$ denotes the control change in the next time step $\boldsymbol{u}(t + \Delta t) - \boldsymbol{u}(t)$. Inserting above result into equation (2.37) yields

$$\underbrace{\boldsymbol{\gamma}(t + \Delta t) - \boldsymbol{\gamma}(t)}_{\Delta \boldsymbol{\gamma}} = \underbrace{\frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} \Delta \boldsymbol{x}}_{\approx \boldsymbol{0}} + \underbrace{\frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}}_{\boldsymbol{B}(\boldsymbol{x},\boldsymbol{u})} \Delta \boldsymbol{u} + h.o.t. \tag{2.39}$$

The effect of change in controls $\Delta \boldsymbol{u}$ on the pseudo control $\boldsymbol{\gamma}(t + \Delta t)$ is assumed to be a lot stronger than the effect of the change in states $\Delta \boldsymbol{x}$ on the pseudo control. This is the main assumption of the INDI control strategy. For this reason INDI neglects the first term in above equation resulting in

$$\Delta \boldsymbol{\gamma} = \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{u}) \Delta \boldsymbol{u} \tag{2.40}$$

Finally, INDI performs the inversion on above simplified linear equation. This leads to the control difference

$$\Delta \boldsymbol{u} = \boldsymbol{B}^{-1}(\boldsymbol{x}, \boldsymbol{u}) \Delta \boldsymbol{\gamma} \tag{2.41}$$

with $\Delta\boldsymbol{\gamma}$ being the desired change in the pseudo control. Note that the actuators have to be fast enough to provide the desired change $\Delta\boldsymbol{u}$ in order for the inversion in above equation (2.41) to work.

$$\Delta\boldsymbol{\gamma} = \boldsymbol{\nu} - \boldsymbol{\gamma} \tag{2.42}$$

Unlike NDI, INDI does not rely on the zero dynamics $\boldsymbol{a}(\boldsymbol{x})$ in equation (2.32). From an INDI perspective, the current pseudo control $\boldsymbol{\gamma}$ already contains that information. The challenge of INDI consequently is to be able to determine a good estimation of the current pseudo control $\hat{\boldsymbol{\gamma}}$. In contrast to NDI measurement information can provide for that.

Chapter 3

Closed Loop Model of the UAV

The depicted aircraft is a transition UAV with vertical takeoff and landing as well as classic wingborne capabilities. Two main propellers ensure vertical take off and landing while a front propeller, wings and an elevator produce the desired forces and moments in wingborne flight. The original 2-dimensional longitudinal model was developed in [**?**] and is extended here in order to introduce a model filter.

3.1 Nomenclature and Coordinate Frames

The nomenclature and coordinate frames follow [**?**] and will be shortly introduced here. Since the considered problem will be two-dimensional the entries corresponding to the third dimension will be neglected here.

3.1.1 Coordinate Frames and Angles

This subsection introduces the coordinate systems and the corresponding angles used in this thesis.

Earth Centerd Earth Fixed Frame: E-Frame

In the shape of this thesis the rotation of the earth will be neglected. For this simplification the E frame is seen as inertial frame with its origin in the earths origin. The x_E axis lies in the equatorial plane and points towards the Greenwich meridian. The z_E axis coincides with the axis of earth rotation. The y_E axis completes the right handed coordinate system.

North East Down Frame: O-Frame

The norht east down frame (NED) has its origin in the reference point of the aircraft R. The x_O and y_O exes lie parallel to the local geoid surface. The x_O axis shows towards the geographic north pole, the z_O axis is perpendicular to the local geoid surface and shows down. The y_O axis completes the right handed system. For the 2-dimensional application only the x_E and z_E components will be used. The lateral motion in y_E will be neglected.

Body Fixed Frame: B-Frame

The origin of the body fixed frame B is the reference point R of the aircraft.The x_B axis points towards the nose of the aircraft and the y_B axis towards the right wing. The z_B axis completes the right hand system. The main use of the B frame is to specify the geometry of the aircraft (e.g. positions of propellers, elevator and wings). For the 2-dimensional case the angle between B and O frame is just the pitch angle θ, see figure 3.1.

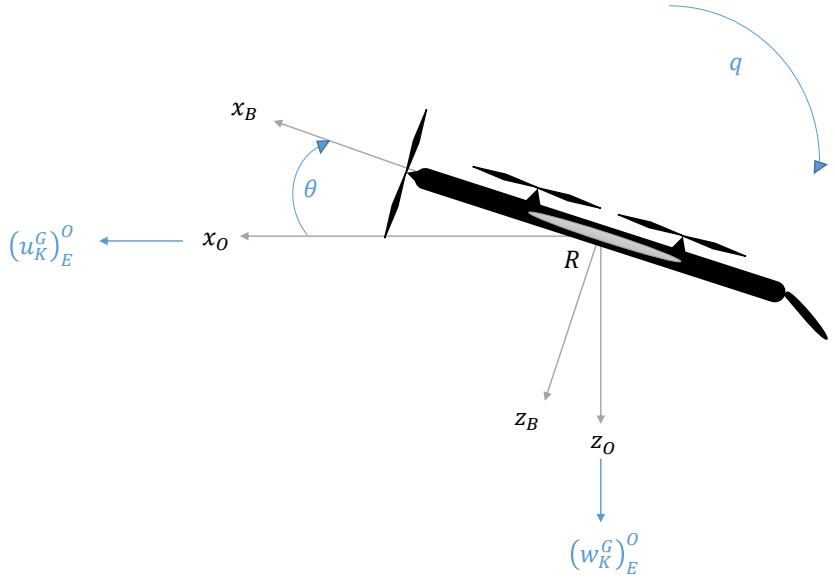

Figure 3.1: Vehicle configuration and coordinate systems

Aerodynamic Frame: A-Frame

As for the body fixed frame the origin of the A frame lies within the reference point R of the aircraft. The x_A axis points towards the aerodynamic velocity. For the 2-dimensional case the aerodynamic frame and the B frame are just rotated by the angle α_A. The angle between the O frame and the aerodynamic frame is the aerodynamic flight path angle γ_A.

3.1.2 Kinematic Notation

This subsection focusses on the kinematic notation and shows the nomenclature with simple examples

Notation of Points

The position of the reference point R in the body frame B is given by

$$\left(\boldsymbol{r}^R\right)_B = \begin{bmatrix} x^R \\ y^R \\ z^R \end{bmatrix}_B = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_B \tag{3.1}$$

The relative position of the reference point R and the point of the first propeller P_1 is given by

$$\left(\boldsymbol{r}^{RP_1}\right)_B = \begin{bmatrix} x^{RP_1} \\ y^{RP_1} \\ z^{RP_1} \end{bmatrix}_B \tag{3.2}$$

Notation of Velocities

Velocities are time derivatives of the position with respect to a certain frame. Let this frame be the earth centered earth fixed frame E here. Velocities also belong to a certain type (A(erodynamic), K(inematic), W(ind)). The wind velocity with respect to the E frame given in the NED frame O at the reference point reads

$$\left(\boldsymbol{v}_W^R\right)_O^E = \begin{bmatrix} u_W^R \\ v_W^R \\ w_W^R \end{bmatrix}_O^E \tag{3.3}$$

Notation of Accelerations

Since accelerations are time derivatives of velocities they require one further coordinate system for specification. The kinematic acceleration of the center of gravity G with respect to the E and O frame, given in the B frame is denoted as

$$\left(\boldsymbol{a}_K^G\right)_O^{EO} = \begin{bmatrix} \dot{u}_K^G \\ \dot{v}_K^G \\ \dot{w}_K^G \end{bmatrix}_O^{EO} \tag{3.4}$$

Notation of Angular Rates

Angular rates describe the relative rotation of two coordinate frames with respect to each other. The kinematic body fixed rates between O and B frame given in the body frame are given by

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \left(\omega_K^{OB}\right)_B = \begin{bmatrix} \omega_{x,K}^{OB} \\ \omega_{y,K}^{OB} \\ \omega_{z,K}^{OB} \end{bmatrix}_B \tag{3.5}$$

Notation of Angular Accelerations

Angular accelerations are time derivatives of angular rates with respect to a specified coordinate system. The angular accelerations of the body fixed rotation rates with respect to the B frame reads

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \left(\dot{\omega}_K^{OB}\right)_B^B \tag{3.6}$$

Notation of Forces

The acting point and the type specify a force. The types are A(erodynmic), P(ropulsive), G(ravitational) and T(otal). For example, the aerodynamic force acting in the reference point R given in the O frame is

$$\left(\boldsymbol{F}_A^R\right)_O = \begin{bmatrix} X_A^G \\ Y_A^G \\ Z_A^G \end{bmatrix}_O \tag{3.7}$$

Notation of Moments

As the forces, the moments also are specified by an acting point and a type. The propulsive moment acting in the reference point R given in the body fixed frame B follows

$$\left(\boldsymbol{M}_P^R\right)_B = \begin{bmatrix} L_P^R \\ M_P^R \\ N_P^R \end{bmatrix}_B \tag{3.8}$$

3.1.3 Mathematical Preliminaries

This subsection introduces mathametical preliminaries that will be used throughout this thesis, such as the hadamarad product for example.

Hadamarad Product

The hadamarad product is the elementwise product of two matrices. For the simple example of two 2×2 matrices the hadamarad product is

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \odot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix} \tag{3.9}$$

3.2 Plant Model

This section introduces the complete flight dynamics model with sensors and environmental influences.

3.2.1 Equations of Motion

This subsection derives the equations of motion, consisting of the rigid body equations, kinematics and modelled forces.

Rigid Body Equations

The considered model is just two-dimensional and only takes into account longitudinal motion. The involved states consequently are horizontal and vertical position and velocities as well as the pitch angle and pitch rate. In [**?**] all forces and moments were derived with respect to the center of gravity. This thesis uses a reference point R, which in the nominal case coincides with the center of gravity. The reason therefore is to investigate the behavior of the closed loop system also under uncertainty of the center of gravity. The weight balance and geometry of the vehicle relative to the reference point R is given in table 3.2. Considering the shift between reference point and center of gravity the time derivatives of the states follow

$$\underbrace{\begin{bmatrix} m & 0 & -m\left(r_y^{GR}\right)_B \\ 0 & m & m\left(r_x^{GR}\right)_B \\ -m\left(r_y^{GR}\right)_B & m\left(r_x^{GR}\right)_B & I_{yy}^R \end{bmatrix}}_{\mathbf{L}} \begin{bmatrix} \left(\dot{u}_K^G\right)_O^{EO} \\ \left(\dot{w}_K^G\right)_O^{EO} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \left(X_T^R\right)_O \\ \left(Z_T^R\right)_O \\ \left(M_T^R\right)_O \end{bmatrix} \tag{3.10}$$

The one dimensional strap down equation is

$$\dot{\theta} = q \tag{3.11}$$

This thesis neglects the influence of the position of the vehicle on the flight system dynamics. Therefore the position differential equations are decoupled:

$$\begin{aligned} \dot{x}_O^G &= \left(u_K^G\right)_O^E \\ \dot{y}_O^G &= \left(w_K^G\right)_O^E \end{aligned} \tag{3.12}$$

The total force and moment in equation (3.10) follow

$$\begin{aligned} \begin{bmatrix} \left(X_T^R\right)_O \\ \left(Z_T^R\right)_O \end{bmatrix} &= \begin{bmatrix} \left(X_A^R\right)_O \\ \left(Z_A^R\right)_O \end{bmatrix} + \begin{bmatrix} \left(X_P^R\right)_O \\ \left(Z_P^R\right)_O \end{bmatrix} + \begin{bmatrix} \left(X_G^R\right)_O \\ \left(Z_G^R\right)_O \end{bmatrix} \\ \left(M_T^R\right)_O &= \left(M_A^R\right)_O + \left(M_P^R\right)_O + \left(M_G^R\right)_O \end{aligned} \tag{3.13}$$

where A, P and G denote the cause of the force (aerodynamics, propulsion, gravitation).

Kinematics

The aerodynamic velocity follows as difference between kinematic velocity and the velocity of the wind

$$\begin{bmatrix} \left(u_A^G\right)_O^E \\ \left(w_A^G\right)_O^E \end{bmatrix} = \begin{bmatrix} \left(u_K^G\right)_O^E \\ \left(w_K^G\right)_O^E \end{bmatrix} - \begin{bmatrix} \left(u_W^G\right)_O^E \\ \left(w_W^G\right)_O^E \end{bmatrix} \tag{3.14}$$

Note that the aerodynamic velocity is assumed constant over the whole body here, meaning that the effect of the pitch rate q on aerodynamics is neglected. The aerodynamic flight path angle follows

$$\gamma_A^G = \operatorname{atan}\left(\frac{-\left(w_A^G\right)_O^E}{\left(u_A^G\right)_O^E}\right) \tag{3.15}$$

Now we can define the rotation matrixes between O, A and B frame

$$\begin{aligned} \boldsymbol{M}_{BO} &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \\ \boldsymbol{M}_{AO} &= \begin{bmatrix} \cos(\gamma_A) & -\sin(\gamma_A) \\ \sin(\gamma_A) & \cos(\gamma_A) \end{bmatrix} \\ \boldsymbol{M}_{BA} &= \begin{bmatrix} \cos(\alpha_A) & -\sin(\alpha_A) \\ \sin(\alpha_A) & \cos(\alpha_A) \end{bmatrix} \end{aligned} \tag{3.16}$$

The aerodynamic velocities in the body fixed frame are given by

$$\begin{bmatrix} \left(u_A^G\right)_B^E \\ \left(w_A^G\right)_B^E \end{bmatrix} = \boldsymbol{M}_{BO} \begin{bmatrix} \left(u_A^G\right)_O^E \\ \left(w_A^G\right)_O^E \end{bmatrix} \tag{3.17}$$

which yields the aerodynamic angle of attack

$$\alpha_A^G = \operatorname{atan}\left(\frac{\left(w_A^G\right)_B^E}{\left(u_A^G\right)_B^E}\right) \tag{3.18}$$

Aerodynamics

The aerodynamic forces and moments consists of three parts: wing aerodynamics, elevator aerodynamics and body aerodynamics. The modelled lift $\left(L^A\right)_A$ and drag $\left(D^A\right)_A$ of the aerodynamic surfaces follow

$$\begin{aligned} \left(L^A\right)_A &= \bar{q} S C_L \\ \left(D^A\right)_A &= \bar{q} S C_D \end{aligned} \tag{3.19}$$

where S is the surface area, C_L the lift coefficient C_D the drag coefficient and $\bar{q}$ the dynamic pressure following

$$\bar{q} = \frac{1}{2}\rho\left(\left[\left(u_A^G\right)_O^E\right]^2 + \left[\left(w_A^G\right)_O^E\right]^2\right) \tag{3.20}$$

The aerodynamic forces of an aerodynamic surface in the aerodynamic frame are then given by

$$\begin{bmatrix} \left(X_A^R\right)_A \\ \left(Z_A^R\right)_A \end{bmatrix} = -\begin{bmatrix} \left(D^A\right)_A \\ \left(L^A\right)_A \end{bmatrix} = -\bar{q} S \begin{bmatrix} C_D \\ C_L \end{bmatrix} \tag{3.21}$$

Turning into the body frame yields

$$\begin{bmatrix} \left(X_A^R\right)_B \\ \left(Z_A^R\right)_B \end{bmatrix} = -\bar{q} S \boldsymbol{M}_{BA} \begin{bmatrix} C_D \\ C_L \end{bmatrix} \tag{3.22}$$

In [**?**] the lift coefficient was modelled linear in the aerodynamic angle of attack. Since this thesis also investigates the influence of wind on the performance of the closed loop plant, the lift modelling will be adapted here. When the aircraft is hovering and exposed to tail wind, the

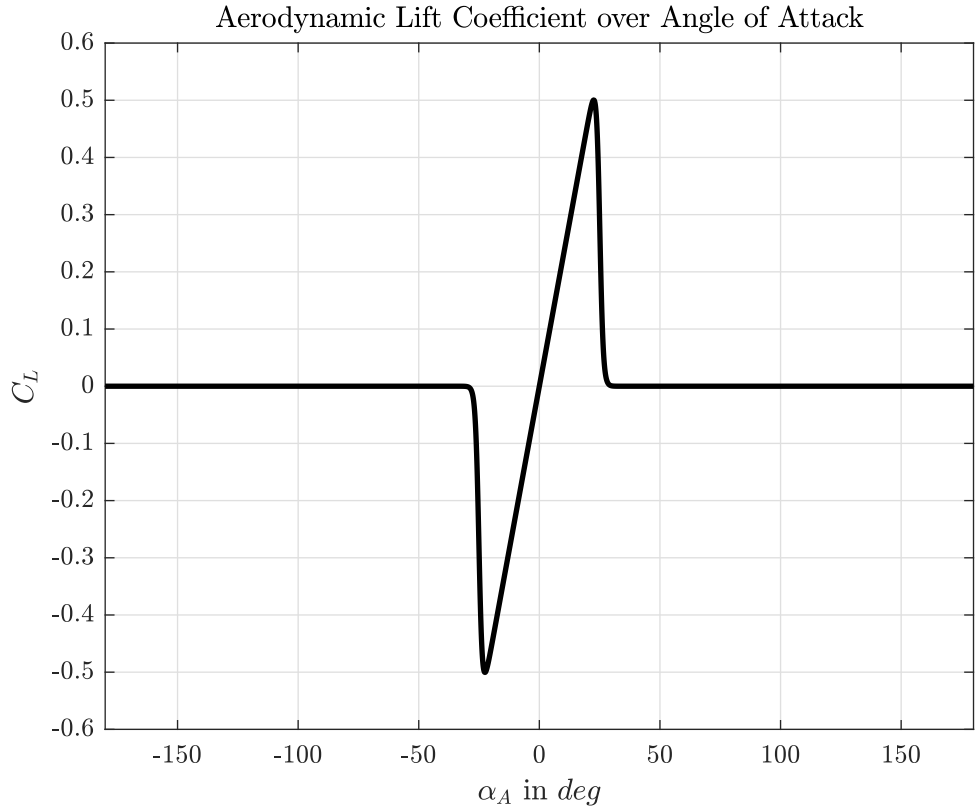

Figure 3.2: Lift Coefficient over Angle of Attack

aerodynamic angle of attack of the vehicle will be close to π which would first of all lead to very high lift coefficients and a discontinuous step at π because of the range for α_A $[-\pi...\pi]$. In order to prevent problems arising due to these tail winds (or also backward kinematic speeds) this thesis introduces an aerodynamic angle of attack based blending of the lift coefficient C_L. The blending λ_{C_L} follows a sigmoidal of the angle of attack

$$\lambda_{C_L} = \frac{1}{1 + e^{k(|\alpha_A| - \frac{25\pi}{180})}} \tag{3.23}$$

This blending now is included in the lift coefficients. The lift coefficient of the wing consequently is

$$\begin{aligned} C_{L,wing} &= C_{L\alpha,wing}\alpha_A\lambda_{C_L} \\ C_{D,wing} &= C_{D0,wing} + k_{wing}C_{L,wing}^2 \end{aligned} \tag{3.24}$$

The dependency of the lift coefficient on the aerodynamic angle of attack is shown in figure 3.2. The overall wing aerodynamic force in the body frame results in

$$\begin{bmatrix} \left(X_{A,wing}^R\right)_B \\ \left(Z_{A,wing}^R\right)_B \end{bmatrix} = -\bar{q}S_{wing}\boldsymbol{M}_{BA} \begin{bmatrix} C_{D0,wing} + k_{wing}C_{L,wing}^2 \\ C_{L\alpha,wing}\alpha_A\lambda_{C_L} \end{bmatrix} \tag{3.25}$$

Since the points A and R coincide, the resulting moment is zero.

$$\left(M_{A,wing}^R\right)_B = 0 \tag{3.26}$$

For simplification reasons, the elevator lift modelling uses the same lift blending as the wings, although the effective aerodynamic angle of attack at the elevator differs from the aerodynamic angle of atack by the elevator deflection η. With this simplification the elevator aerodynamic coefficients are

$$\begin{aligned} C_{L,elev} &= C_{L\alpha,elev}\left(\alpha_A + \eta\right)\lambda_{C_L} \\ C_{D,elev} &= C_{D0,elev} + k_{elev}C_{L,elev}^2 \end{aligned} \tag{3.27}$$

where η denotes the elevator deflection that changes the aerodynamic angle of attack at the elevator. The total aerodynamic force of the elevator consequently is

$$\begin{bmatrix} \left(X_{A,elev}^R\right)_B \\ \left(Z_{A,elev}^R\right)_B \end{bmatrix} = -\bar{q}S_{elev}\boldsymbol{M}_{BA}\begin{bmatrix} C_{D0,elev} + k_{elev}C_{L,elev}^2 \\ C_{L\alpha,elev}\left(\alpha_A + \eta\right)\lambda_{C_L} \end{bmatrix} \tag{3.28}$$

Since the aerodynamic reference point of the elevator and the reference point R do not coincide the moment of the elevator is

$$\left(M_{A,elev}^R\right)_B = \left(z^{RE}\right)_B\left(X_{A,elev}^R\right)_B - \left(x^{RE}\right)_B\left(Z_{A,elev}^R\right)_B \tag{3.29}$$

where $\left(x^{RE}\right)_B$ and $\left(z^{RE}\right)_B$ define the distance between elevator and reference point in the body frame.

The aerodynamic effect of the body is modelled as drag only

$$\begin{aligned} C_{L,body} &= 0 \\ C_{D,wing} &= C_{D0,wing} \end{aligned} \tag{3.30}$$

This results in the overall wing aerodynamic force in the body frame

$$\begin{bmatrix} \left(X_{A,body}^R\right)_B \\ \left(Z_{A,body}^R\right)_B \end{bmatrix} = -\bar{q}S_{body}\boldsymbol{M}_{BA}\begin{bmatrix} C_{D0,body} \\ 0 \end{bmatrix} \tag{3.31}$$

As for the main wing, the resulting moment is zero.

$$\left(M_{A,body}^R\right)_B = 0 \tag{3.32}$$

The total aerodynamic force in the north east down frame consequently is

$$\begin{bmatrix} \left(X_A^R\right)_O \\ \left(Z_A^R\right)_O \end{bmatrix} = \boldsymbol{M}_{BO}^T\left(\begin{bmatrix} \left(X_{A,wing}^R\right)_B \\ \left(Z_{A,wing}^R\right)_B \end{bmatrix} + \begin{bmatrix} \left(X_{A,elev}^R\right)_B \\ \left(Z_{A,elev}^R\right)_B \end{bmatrix} + \begin{bmatrix} \left(X_{A,body}^R\right)_B \\ \left(Z_{A,body}^R\right)_B \end{bmatrix}\right) \tag{3.33}$$

The total aerodynamic moment is just the moment of the elevator:

$$\left(M_A^R\right)_O = \left(M_{A,elev}^R\right)_B \tag{3.34}$$

Propulsion

Modelling the liftforce of the propellers as quadratic to the turn rate w_P yields for the total force of the first propeller:

$$\begin{bmatrix} \left(X_{P1}^{P1}\right)_B \\ \left(Z_{P1}^{P1}\right)_B \end{bmatrix} = \begin{bmatrix} k_{P1} w_{P1}^2 \\ 0 \end{bmatrix} \tag{3.35}$$

where k_{P1} denotes the propeller effectiveness. Since the reference point R lies on the axis of the first propeller the corresponding moment is zero:

$$\left(M_{P1}^{R}\right)_B = 0 \tag{3.36}$$

The second propeller is perpendicular to the first propeller and produces the forces

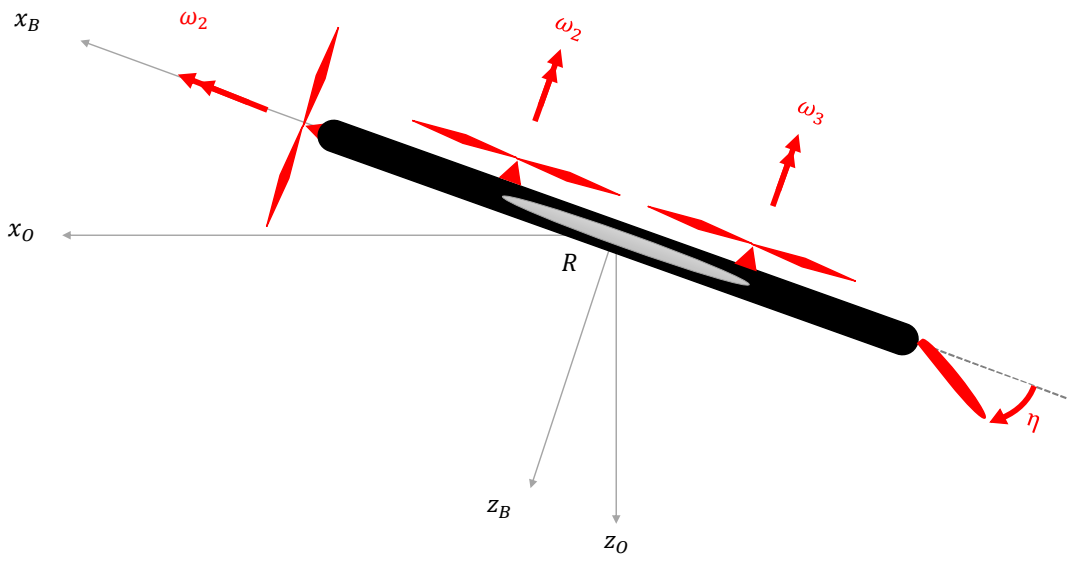

Figure 3.3: Control Effectors

$$\begin{bmatrix} \left(X_{P2}^{P2}\right)_B \\ \left(Z_{P2}^{P2}\right)_B \end{bmatrix} = \begin{bmatrix} 0 \\ -k_{P2} w_{P2}^2 \end{bmatrix} \tag{3.37}$$

and a pitch moment

$$\left(M_{P2}^{R}\right)_B = -\left(x_{RP2}\right)_B \left(Z_{P2}^{P2}\right)_B \tag{3.38}$$

The third propeller follows according to the second propeller:

$$\begin{bmatrix} \left(X_{P3}^{P3}\right)_B \\ \left(Z_{P3}^{P3}\right)_B \end{bmatrix} = \begin{bmatrix} 0 \\ -k_{P3} w_{P3}^2 \end{bmatrix} \tag{3.39}$$

with a pitch moment

$$\left(M_{P3}^{R}\right)_B = -\left(x_{RP3}\right)_B \left(Z_{P3}^{P3}\right)_B \tag{3.40}$$

The total propulsion force is

$$\begin{bmatrix} \left(X_P^R\right)_O \\ \left(Z_P^R\right)_O \end{bmatrix} = \boldsymbol{M}_{BO}^T \left(\begin{bmatrix} \left(X_{P1}^R\right)_B \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \left(Z_{P2}^R\right)_B \end{bmatrix} + \begin{bmatrix} 0 \\ \left(Z_{P3}^R\right)_B \end{bmatrix} \right) \tag{3.41}$$

The total propulsive moment equals

$$\left(M_P^R\right)_O = \left(M_{P2}^R\right)_B + \left(M_{P3}^R\right)_B \tag{3.42}$$

Gravitational Forces and Moments

The gravitational Force in the north east down frame (O) is

$$\begin{bmatrix} \left(X_G^R\right)_O \\ \left(Z_G^R\right)_O \end{bmatrix} = \begin{bmatrix} \left(X_G^G\right)_O \\ \left(Z_G^G\right)_O \end{bmatrix} = \begin{bmatrix} 0 \\ mg \end{bmatrix} \tag{3.43}$$

Turning into the body frame yields

$$\begin{bmatrix} \left(X_G^G\right)_B \\ \left(Z_G^G\right)_B \end{bmatrix} = \boldsymbol{M}_{BO} \begin{bmatrix} \left(X_G^G\right)_O \\ \left(Z_G^G\right)_O \end{bmatrix} \tag{3.44}$$

And the gravitational moment around the reference point equals

$$\left(M_G^R\right)_O = (z_{RG})_B \left(X_G^G\right)_B - (x_{RG})_B \left(Z_G^G\right)_B \tag{3.45}$$

3.2.2 Actuator Model

The dynamic model of the actuator is a second order system

$$u(s) = \frac{w_{0,u}^2}{s^2 + 2\zeta_u \omega_{0,u} s + \omega_{0,u}^2} \tag{3.46}$$

with commanded control u_c, real control u the natural frequency of the actuator $w_{0,u}$ and the corresponding relative damping ζ_u. Saturations in the controls and their time derivatives model the physical operational range:

$$\begin{aligned} u_{min} \leq u \leq u_{max} \\ \dot{u}_{min} \leq \dot{u} \leq \dot{u}_{max} \end{aligned} \tag{3.47}$$

3.2.3 Sensor Model

The dirt effects in the sensor models are gaussian white noise, bias terms and time delays. The main modelled dirt effect is the gaussian white noise with time discrete standard deviation σ_{WN} given in table 3.1. The second part is a constant bias that is assumed to follow a normal distribution

$$b = \mathcal{N}(0, \sigma_{bias}) \tag{3.48}$$

with the bias b and the standard deviation of the bias σ_{bias}. The last modelled sensor dirt effect are time delays in the different channels, which are also given in table 3.1.

Sensor Models and INS Emulation				
		White Noise	Bias	Time Delay
Channel	Unit	σ_{WN}	σ_{bias}	δt in s
$\left(u_K^G\right)_O^E$	m/s	0.1	0.1	0.1
$\left(w_K^G\right)_O^E$	m/s	0.1	0.1	0.1
θ	deg	0.1	0.1	0.1
q	deg/s	0.2	0.2	0.1
$\left(\dot{u}_K^G\right)_O^E$	m/s^2	0.03	0.03	0.1
$\left(\dot{w}_K^G\right)_O^E$	m/s^2	0.03	0.03	0.1

Table 3.1: Sensor and INS Emulation Data

3.2.4 Wind Modeling

The United States Military Specification [**?**] contains standards for wind models. Two very commonly used standards are the wind shear and the one minus cosine gust model. The shear model can describe the change of static wind speed over altitude. For the sake of simplification this thesis neglects wind shear in order to decouple the position differential equations. In order to model wind gusts the US Military Specification [**?**] contains the one minus cosine gust model. This gust model shows disadvantages in modelling wind gust disturbances in hover flight. In the case of no static wind and no kinematic velocity, a classic one minus cosine wind gust model will produce a zero wind gust. The reason therefor is, that the one minus cosine gust is a spatial gust, that only affects the aircraft, if there is a static relative velocity. In order to overcome these problems and to obtain a wind and wind gust model that covers the full operational range of the vehicle, wind and wind gusts are imposed in the north east down frame (O) as solely depending on time. This implementation can also provide one minus cosine like gusts. The spatial dependency (characterisitc length) of the original formulation just has to be exchanged with a corresponding time scale for the time series. Another way to implement a simple wind gust model are step inputs. in order to prevent sharp non physical edges the step inputs can be PT1 filtered.

3.2.5 Numeric Plant Data

3.3 INDI Controller Model

This section presents the onboard controller structure consisting mainly of the controller in [**?**]. The additional parts added for this thesis are a complementary and a kalman filter.

Weight Balance					
	m	I_{yy}			
Unit	kg	kgm^2			
	5	1			
Geometry					
	$\left(r^{GR}\right)_B$	$\left(r^{RE}\right)_B$	$\left(r^{RP_1}\right)_B$	$\left(r^{RP_2}\right)_B$	$\left(r^{RP_3}\right)_B$
Unit	m	m	m	m	m
x_B	0	-1	1	0.5	-0.5
z_B	0	0	0	-0.2	-0.2

Table 3.2: Numeric Weight Balance and Geometry Data

Environment		
	g	ρ
Unit	m/s^2	kg/m^3
	9.81	1.225

Table 3.3: Numeric Environment Data

Aerodynamic Surfaces				
	$C_{L\alpha}$	C_{D0}	k	S
Unit	-	-	-	m^2
Wing	1.3132	0.2	1	1
Elevator	1.3132	1	0.05	0.1
Body	0	0.4	0	0.2
Propeller Aerodynamics				
	Unit	P_1	P_2	P_3
k_P	kgm/rad^2	10^{-4}	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$

Table 3.4: Numeric Aerodynamic Data

3.3.1 General Overview and Concepts

The controller is a controller based on incremental nonlinear dynamic inversion. The inner loop follows an INDI structure and controls pseudo controls (translational accelerations and pitch accelerations). The outer loop consists of an error controller and a reference model containing

Propeller Dynamics						
	$w_{0,u}$	ζ_u	u_{min}	u_{max}	$\dot{u}_{min}$	$\dot{u}_{max}$
Unit	rad/s	-	rad/s	rad/s	rad/s^2	rad/s^2
P_1	30	1	1	1000	-500	500
P_2	30	1	1	500	-500	500
P_3	30	1	1	500	-500	500
Elevator Dynamics						
	$w_{0,u}$	ζ_u	u_{min}	u_{max}	$\dot{u}_{min}$	$\dot{u}_{max}$
Unit	rad/s	-	deg	deg	deg/s	deg/s
η	30	1	-45	45	-60	60

Table 3.5: Numeric Control Effector Dynamics Data

all plant states:

$$\boldsymbol{x} = \begin{bmatrix} \left(u_K^G\right)_O^E \\ \left(w_K^G\right)_O^E \\ \theta \\ q \end{bmatrix} \tag{3.49}$$

The outer loop controls the tracking of the pilot commands which are

$$\boldsymbol{y}_{cmd} = \begin{bmatrix} u_{cmd} \\ w_{cmd} \end{bmatrix} \tag{3.50}$$

In the following the concepts of virtual control inputs is explained, as this is crucial for the working principle of the controller.

Virtual Control Inputs

For high speed the transition UAV is supposed to turn off its main propellers (P_2, P_3) and use its aerodynamic surfaces in order to control the vertical channel. Controlling the height channel with the elevator requires a build up of a pitch rate by the elevator pitch moment. This pitch rate then builds up a pitch angle which has a direct influence on aerodynamic lift and vertical acceleration. This leads to a relative degree of two in the vertical channel. In hover mode the relative degree of the vertical velocity is only one since the propellers directly generate vertical acceleration. This means that the relative degree changes during operation. However, the derivation of the INDI controller in section 2.2 relies on a constant relative degree. Introducing a new control input can avoid this problem. For this plant the new control (also called virtual controls $\mathbf{v}_{cmd}$) is the pitch angle θ. Treating the pitch angle like a (virtual) control input reduces the relative degree of

the vertical channel in forward flight to one. Extending the controls yields

$$\begin{bmatrix} \boldsymbol{u} \\ \mathbf{v}_{cmd} \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \eta \\ \theta_{cmd} \end{bmatrix} \tag{3.51}$$

The given commanded virtual control θ_{cmd} is now treated like an external pilot command and introduced in the outer loop, more accurately the reference model:

$$\boldsymbol{x}_{cmd} = \begin{bmatrix} u_{cmd} \\ w_{cmd} \\ \theta_{cmd} \\ 0 \end{bmatrix} \tag{3.52}$$

with $\boldsymbol{x}_{cmd}$ being the commanded states for the reference model (see section 3.3.2)

3.3.2 Reference Models and Error Controller

Reference Models

The controller uses decoupled reference models for the two velocity channels and the pitch angle. The Reference model for the horizontal velocity follows the differential equation

$$\nu_u = a_{0,u} \left(u_{cmd} - u_{ref} \right) \tag{3.53}$$

$$\dot{u}_{ref} = \nu_u - \nu_{hedging,u} \tag{3.54}$$

Do not confuse the horizontal velocity component u with the controls $\boldsymbol{u}$. The quantity $\nu_{hedging,u}$ denotes the so called pseudo control hedging. The idea of pseudo control hedging is to slow down the reference model in order to compensate for lack of physical capability of the control effectors [**?**]. The pseudo control hedging only applies to the integration of states of the reference model, not to the desired pseudo control. The hedging signal is chosen to be the so called reaction deficit of the plant, which is nothing else but the desired incremental pseudo control:

$$\boldsymbol{\nu}_{hedging} = \Delta \boldsymbol{\gamma} \tag{3.55}$$

The reference model for the vertical channel follows accordingly

$$\nu_w = a_{0,w} \left(w_{cmd} - w_{ref} \right) \tag{3.56}$$

$$\dot{w}_{ref} = \nu_w - \nu_{hedging,w} \tag{3.57}$$

The reference model for the pitch dynamics is a second order system

$$\nu_\theta = a_{0,\theta} \left(\theta_{cmd} - \theta_{ref} \right) - a_{1,\theta} q_{ref} \tag{3.58}$$

$$\dot{\theta}_{ref} = q_{ref} \tag{3.59}$$

$$\dot{q}_{ref} = \nu_\theta - \nu_{hedging,\theta} \tag{3.60}$$

Combinging all equations above leads to the total fourth order system

$$\underbrace{\begin{bmatrix} \dot{u}_{ref} \\ \dot{w}_{ref} \\ \dot{\theta}_{ref} \\ \dot{q}_{ref} \end{bmatrix}}_{\dot{\boldsymbol{x}}_{ref}} = \boldsymbol{A}_{ref} \left(\underbrace{\begin{bmatrix} u_{ref} \\ w_{ref} \\ \theta_{ref} \\ q_{ref} \end{bmatrix}}_{\boldsymbol{x}_{ref}} - \underbrace{\begin{bmatrix} u_{cmd} \\ w_{cmd} \\ \theta_{cmd} \\ 0 \end{bmatrix}}_{\boldsymbol{x}_{cmd}} \right) - \boldsymbol{C}_\gamma^T \underbrace{\begin{bmatrix} \nu_{hedging,u} \\ \nu_{hedging,w} \\ \nu_{hedging,\theta} \end{bmatrix}}_{\boldsymbol{\nu}_{hedging}} \tag{3.61}$$

With the matrices

$$\boldsymbol{A}_{ref} = \begin{bmatrix} -a_{0,u} & 0 & 0 & 0 \\ 0 & -a_{0,w} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -a_{0,\theta} & -a_{1,\theta} \end{bmatrix} \qquad \boldsymbol{C}_\gamma^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.62}$$

The reference model pseudo control is

$$\boldsymbol{\nu}_{ref} = \boldsymbol{C}_\gamma \boldsymbol{A}_{ref} \left(\boldsymbol{x}_{ref} - \boldsymbol{x}_{cmd} \right) \tag{3.63}$$

Error Controller

The error controller has a PI structure. The proportional part is chosen according to the reference model dynamics. Let $\hat{\boldsymbol{x}}$ denote a current estimation of the real plant states

$$\boldsymbol{\nu}_{ec} = \boldsymbol{C}_\gamma \left(\boldsymbol{A}_{ref} \left(\hat{\boldsymbol{x}} - \boldsymbol{x}_{ref} \right) + \boldsymbol{A}_{integral} \int \hat{\boldsymbol{x}} - \boldsymbol{x}_{ref} dt \right) \tag{3.64}$$

with the integral gain $\boldsymbol{A}_{integral}$ only weighing steady state tracking errors in the velocities and pitch angle:

$$\boldsymbol{A}_{integral} = \begin{bmatrix} a_{-1,u} & 0 & 0 & 0 \\ 0 & a_{-1,w} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a_{-1,\theta} & 0 \end{bmatrix} \tag{3.65}$$

The total desired pseudo control is the sum of reference pseudo control and additional error controller pseudo control

$$\boldsymbol{\nu} = \boldsymbol{\nu}_{ref} + \boldsymbol{\nu}_{ec} \tag{3.66}$$

The structure of the error controller cancels the $\boldsymbol{A}_{ref}\boldsymbol{x}_{ref}$ in the total desired pseudo control.

Dynamic inversion

Given the desired pseudo control the incremental pseudo control follows as

$$\tilde{\boldsymbol{B}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) \begin{bmatrix} \Delta \boldsymbol{u} \\ \Delta \mathbf{v} \end{bmatrix} = \Delta \boldsymbol{\gamma} \tag{3.67}$$

with the increments

$$\begin{aligned} \Delta \boldsymbol{\gamma} &= \boldsymbol{\nu} - \hat{\boldsymbol{\gamma}} \\ \Delta \boldsymbol{u} &= \boldsymbol{u}_{cmd} - \hat{\boldsymbol{u}} \\ \Delta \mathbf{v} &= \mathbf{v}_{cmd} - \hat{\boldsymbol{v}} \end{aligned} \tag{3.68}$$

Equations (3.67) and (3.68) are solved for the commanded control $\boldsymbol{u}_{cmd}$. The first question arising from equation (3.67) is how to solve for the controls since $\tilde{\boldsymbol{B}}$ is not quadratic. There are five controls and three desired pseudocontrols. The equationsystem consequently is underdetermined. The main idea to solve this problem is to include additional constraints. The next subsection about control allocation focuesses on this problem.

The other question is how to determine all the required estimates (all quantities denoted with a $\hat{.}$). This question is the main focus of this thesis. The subsection about onboard plant estimation shows the originaly used approach [**?**] and how it will be adapted in this thesis.

3.3.3 Control Allocation

The depicted controller uses a pseudo inverse weighted control allocation with an additional nullspace transition in order to calculate the desired changes in the controls from equation (3.67).

General Approach

The first step of the control allocation is a weighted pseudoinverse giving a first guess for the required increments in controls. In the second step the incremental controls are moved in the nullspace of incremental pseudo controls in order to fulfill constraints. These constraints define desired system behavior, e.g. to turn off the main propellers in wingorne mode.

The weighted pseudo inverse solution of the incremental dynamic inversion follows

$$\Delta \boldsymbol{u}_{PI} = \boldsymbol{W}_{\boldsymbol{u}} \left(\tilde{\boldsymbol{B}} \boldsymbol{W}_{\boldsymbol{u}} \right)^{+} \Delta \boldsymbol{\nu} \tag{3.69}$$

where $+$ denotes the pseudo inverse and $\boldsymbol{W}_{\boldsymbol{u}}$ denotes a weighting matrix. The weighted pseudo inverse minimizes the 2-norm of $\boldsymbol{W}_{\boldsymbol{u}}^{-1} \Delta \boldsymbol{u}$ under the constraint of equation (3.67). After the pseudoinverse, a change of $\Delta \boldsymbol{u}$ that is in the nullsapce of $\hat{\boldsymbol{B}}$ does not affect the resulting increment in desired pseudo controls $\Delta \boldsymbol{\nu}$.

$$\Delta \boldsymbol{\nu} \approx \tilde{\boldsymbol{B}} \Delta \boldsymbol{u} = \tilde{\boldsymbol{B}} \left(\Delta \boldsymbol{u}_{PI} + \Delta \boldsymbol{u}_{NST} \right) = \tilde{\boldsymbol{B}} \Delta \boldsymbol{u}_{PI} \tag{3.70}$$

where $\Delta \boldsymbol{u}_{NST}$ denotes a control increment in the nullspace of $\tilde{\boldsymbol{B}}$. Let $\boldsymbol{N}_{\tilde{\boldsymbol{B}}} \in \mathbb{R}^{n_u \times (n_u - n_\nu)}$ denote the said nullspace of $\tilde{\boldsymbol{B}}$. The nullspace incremenets $\Delta \boldsymbol{u}_{NST}$ can drive additional constraints to

zero. Let $\hat{\boldsymbol{c}}$ denote those desired constraints. The additional null space increment fullfilling the constraints is

$$\Delta \boldsymbol{u}_{NST} = -\boldsymbol{N}_{\tilde{\boldsymbol{B}}} \left(\hat{\boldsymbol{B}}_{\boldsymbol{c}} \boldsymbol{N}_{\tilde{\boldsymbol{B}}} \right)^{+} \left(\hat{\boldsymbol{B}}_{\boldsymbol{c}} \Delta \boldsymbol{u}_{PI} + \hat{\boldsymbol{c}} \right) \tag{3.71}$$

where $\hat{\boldsymbol{B}}_{\boldsymbol{c}}$ is the estimated derivative of the constraint $\boldsymbol{c}$ with respect to the controls $\Delta \boldsymbol{u}$ and where $\hat{\boldsymbol{c}}$ denotes the current estimate of the constraint values.

Constraints and Weights

Now that we there is a general approach to handle weights and constraints in the control allocation, the question arises how to specify them in order to define desired allocation behavior. Since the desired behavior of the aircraft depends on the current mode (hovering or wingborne), an airspeed depending blending λ is introduced:

$$\lambda \left(V_A \right) = \frac{1}{1 + e^{-1.1488(V_A - 12)}} \tag{3.72}$$

This blending has a sigmoidal shape and equals one for wingborne and zero for hover flight. For the transition and the braking phase (back transition) this blending provides a smooth function to blend the weights and constraints between wingborne and hover. The front propeller is supposed to be used equally in wingborne and hover flight which makes its weight equal to one over the whole envelope. In hover, the two main propellers should be used fully and be turned off for wingborne mode, which leads to the $1 - \lambda$ terms in equation (3.73). In contrast to the main propellers the elevator shall be used only in wingborne flight but not in hover. In wingborne mode, the controller should use the virtual control θ fully. For low velocities in hover θ shall be used for braking. For this reason, the weight for the virtual control θ should not blend completely to zero but only to 0.1. This is ensured by the term $\max(\lambda, 0.1)$.

$$\begin{aligned} \boldsymbol{W}_{\boldsymbol{u}} &= \begin{bmatrix} W_{\omega_1} & 0 & 0 & 0 & 0 \\ 0 & W_{\omega_2} & 0 & 0 & 0 \\ 0 & 0 & W_{\omega_3} & 0 & 0 \\ 0 & 0 & 0 & W_{\eta} & 0 \\ 0 & 0 & 0 & 0 & W_{\theta} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1-\lambda & 0 & 0 & 0 \\ 0 & 0 & 1-\lambda & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & \max(\lambda, 0.1) \end{bmatrix} \end{aligned} \tag{3.73}$$

Since the weights only operate on the increments and not the total control values, additional constraints drive the control effectors to their desired operation point. The constraints are given in equation (3.74). The first constraint pulls the pitch angle to zero in hover flight and releasess the pitch angle for wingborne flight. The second constraint ensures that the main propellers are turned off in wingborne flight, while the elevator will be kept at zero during hover. The constraints

are normalized with respect to the maximum controls, so that the order of magnitude of the two constraints is comparable.

$$\begin{aligned}\hat{c}_1 &= (1-\lambda)\left(\frac{\hat{\theta}}{45\frac{\pi}{180}}\right)^2 \\ \hat{c}_2 &= \lambda\left[\left(\frac{\hat{\omega}_2}{\omega_{2,max}}\right)^2 + \left(\frac{\hat{\omega}_3}{\omega_{3,max}}\right)^2\right] + (1-\lambda)\left(\frac{\hat{\eta}}{\eta_{max}}\right)^2 \\ \hat{\boldsymbol{c}} &= \begin{bmatrix}\hat{c}_1 \\ \hat{c}_2\end{bmatrix}\end{aligned} \tag{3.74}$$

3.3.4 Onboard Plant Estimation

The onboard plant estimation estimates all required physical model quantities for the INDI controllers. The required quantities are states, controls, pseudo controls, constraints and derivative information required for the incremental inversion. Figure 3.4 shows the main strucuture of the onboard plant estimation. The first step is to estimate the current controls which is done by forwarding the control commands through dynamic onboard actuator models. In the second step, which is the main part of this thesis, a model filter determines model based feedworward pseudocontrols $\hat{\boldsymbol{\gamma}}_{OBPM}$, estimated states $\hat{\boldsymbol{x}}$, constraints $\hat{\boldsymbol{c}}$, cost functions $\hat{\boldsymbol{J}}$ and the incremental inversion matrix $\tilde{\boldsymbol{B}}$ by using measurements of the vehicle states $\boldsymbol{x}_meas$ and estimated controls $\hat{\boldsymbol{u}}$. In the final step a complementary filter combines measured pseudo controls $\boldsymbol{\gamma}_{meas}$ with aforementioned model based feedforward pseudo controls in order to obtain an estimate for the pseudo controls.

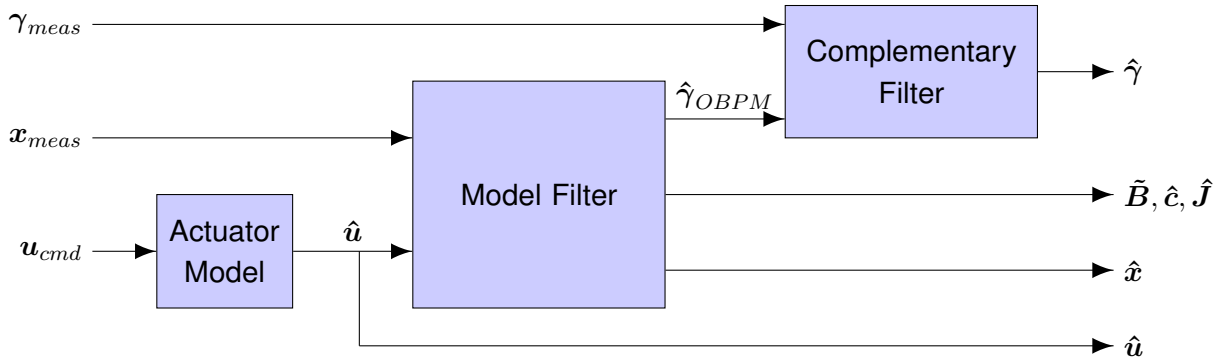

Figure 3.4: Onboard plant estimation structure

Complementary Filter

The complementary filter uses two sources of information for the pseudo control estimation. The low frequency part of the pseudo control estimation is obtained from the pseudo control measurements (e.g. accelerometers). Using the measurements also for the high frequent parts is unfeasible because of high frequency noise in the measurement signals. In order to keep the

bandwidth of the estimated pseudocontrols high, the complementary filter uses model based pseudo control information in order to estimate the high frequency parts of the pseudo control. This complementary fusion of signals is done by PT1 filters. As the next equation shows the two PT1 filters used for measurements and model based signals are complementary, which means their sum equals one for all frequencies.

$$\hat{\boldsymbol{\gamma}} = \frac{\boldsymbol{I}s}{\boldsymbol{I}s + \boldsymbol{\Omega}_{CF}}\hat{\boldsymbol{\gamma}} + \frac{\boldsymbol{\Omega}_{CF}}{\boldsymbol{I}s + \boldsymbol{\Omega}_{CF}}\hat{\boldsymbol{\gamma}} = \frac{\boldsymbol{I}s}{\boldsymbol{I}s + \boldsymbol{\Omega}_{CF}}\hat{\boldsymbol{\gamma}}_{OBPM} + \frac{\boldsymbol{\Omega}_{CF}}{\boldsymbol{I}s + \boldsymbol{\Omega}_{CF}}\hat{\boldsymbol{\gamma}}_{meas} \tag{3.75}$$

where $\boldsymbol{\Omega}_{CF}$ denotes the complementary filter frequencies given by

$$\boldsymbol{\Omega}_{CF} = \begin{bmatrix} \omega_{CF,\dot{u}} & 0 & 0 \\ 0 & \omega_{CF,\dot{w}} & 0 \\ 0 & 0 & \omega_{CF,\dot{q}} \end{bmatrix} \tag{3.76}$$

Note that in equation (3.75) there are no real measurements for the pitch acceleration pseudo control. While for the translational acceleration channels the estimated pseudo control consists of a highpass model feedworward part and a lowpassed accelerometer part, the pitch channel also requires a highpass on the pitch rate measurement:

$$\dot{\hat{q}} = \frac{s}{s + \omega_{CF,\dot{q}}}\left(\dot{\hat{q}}_{OBPM} + \omega_{CF,\dot{q}} q_{meas}\right) \tag{3.77}$$

Model Fitler

Figure 3.5 shows the structure of the model filter in figure 3.4. It consists of a Kalman filter and the onboard plant model (OBPM), which will be the focus of the next subsection. The Kalman filter performs a data fusion of measurement information and infromation of the onboard model. The measurement information of the Kalamn filter are the kinematic velocities, the pitch angle and the pitch rate. An inertial navigation system (INS) can provide all these quantities. The output of the Kalman filter are estimated states $\hat{\boldsymbol{x}}$ and parameters $\hat{\boldsymbol{p}}$. Since an extended Kalman filter will be used, the OBPM is evaluated at the estimated states and parameters. The required derivative information and the time derivative of the states is then fed back to the Kalman filter.

The (extended) Kalman filter in figure 3.5 is the main focus of this thesis. In order to investigate the effect of the Kalman filter on the closed loop performance, a baseline controller without the Kalman filter is used for comparison. For the baseline case, the estimated values of the states are simply the measured states.

$$\hat{\boldsymbol{x}} = \boldsymbol{x}_{meas} \tag{3.78}$$

Without a Kalman filter there is no parameter estimation. That means the estimated parameters equal a constant nominal value.

$$\hat{\boldsymbol{p}} = \boldsymbol{p}_{nominal} \tag{3.79}$$

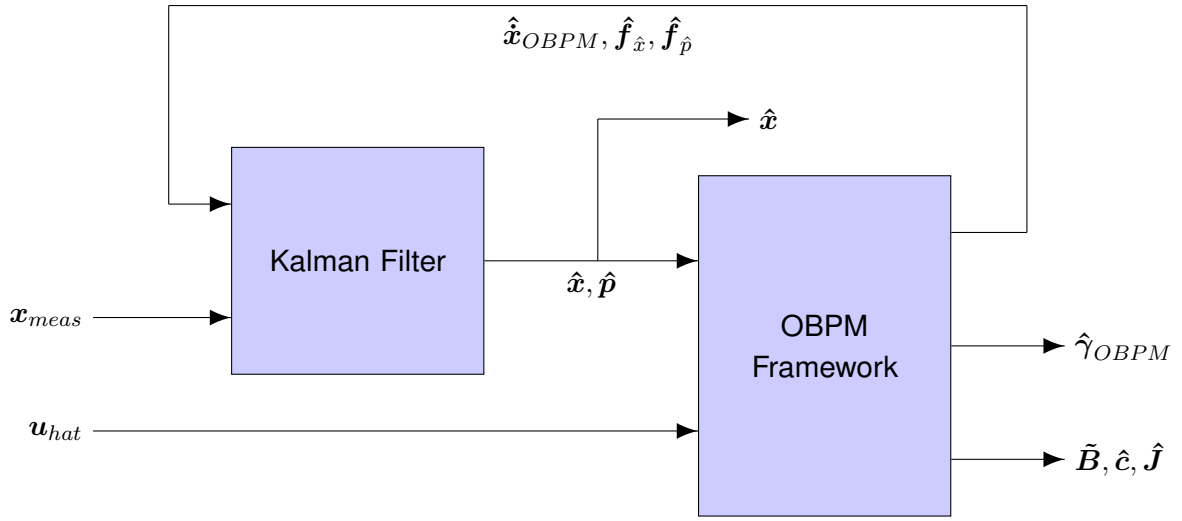

Figure 3.5: Model filter structure

Onboard Plant Model

The onboard plant model (OBPM) implements the flight dynamics model given in section 3.2.1. For the nominal case, the OBPM equals the flight dynamics of the real plant. The inputs of the OBPM are the estimated states $\hat{x}$ and controls $\hat{u}$. The output of the OBPM is the state derivative information $\hat{\dot{x}}_{OBPM}$ and consequently also the OBPM estimated pseudo controls $\hat{\gamma}_{OBPM}$. Another output of the OBPM are the constraints $\hat{c}$.

States				
	u	w	θ	q
Unit	m/s	m/s	deg	deg/s
Δ_{x_i}	0.1	0.1	1	1
Controls				
	ω_1	ω_2	ω_3	η
Unit	rad/s	rad/s	rad/s	deg
Δ_{u_i}	5	5	5	1

Table 3.6: OBPM Perturbation Data

Since the incremental dynamic inversion and the extended Kalman filter require derivatives of the OBPM outputs with respect to the OBPM inputs, the OBPM is embedded in a framework that is able to calculate these derivatives. The OBPM framework determines the gradients by finite difference approximations. For example, the derivative of the first constraint with respect to the first control ω_1 follow

$$\frac{\delta c_1}{\delta \omega_1} \approx \frac{c_1(\omega_1 + \Delta_{\omega_1}) - c_1(\omega_1)}{\Delta_{\omega_1}} \tag{3.80}$$

where Δ_{ω_1} denotes the perturbation in the first control. The numeric values for the OBPM perturbations are given in table 3.6.

3.3.5 Numeric Controller Data

Reference Models			
	u	w	θ
a_0	3	3	36
a_1			12
Error Controller			
	u	w	θ
a_0	3	3	36
a_1			12
a_{-1}			36

Table 3.7: Numeric Reference Model and Error Controller Data

Estimated Propeller Dynamics						
	$w_{0,u}$	ζ_u	u_{min}	u_{max}	$\dot{u}_{min}$	$\dot{u}_{max}$
Unit	rad/s	-	rad/s	rad/s	rad/s^2	rad/s^2
P_1	30	1	1	1000	-500	500
P_2	30	1	1	500	-500	500
P_3	30	1	1	500	-500	500
Estimated Elevator Dynamics						
	$w_{0,u}$	ζ_u	u_{min}	u_{max}	$\dot{u}_{min}$	$\dot{u}_{max}$
Unit	rad/s	-	deg	deg	deg/s	deg/s
η	30	1	-45	45	-60	60

Table 3.8: Numeric Control Effector Estimation Dynamics Data

Complementary Filter				
	unit	u-channel	w-channel	θ-channel
ω_{CF}	rad/s	20	20	20

Table 3.9: Numeric Complementary Filter Data

Chapter 4

Requirements Capture

This chapter performs a requirements analysis for the model filter. The main focus will be overall closed loop performance requirements. The analysis will be sepereated in a linear performance requirements and a nonlinear performance requirements part.

4.1 Linear Closed Loop Performance Measures

This section focusses on linear closed loop performance measures. Therefore the closed loop plant is linearized and linear robustness analysis is carried out.

4.1.1 Linearization of the Closed Loop

Suppose the closed loop is operating in a stationary trim point (e.g. hover, wingborne flight) with $\dot{\boldsymbol{x}} = \mathbf{0}$. The linear dynamics of the plant around the operating point then is

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{4.1}$$

Using the same linear dynamics for the state estimation in a continoous Kalman filter yields

$$\dot{\hat{\boldsymbol{x}}} = \hat{\boldsymbol{A}}\hat{\boldsymbol{x}} + \hat{\boldsymbol{B}}\hat{\boldsymbol{u}} + \boldsymbol{K}\left(\boldsymbol{x} - \hat{\boldsymbol{x}}\right) \tag{4.2}$$

with the stationary Kalman gain $\boldsymbol{K}$. Now lets have a look at the commanded controls. The extension of the controls with the pitch angle leads to the INDI inversion law

$$\begin{bmatrix}\Delta\boldsymbol{u} \\ \Delta\mathbf{v}\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{C}_\gamma\hat{\boldsymbol{B}} & \boldsymbol{C}_\gamma\hat{\boldsymbol{A}}_\mathrm{v}\end{bmatrix}^+}_{\tilde{\boldsymbol{B}}_\gamma^+}\Delta\boldsymbol{\gamma} \tag{4.3}$$

where $\hat{\boldsymbol{A}}_\mathbf{v}$ denotes the columns of $\hat{\boldsymbol{A}}$ corresponding to the virtual control states. The $[...]^+$ denotes a pseudo inverse. According to the linearization in this subsection the pseudo inverse is again assumed to be a constant gain. For the following derivations we need real controls and virtual controls seperately:

$$\begin{aligned}\Delta\boldsymbol{u} &= \boldsymbol{S}_u\tilde{\boldsymbol{B}}_\gamma^+\Delta\boldsymbol{\gamma} \\ \Delta\mathbf{v} &= \boldsymbol{S}_\mathrm{v}\tilde{\boldsymbol{B}}_\gamma^+\Delta\boldsymbol{\gamma}\end{aligned} \tag{4.4}$$

The matrices $\boldsymbol{S}_u$ and $\boldsymbol{S}_{\mathrm{v}}$ are selection matrices given by

$$\begin{aligned} \boldsymbol{S}_u &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\ \boldsymbol{S}_{\mathrm{v}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.5}$$

The desired change in pseudo control is

$$\begin{aligned} \Delta\boldsymbol{\gamma} &= \boldsymbol{\nu} - \hat{\boldsymbol{\gamma}} \\ &= \boldsymbol{\nu}_{ref} + \boldsymbol{\nu}_{ec} - \hat{\boldsymbol{\gamma}} \\ &= \boldsymbol{C}_\gamma \boldsymbol{A}_{ref} \left(\boldsymbol{x}_{ref} - \boldsymbol{x}_{cmd}\right) + \boldsymbol{C}_\gamma \boldsymbol{A}_{ec,P} \left(\hat{\boldsymbol{x}} - \boldsymbol{x}_{ref}\right) + \boldsymbol{\nu}_{ec,I} - \hat{\boldsymbol{\gamma}} \end{aligned} \tag{4.6}$$

The integral part of the error controller $\boldsymbol{\nu}_{ec,I}$ satisfies

$$\dot{\boldsymbol{\nu}}_{ec,I} = \boldsymbol{C}_\gamma \boldsymbol{A}_{ec,I} \left(\hat{\boldsymbol{x}} - \boldsymbol{x}_{ref}\right) \tag{4.7}$$

Note that the commanded states result from the external commands $\boldsymbol{y}_{cmd}$ and internal virtual control commands.

$$\boldsymbol{x}_{cmd} = \boldsymbol{C}_y^T \boldsymbol{y}_{cmd} + \boldsymbol{C}_{\mathrm{v}}^T \mathbf{v}_{cmd} \tag{4.8}$$

The external comanded input is just the velocities which corresponds to the first two states of the system. The fed back virtual control is the ptich angle, which corresponds to the third state command. The matrices $\boldsymbol{C}_y^T$ and $\boldsymbol{C}_{\mathrm{v}}^T$ consequently follow:

$$\begin{aligned} \boldsymbol{C}_y^T &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \boldsymbol{C}_{\mathrm{v}}^T &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \end{aligned} \tag{4.9}$$

The commanded virtual control is the current estimate of the virtual control plus the desired change in the virtual control. Since the virtual control is a plant state the estimate of the virtual control is simply the estimate of the corresponding plant state. The commanded virtual control then follows:

$$\mathbf{v}_{cmd} = \hat{\mathbf{v}} + \Delta\mathbf{v} = \boldsymbol{C}_{\mathrm{v}} \hat{\boldsymbol{x}} + \Delta\mathbf{v} \tag{4.10}$$

Inserting this into equation (4.8) leads to

$$\boldsymbol{x}_{cmd} = \boldsymbol{C}_y^T \boldsymbol{y}_{cmd} + \boldsymbol{C}_{\mathrm{v}}^T \boldsymbol{C}_{\mathrm{v}} \hat{\boldsymbol{x}} + \boldsymbol{C}_{\mathrm{v}}^T \Delta\mathbf{v} \tag{4.11}$$

Combining the last equation and equation (4.6) yields

$$\begin{aligned}\Delta\boldsymbol{\gamma} = &\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ref}-\boldsymbol{A}_{ec,P}\right)\boldsymbol{x}_{ref}+\boldsymbol{C}_\gamma\boldsymbol{C}_\gamma\boldsymbol{A}_{ec,P}\hat{\boldsymbol{x}}+\boldsymbol{\nu}_{ec,I}-\hat{\boldsymbol{\gamma}}\\ &-\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{C}_{\mathrm{v}}\hat{\boldsymbol{x}}-\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\boldsymbol{y}_{cmd}\\ &-\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\Delta\mathbf{v}\end{aligned} \tag{4.12}$$

Above equation illustrates the algebraic virtual control loop. The required change in virtual controls is contained in the left and right hand side of equation. This is easier to see when taking into account equation (4.3) and moving all virtual control parts to the left hand side:

$$\begin{aligned}\underbrace{\left(\tilde{\boldsymbol{B}}_\gamma+\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{S}_{\mathrm{v}}\right)}_{\boldsymbol{T}}\begin{bmatrix}\Delta\boldsymbol{u}\\ \Delta\mathbf{v}\end{bmatrix} = &\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ref}-\boldsymbol{A}_{ec,P}\right)\boldsymbol{x}_{ref}\\ &+\left(\boldsymbol{C}_\gamma\boldsymbol{A}_{ec,P}-\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{C}_{\mathrm{v}}\right)\hat{\boldsymbol{x}}\\ &+\boldsymbol{\nu}_{ec,I}-\hat{\boldsymbol{\gamma}}\\ &-\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\boldsymbol{y}_{cmd}\end{aligned} \tag{4.13}$$

With the help of the new defined matrix $\boldsymbol{T}$ we can solve for the desired change in pseudo control:

$$\begin{aligned}\Delta\boldsymbol{\gamma} = &\tilde{\boldsymbol{B}}_\gamma\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ref}-\boldsymbol{A}_{ec,P}\right)\boldsymbol{x}_{ref}\\ &+\tilde{\boldsymbol{B}}_\gamma\boldsymbol{T}^{-1}\left(\boldsymbol{C}_\gamma\boldsymbol{A}_{ec,P}-\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{C}_{\mathrm{v}}\right)\hat{\boldsymbol{x}}\\ &+\tilde{\boldsymbol{B}}_\gamma\boldsymbol{T}^{-1}\boldsymbol{\nu}_{ec,I}\\ &-\tilde{\boldsymbol{B}}_\gamma\boldsymbol{T}^{-1}\hat{\boldsymbol{\gamma}}\\ &-\tilde{\boldsymbol{B}}_\gamma\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\boldsymbol{y}_{cmd}\end{aligned} \tag{4.14}$$

Now that we have the desired change in pseudocontrol, we can use the result to write the differential equation for the reference model

$$\begin{aligned}\dot{\boldsymbol{x}}_{ref} &= \boldsymbol{A}_{ref}\left(\boldsymbol{x}_{ref}-\boldsymbol{x}_{cmd}\right)-\boldsymbol{C}_\gamma^T\Delta\boldsymbol{\gamma}\\ &= \boldsymbol{A}_{ref}\boldsymbol{x}_{ref}-\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\boldsymbol{y}_{cmd}-\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{C}_{\mathrm{v}}\hat{\boldsymbol{x}}-\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\Delta\mathbf{v}-\boldsymbol{C}_\gamma^T\Delta\boldsymbol{\gamma}\end{aligned} \tag{4.15}$$

Inserting the results from equations (4.3) and (4.14)

$$\begin{aligned}\dot{\boldsymbol{x}}_{ref} = &\left[\boldsymbol{A}_{ref}-\left(\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{S}_{\mathrm{v}}+\boldsymbol{C}_\gamma^T\tilde{\boldsymbol{B}}_\gamma\right)\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ref}-\boldsymbol{A}_{ec,P}\right)\right]\boldsymbol{x}_{ref}\\ &-\left[\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{C}_{\mathrm{v}}+\left(\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{S}_{\mathrm{v}}+\boldsymbol{C}_\gamma^T\tilde{\boldsymbol{B}}_\gamma\right)\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ec,P}-\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{C}_{\mathrm{v}}\right)\right]\hat{\boldsymbol{x}}\\ &-\left[\left(\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{S}_{\mathrm{v}}+\boldsymbol{C}_\gamma^T\tilde{\boldsymbol{B}}_\gamma\right)\boldsymbol{T}^{-1}\right]\boldsymbol{\nu}_{ec,I}\\ &+\left[\left(\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{S}_{\mathrm{v}}+\boldsymbol{C}_\gamma^T\tilde{\boldsymbol{B}}_\gamma\right)\boldsymbol{T}^{-1}\right]\hat{\boldsymbol{\gamma}}\\ &-\left[\boldsymbol{A}_{ref}\boldsymbol{C}_y^T-\left(\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{S}_{\mathrm{v}}+\boldsymbol{C}_\gamma^T\tilde{\boldsymbol{B}}_\gamma\right)\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\right]\boldsymbol{y}_{cmd}\end{aligned} \tag{4.16}$$

So far we have the differential equations for the real plant, the estimated plant, the reference model and the integral error controller part. We still need the equations for real and estimated controll effectors and for the pseudo control complementary filter. We can assemble the second

order control effector dynamics in two matrices:

$$\begin{aligned}\Omega_{act} &= \begin{bmatrix} \omega_{0,P1}^2 & 0 & 0 & 0 \\ 0 & \omega_{0,P2}^2 & 0 & 0 \\ 0 & 0 & \omega_{0,P3}^2 & 0 \\ 0 & 0 & 0 & \omega_{0,\eta}^2 \end{bmatrix} \\ D_{act} &= \begin{bmatrix} 2\omega_{0,P1}\zeta_{P1} & 0 & 0 & 0 \\ 0 & 2\omega_{0,P2}\zeta_{P2} & 0 & 0 \\ 0 & 0 & 2\omega_{0,P3}\zeta_{P3} & 0 \\ 0 & 0 & 0 & 2\omega_{0,\eta}\zeta_{\eta} \end{bmatrix}\end{aligned} \tag{4.17}$$

Which leads to the control effector dynamics equation

$$\begin{aligned}\ddot{\boldsymbol{u}} &= -\boldsymbol{\Omega}_{act}\boldsymbol{u} - \boldsymbol{D}_{act}\dot{\boldsymbol{u}} + \boldsymbol{\Omega}_{act}\boldsymbol{u}_{cmd} \\ &= -\boldsymbol{\Omega}_{act}\boldsymbol{u} - \boldsymbol{D}_{act}\dot{\boldsymbol{u}} + \boldsymbol{\Omega}_{act}\hat{\boldsymbol{u}} + \boldsymbol{\Omega}_{act}\boldsymbol{S}_u\tilde{\boldsymbol{B}}_\gamma^+\Delta\boldsymbol{\gamma}\end{aligned} \tag{4.18}$$

Here we can insert the result for the desired change in pseudo control again and end up with

$$\begin{aligned}\ddot{\boldsymbol{u}} = &-\boldsymbol{\Omega}_{act}\boldsymbol{u} - \boldsymbol{D}_{act}\dot{\boldsymbol{u}} + \boldsymbol{\Omega}_{act}\hat{\boldsymbol{u}} \\ &+ \left[\boldsymbol{\Omega}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ref} - \boldsymbol{A}_{ec,P}\right)\right]\boldsymbol{x}_{ref} \\ &+ \left[\boldsymbol{\Omega}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ec,P} - \boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{C}_{\mathrm{v}}\right)\right]\hat{\boldsymbol{x}} \\ &+ \left[\boldsymbol{\Omega}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\right]\boldsymbol{\nu}_{ec,I} \\ &- \left[\boldsymbol{\Omega}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\right]\hat{\boldsymbol{\gamma}} \\ &- \left[\boldsymbol{\Omega}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\right]\boldsymbol{y}_{cmd}\end{aligned} \tag{4.19}$$

The estimated actuator model follows accordingly:

$$\begin{aligned}\ddot{\hat{\boldsymbol{u}}} = &-\hat{\boldsymbol{D}}_{act}\dot{\hat{\boldsymbol{u}}} \\ &+ \left[\hat{\boldsymbol{\Omega}}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ref} - \boldsymbol{A}_{ec,P}\right)\right]\boldsymbol{x}_{ref} \\ &+ \left[\hat{\boldsymbol{\Omega}}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\left(\boldsymbol{A}_{ec,P} - \boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{C}_{\mathrm{v}}\right)\right]\hat{\boldsymbol{x}} \\ &+ \left[\hat{\boldsymbol{\Omega}}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\right]\boldsymbol{\nu}_{ec,I} \\ &- \left[\hat{\boldsymbol{\Omega}}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\right]\hat{\boldsymbol{\gamma}} \\ &- \left[\hat{\boldsymbol{\Omega}}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\right]\boldsymbol{y}_{cmd}\end{aligned} \tag{4.20}$$

The complementary filter follows the derivations in the last chapter:

$$\begin{aligned}\dot{\hat{\boldsymbol{\gamma}}} &= -\boldsymbol{\Omega}_{cf}\hat{\boldsymbol{\gamma}} + \dot{\hat{\boldsymbol{\gamma}}}_{obpm} + \boldsymbol{\Omega}_{cf}\boldsymbol{\gamma} \\ &= -\boldsymbol{\Omega}_{cf}\hat{\boldsymbol{\gamma}} + \boldsymbol{C}_\gamma\hat{\boldsymbol{A}}\dot{\hat{\boldsymbol{x}}} + \boldsymbol{C}_\gamma\hat{\boldsymbol{B}}\dot{\hat{\boldsymbol{u}}} + \boldsymbol{\Omega}_{cf}\boldsymbol{C}_\gamma\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\Omega}_{cf}\boldsymbol{C}_\gamma\boldsymbol{B}\boldsymbol{u} \\ &= -\boldsymbol{\Omega}_{cf}\hat{\boldsymbol{\gamma}} + \boldsymbol{C}_\gamma\hat{\boldsymbol{A}}\left(\hat{\boldsymbol{A}} - \boldsymbol{K}\right)\hat{\boldsymbol{x}} + \boldsymbol{C}_\gamma\hat{\boldsymbol{A}}\hat{\boldsymbol{B}}\hat{\boldsymbol{u}} \\ &\quad + \left(\boldsymbol{C}_\gamma\hat{\boldsymbol{A}}\boldsymbol{K} + \boldsymbol{\Omega}_{cf}\boldsymbol{C}_\gamma\boldsymbol{A}\right)\boldsymbol{x} + \boldsymbol{\Omega}_{cf}\boldsymbol{C}_\gamma\boldsymbol{B}\boldsymbol{u} + \boldsymbol{C}_\gamma\hat{\boldsymbol{B}}\dot{\hat{\boldsymbol{u}}}\end{aligned} \tag{4.21}$$

Now we can assemble the total closed loop system. The assembled closed loop state is

$$\boldsymbol{x}_{cl} = \begin{bmatrix} \boldsymbol{x} \\ \hat{\boldsymbol{x}} \\ \boldsymbol{x}_{ref} \\ \boldsymbol{\nu}_{ec,I} \\ \hat{\boldsymbol{\gamma}} \\ \boldsymbol{u} \\ \hat{\boldsymbol{u}} \\ \dot{\boldsymbol{u}} \\ \hat{\dot{\boldsymbol{u}}} \end{bmatrix} \tag{4.22}$$

The linear state space representation of the full closed loop linear model then is

$$\begin{aligned} \dot{\boldsymbol{x}}_{cl} &= \boldsymbol{A}_{cl}\boldsymbol{x}_{cl} + \boldsymbol{B}_{cl}\boldsymbol{y}_{cmd} \\ \boldsymbol{y} &= \boldsymbol{C}_{cl}\boldsymbol{x}_{cl} \end{aligned} \tag{4.23}$$

with the system input and output matrices $\boldsymbol{B}_{cl}$ and $\boldsymbol{C}_{cl}$. For the sake of space consumption the closed loop dynamics matrix will not be represented in full shape. The matrices $\boldsymbol{B}_{cl}$ and $\boldsymbol{C}_{cl}$ are given in the appendix.

4.1.2 Trimming of the Closed Loop

For the sake of simplicity the closed loop is not trimmed analytically. For the trimming the nominal nonlinear baseline simulation from section 4.2.1 is used. Since the onboard plant model for the nominal baseline simulation coincides with the real plant (only the sampletime is different), the finite difference gradients of the OBPM framework can be used to determine the corresponding matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ for each time step. The linearization then holds for the deviations from the baseline trajectory and the linear analysis can be carried out.

4.1.3 Uncertainty Modelling for the Linear Closed Loop

This subsection introduces the uncertainty modelling for the linear closed loop. The general model of dynamic uncertainties follows the so called $\boldsymbol{\Delta}\boldsymbol{M}(s)$ analysis model. Later specific uncertainty sources (input channel, output channel, plant parametric uncertainties) are taken into account.

$\boldsymbol{\Delta}\boldsymbol{M}(s)$ Analysis Model

The $\boldsymbol{\Delta}\boldsymbol{M}(s)$ analysis model is a general approach to investigate the effect of uncertainties on the behavior of a linear dynamic system [**?** , p.109-110]. The idea is to split up the dynamics of the system in a nominal dynamics part $\boldsymbol{M}(s)$ and the uncertainties $\boldsymbol{\Delta}(s)$. Figure 4.1 depicts this structure. If the uncertainty $\boldsymbol{\Delta}$ equals zero, the uncertain system will equal the nominal system $\boldsymbol{M}(s)$. If the uncertainty $\boldsymbol{\Delta}$ does not equal zero it will introduce additional closed loop dynamics. These additional closed loop dynamics can drive the system unstable. The maximum uncertainty $\boldsymbol{\Delta}$ under which the system remains stable is a measure for robustness of the system.

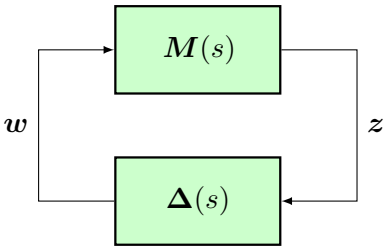

Figure 4.1: Uncertainty Modeling with the ΔM analysis model

Obtaining $M(s)$ and Δ

The linear dynamic system $M(s)$ and the matrix Δ required for the analysis model can be obtained by the following approach [**?**]. Assume the actual closed loop matrix A_{cl} equals the sum of the nominal closed loop matrix and a linear combination of different error matrices.

$$A_{cl} = A_{cl,nom} + \sum_{i=1}^{n} E_i \delta_i \tag{4.24}$$

Now perform a singular value composition of E_i

$$E_i = U \Sigma V^* \tag{4.25}$$

Since most uncertainties only affect certain parts of the closed loop matrix E_i will not have full rank. Let k_i denote the rank of E_i. We now can discard the parts of the U, Σ and V^* matrices that correspond to singular values that are zero. The 'reduced' singular value composition then is

$$E_i = U_{red} \Sigma_{red} V^*_{red} \tag{4.26}$$

$$\begin{aligned} \beta_i &= U_{red} \Sigma_{red}^{\frac{1}{2}} \\ \alpha_i &= \Sigma_{red}^{\frac{1}{2}} V^*_{red} \end{aligned} \tag{4.27}$$

Now we can write the closed loop dynamics matrix as

$$A_{cl} = A_{cl,nom} + \sum_{i=1}^{n} \beta_i \delta_i \alpha_i \tag{4.28}$$

The idea of this decomposition is to write the sum on the right side of equation (4.28) as feedback for the nominal stable system. The nominal stable system will have a state space representation with the matrices A_{nom}, B_{nom} and C_{nom}. This nominal stable system then corresponds to the $M(s)$ in figure 4.1. The feedback gain connecting output and input of the nominal system then is the corresponding Δ.

$$\begin{aligned} \dot{x}_{cl} &= A_{cl,nom} x_{cl} + B_{nom} u \\ u &= \Delta y \\ y &= C_{nom} x_{cl} \end{aligned} \tag{4.29}$$

The corresponding input and output matrices follow

$$\begin{aligned} \boldsymbol{B}_{nom} &= \begin{bmatrix} \boldsymbol{\beta}_1 & \dots & \boldsymbol{\beta}_n \end{bmatrix} \\ \boldsymbol{C}_{nom} &= \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_n \end{bmatrix} \end{aligned} \tag{4.30}$$

$$\boldsymbol{\Delta} = \begin{bmatrix} \delta_1 \boldsymbol{I}^{k_1 \times k_1} & & \\ & \ddots & \\ & & \delta_n \boldsymbol{I}^{k_n \times k_n} \end{bmatrix} \tag{4.31}$$

with k_i being the rank of $\boldsymbol{E}_i$ The nominal dynamics follows the nominal state space representation

$$\boldsymbol{M} = \boldsymbol{C}_{nom} \left(s\boldsymbol{I} - \boldsymbol{A}_{nom} \right)^{-1} \boldsymbol{B}_{nom} \tag{4.32}$$

Plant Input Uncertainties

The uncertainty model at the input breakpoint will be applied between the actuator models and the plant dynamics model. With this breakpoint the perturbed inputs will not contribute to entries in the actuator models, only to entries in the plant. The corresponding rows in the closed loop dynamcis matrix $\boldsymbol{A}_{cl}$ are the ones concerning the real plant state time derivatives $\dot{\boldsymbol{x}}$ and the pseudo control measurement part in the complementary filter for the pseudo controls. The modelled deviated inputs follow

$$u_{dev,i} = \left(1 + \delta_{input,i}\right) u_{AM,i} \tag{4.33}$$

with $u_{dev,i}$ being the ith deviated control applied to the real plant and pseudo control complementary filter, the nominal control of the actuator model $u_{AM,i}$ and the deviation $\delta_{in,i}$. The deviated closed loop matrix due to input uncertainty then follows

$$\boldsymbol{A}_{cl,in} = \boldsymbol{A}_{cl,nom} + \sum_{i=1}^{4} \boldsymbol{E}_{in,i} \delta_{in,i} \tag{4.34}$$

Plant Output Uncertainties

The plant output uncertainties are modelled accordingly to the input uncertainties as decoupled multiplictive errors in the output channels. The output channels breakpoint is between the real plant states and the measured states $\boldsymbol{x}_{meas}$ as well as the real pseudo controls and measured pseudo controls.

$$o_{meas,i} = \left(1 + \delta_{out,i}\right) o_{real,i} \tag{4.35}$$

where $o_{meas,i}$ denotes the ith output measurement enterring the controller and $o_{real,i}$ denotes the real output. The seven uncertain output channels with independent error matrices $\boldsymbol{E}_{out,i}$ lead to the total closed loop matrix $\boldsymbol{A}_{cl}$

$$\boldsymbol{A}_{cl,out} = \boldsymbol{A}_{cl,nom} + \sum_{i=1}^{7} \boldsymbol{E}_{out,i} \delta_{out,i} \tag{4.36}$$

Plant Dynamics Uncertainties

The open loop performance of the Kalman filter depends on the estimated model dynamics. High uncertainties in the model dynamics correspond to a higher process noise leading to higher Kalman gains. For the closed loop these uncertainties are contained in the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ as real valued perturbations. Note that it is not feasible to model the uncertainty in the estimated quantities $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$, since they have a nonlinear influence on the closed loop matrix $\boldsymbol{A}_{cl}$. Allocating the uncertainties in $\boldsymbol{A}$ and $\boldsymbol{B}$ allows us to perform the same linear robustness analysis as for the input and output channels.

Modelling the input and output uncertainties was quite intuitive since there are four/seven seperate channels in both cases. The assumption of independet actuator channels and indepedet sensor channels leaded to one complex scalar uncertainty for each channel.

For the uncertainty modelling of the plant we need to model multiplications with real valued uncertain matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. One way to tackle this issue is to split up the matrices into their 16 entries each and to perform a robustness analysis on all these channels. This approach leads to 32 uncertain scalars. Taking the physical understanding of the plant into account can help to reduce this number. The strapdown equation (third row of $\boldsymbol{A}$) never contains any uncertainties which cancels four entries. Since all control effectors only have direct influence on forces and moments the third row in B is always zero and also not uncertain, which also reduces the number of entries by four. The next assumption is that the main influence of uncertainty in the propellers is the effectiveness of the propellers and not the angle or position of the resulting force. This leads to the same multiplicative uncertainty for the $\dot{u}$,$\dot{w}$ and $\dot{q}$ channels for each propeller. Since the influence of the pitch rate on aerodynamic forces and moments was neglected throghout the modelling, the uncertainties in those effects will also be neglected here (fourth column of $\boldsymbol{A}$). Following these simplifications results in an $\boldsymbol{A}$ matrix with 9 and a $\boldsymbol{B}$ matrix with 6 uncertain parameters. The deviated matrices then follow

$$\begin{aligned}
\boldsymbol{A} &= \boldsymbol{A}_{nom} \odot \begin{bmatrix} 1+\delta_1 & 1+\delta_2 & 1+\delta_3 & 1 \\ 1+\delta_4 & 1+\delta_5 & 1+\delta_6 & 1 \\ 1 & 1 & 1 & 1 \\ 1+\delta_7 & 1+\delta_8 & 1+\delta_9 & 1 \end{bmatrix} \\
\boldsymbol{B} &= \boldsymbol{B}_{nom} \odot \begin{bmatrix} 1+\delta_{10} & 1+\delta_{11} & 1+\delta_{12} & 1+\delta_{13} \\ 1+\delta_{10} & 1+\delta_{11} & 1+\delta_{12} & 1+\delta_{14} \\ 1 & 1 & 1 & 1 \\ 1+\delta_{10} & 1+\delta_{11} & 1+\delta_{12} & 1+\delta_{15} \end{bmatrix}
\end{aligned} \tag{4.37}$$

where $\odot$ denotes the elementwise or hadamarad product and $\boldsymbol{A}_{nom}$, $\boldsymbol{B}_{nom}$ the nominal matrices. The corresponding error matrices $E_1 ... E_{15}$ are given by inserting the deviated matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in the closed loop matrix $\boldsymbol{A}_{cl}$ and extracting the terms with $\delta_1 ... \delta_{15}$.

4.1.4 Robustness Analysis

Now that we have three $\boldsymbol{\Delta M}$ analysis models for the input, output and plant dynamics respectively, how to perform robustness analysis? The overall transfer function of the system

follows

$$\boldsymbol{G}(s) = \frac{\boldsymbol{M}(s)}{\boldsymbol{I} + \Delta \boldsymbol{M}(s)} \tag{4.38}$$

The magnitude of $\boldsymbol{\Delta}$ that the system can cope with without becoming unstable is a measure for robustness of the system. For the easiest case, where $\boldsymbol{\Delta}$ is a full complex valued matrix, e.g.

$$\boldsymbol{\Delta} = \begin{bmatrix} \delta_1 & \delta_2 \\ \delta_3 & \delta_4 \end{bmatrix} \tag{4.39}$$

the robustness of the system can be measured by the small gain theorem [**?**]. The disadvatage of this approach is, that the robustness bounds for the different δ_i can be very conservative if the matrix $\boldsymbol{\Delta}$ has a sparse structure. In this considered application the attained $\boldsymbol{\Delta}$s surely have a sparse structure. As equation (4.31) shows, $\boldsymbol{\Delta}$ has a diagonal structure since all the cross coupling elements between different δ_i equal zero. A less conservative bound for this sparse structure is the so called structured singular value. The structured singular value exploits the sparse structure of $\boldsymbol{\Delta}$ and gives less conservative bounds for the entries δ_i. For this thesis theMATLAB® implimentation for the structured singular value will be used [**?**] to asses robustness of the closed loop.

4.2 Nonlinear Performance Measures

This section focusses on the performance criteria analysis of the nonlinear closed loop. Therefore the baseline for the nonlinear simulation is presented and a scalar closed loop performance measure is introduced.

4.2.1 Simulation of the Nonlinear Closed Loop

The closed loop model is implemented in SIMULINK. The continuous plant is integrated with a sampletime of $0.001s$ and the Runge Kutta integration scheme ode4. The controller has a discrete sampletime of $0.005s$.

In order to assess and compare the performance of different controllers, a baseline command will be applied to the closed loop. The baseline command begins with ahover phase with small velocity commands. Then it transitions to full wingborne flight with $20m/s$ and some vertical velocity commands. In the end it breaks back to hover mode. The total duration of the flown trajectory is $100s$. Figure 4.2 shows the command tracking in velocities The baseline simulation is performed with the baseline controller without the Kalman filter. Wind, Parametric uncertainties, actuator uncertainties, sensor bias errors and sensor time delays are neglected. White sensor noise is considered.

The baseline controller achieves similar command tracking as in [**?**]. This could be expected since the baseline only contained three changes compared to [**?**]. The first change, the cutoff of the aerodynamic lift coefficient mainly affects the controller performance for high aerodynamic angles of attack. Since there is no wind the aerodynamic velocities for the high angle of attack situations (e.g. backward hover flight, hover descent and ascent) are quite low, which makes the

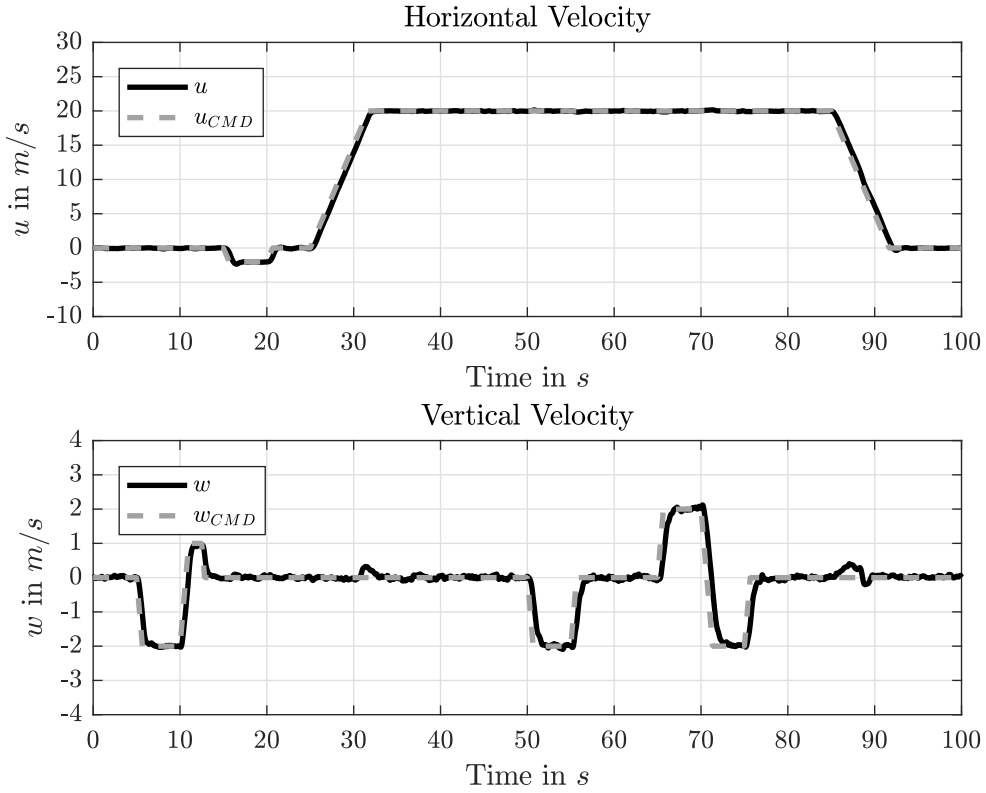

Figure 4.2: Translational States of the Baseline Simulation

effect of this change also quite low. The baseline controller also is robust against the inclusion of white noise in the sensor measurements. Since all model parameters are assumed to be perfectly known for the baseline simulation, the extension of the pseudo control estimation with a complementary filter improves the pseudo control estimates. Consequently the inclusion of the complementary filter only improves the overall controller performance for the baseline simulation and reduces effects of the considered sensor errors due to the second change.

During hover there is a $2\ m/s$ ascent and a $1\ m/s$ descent command. There also is a $2\ m/s$ backward velocity command. From now on, let the hover phase denote the first 25 seconds of the simulation. Over the next phase, the so called transition ($25\ s - 35\ s$, figure 4.2), the vehicle accelerates to $20\ m/s$ forward velocity without any vertical velocity commands. The following wingborne phase ($35\ s - 85\ s$) keeps the forward velocity constant at $20\ m/s$. Over the airborne flight ascent and descent commands of $2\ m/s$ are commanded. The final phase, the so called back transition or braking begins approximately at 85 seconds of the simulation and takes 10 seconds. During braking the vehicle deccelerates back to zero forward velocity without any commands in the vertical channel. The remaininng 5 seconds of the simulation are just zero velocity hover.

As already mentioned above, the performance of the baseline controller is similar to the performance of the controller in [**?**]. Nevertheless, the inclusion of sensor noise leads to noise in the command tracking as well, as figure 4.2 shows.

Figure 4.3 shows the pitch channel states θ and q. Since the pitch rate and pitch angle are

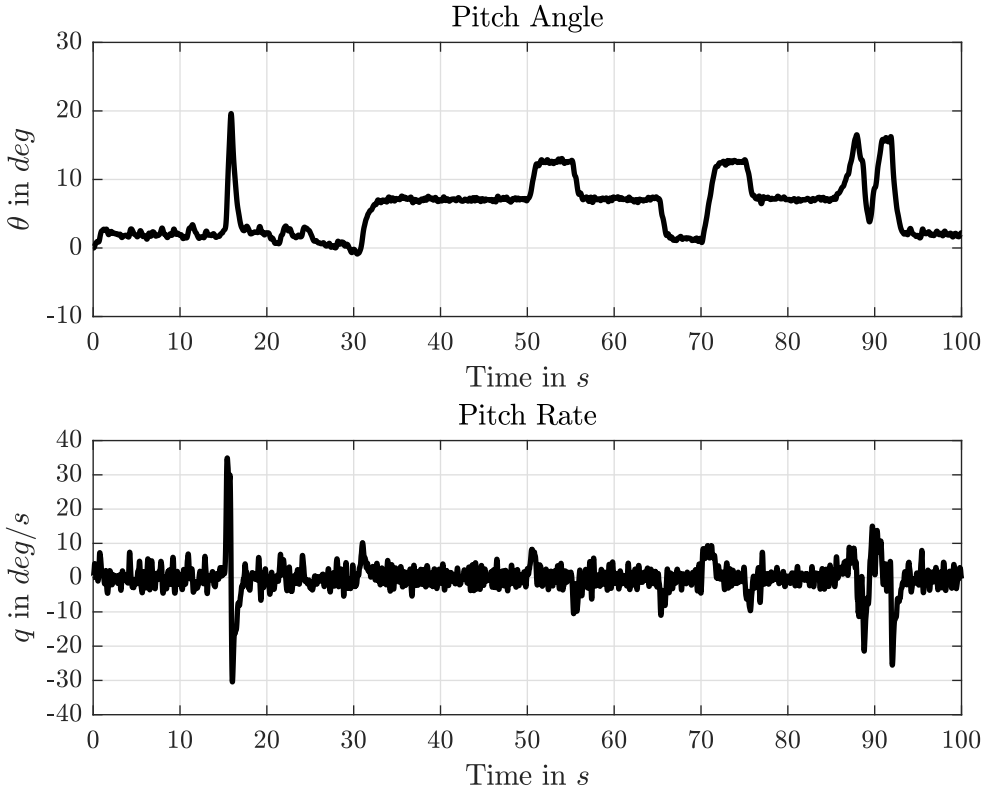

Figure 4.3: Pitch Channel of the Baseline Simulation

no pilot commands, pilot command tracking is not investigated here. The commands for the pitch channel result from the virtual control output θ_{cmd} passed through the reference model. During wingborne without commands in the vertical channel, the pitch angle is at a constant and the vehicle trims itself. For the ascent and descent commands during ariborne flight, the controller uses changes in the pitch angle to achieve the required accelerations. In hover, the constraints keep the pitch angle at zero. Since the front propeller cannot produce backwards thrust, θ is only used in hover for braking as the quite sharp build up of the pitch angle at around 15 seconds shows. Over the transition, the controller slowly increases the pitch angle until it reaches the trim value for wingborne flight. For braking from wingborne to hover the controller also uses an increase in pitch angle. In the beginning of the maneuver θ mainly increases aerodynamic drag while for the lower velocities at the end of the braking the effect of turning the main propellers is dominant. Interestingly, after increasing initially, the pitch angle drops back to zero for the braking process and increases again afterwards. The same behavior was seen in [**?**].

Figure 4.4 shows the commanded controls for the baseline simulation. As one can see, the controller does not use the elevator during hover phase and controls pitch just with the main propellers like a quadrocopter. In wingborne mode, the main propellers are turned off and the aerodynamic surfaces take over control of the height and pitch channel. The front propeller applies thrust over the whole flight envelope.

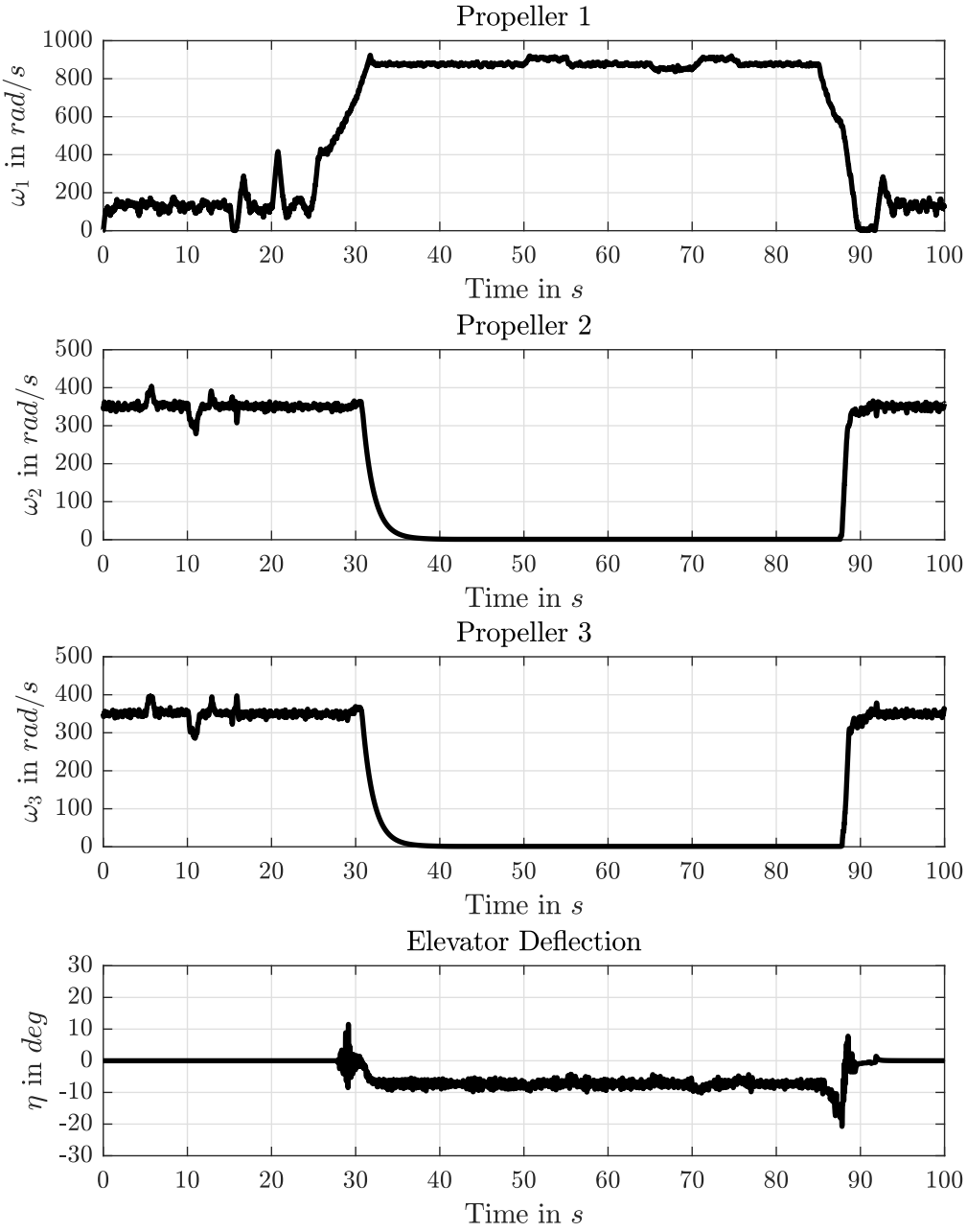

Figure 4.4: Main Propellers

4.2.2 Filter Covariance Consistency

Since the main focus of this thesis is to achieve improved closed loop performance, the actual performance of the estimation is not one of the main interests. An optimal estimator (Kalman filter) chooses the covariances in process noise and measurement noise in a way, that the statistics fit the real uncertainties of the considered system [**?**]. By doing so, the Kalman filter is statistically consistent, meaning that the estimated covariances of the estimation error actually are the covariances of the real estimation error. Since we are interested in closed loop performance, this criteria will be dropped for the design of the filter. Nevertheless, the

consistency of the estimated covariances of the extended Kalman filter will be investigated. In this thesis, the covariances of the filter will not be used outside of the filter. For future works using the covariances outside of the filter might be an option, which is why - although not used for the fitler tuning - the consistency of the filter covariances will be validated.

The Kalman filter covariance consistency can be easily verified by comparing the estimation error and upper and lower error bounds specified by the error covariance.

4.2.3 Nonlinear Closed Loop Performance Measure

After a simulation of the nonlinear closed loop there are many possible measures of the control performance. The most intuitive measures are errors between commanded and achieved states, for example the error between commanded kinematic velocity by the pilot and the real achieved velocity. The problem of latter error is the complete neglection of the pitch channel. Undesired pitch behavior might not be directly observable with this error. For this reason this thesis focusses on the error between reference model states and real states. Since pseudo control hedging pulls the reference model states towards the real plant states, the hedging might cover undesired behavior of the real plant in the aforementioned error. In order to prevent this an additional unhedged reference model is simulated in parallel and used for controller performance assessment. Let $\boldsymbol{e}$ denote the error between unhedged reference model states and the real plant states.

$$\begin{bmatrix} e_u \\ e_w \\ e_\theta \\ e_q \end{bmatrix} = \begin{bmatrix} u_{ref} \\ w_{ref} \\ \theta_{ref} \\ q_{ref} \end{bmatrix} - \begin{bmatrix} u \\ w \\ \theta \\ q \end{bmatrix} \tag{4.40}$$

This error encorporates the full desired behavior of the closed loop system. Since this error is a fourdimensional time depending variable, time domain norms and corresponding weights for a scalar control performance measure are required. The norms used in this thesis are a 2-norm and an infinity norm. The 2-norm measures the deviation of reference model state and real state over the whole flight. The equations for the 2-norm of the errors are

$$\begin{aligned} \|e_u\|_2 &= \left(\int_{t_1}^{t_2} e_u(t)^2 dt \right)^{\frac{1}{2}} \\ \|e_w\|_2 &= \left(\int_{t_1}^{t_2} e_w(t)^2 dt \right)^{\frac{1}{2}} \\ \|e_\theta\|_2 &= \left(\int_{t_1}^{t_2} e_\theta(t)^2 dt \right)^{\frac{1}{2}} \\ \|e_q\|_2 &= \left(\int_{t_1}^{t_2} e_q(t)^2 dt \right)^{\frac{1}{2}} \end{aligned} \tag{4.41}$$

with the start and stop times of the simulation t_1 and t_2 respectively. The infinity norm only measures the worst case error over the whole flight time.

$$\begin{aligned} \|e_u\|_\infty &= \sup|e_u(t)| \\ \|e_w\|_\infty &= \sup|e_w(t)| \\ \|e_\theta\|_\infty &= \sup|e_\theta(t)| \\ \|e_q\|_\infty &= \sup|e_q(t)| \end{aligned} \tag{4.42}$$

Equations (4.41) and (4.42) give eight terms to measure the error. The question that arises now is how to build a scalar performance measure of these eight terms.

The scalar error measure e will follow

$$\begin{aligned} e = \max\Bigg(& \frac{\|e_u\|_2}{\|e_u\|_{2,max}}, \frac{\|e_w\|_2}{\|e_w\|_{2,max}}, \frac{\|e_\theta\|_2}{\|e_\theta\|_{2,max}}, \frac{\|e_q\|_2}{\|e_q\|_{2,max}}, \\ & \frac{\|e_u\|_\infty}{\|e_u\|_{\infty,max}}, \frac{\|e_w\|_\infty}{\|e_w\|_{\infty,max}}, \frac{\|e_\theta\|_\infty}{\|e_\theta\|_{\infty,max}}, \frac{\|e_q\|_\infty}{\|e_q\|_{\infty,max}} \Bigg) \end{aligned} \tag{4.43}$$

where the quantities in the denominators denote the maximum allowed value for the respective error term. The maximum formulation was chosen to mainly penalize the worst case channel. An error measure higher than one consequently means non acceptable behavior in at least one of the eight error terms. An error measure of 0.6 means that for the worst error term the value of the error term is 60% of the maximum allowed value.

The weights in equation (4.43) are given in table 4.1.

2 - norm				
	$\|e_u\|_2$	$\|e_w\|_2$	$\|e_\theta\|_2$	$\|e_q\|_2$
Unit	m/s	m/s	deg	deg/s
Maximum	1	1	1	1
∞ - norm				
	$\|e_u\|_\infty$	$\|e_w\|_\infty$	$\|e_\theta\|_\infty$	$\|e_q\|_\infty$
Unit	m/s	m/s	deg	deg/s
Maximum	1	1	1	1

Table 4.1: Weights for Performance Measure

4.2.4 Nonlinear Robustness Assessment

The uncertainties for the nonlinear robustness assesment follows a similar structure as the linear robustness analysis. The considered uncertainties are parametric uncertainties in the flight dynamics model, sensor noise effects and actuator dynamics uncertainties. The parametric uncertainties are deviations in the parameters of the equations of motion. The deviations are

modelled as zero mean gaussian distributed bias errors. The standard deviations are given in table 4.2. Most of the modelled standard deviations are 20% of the nominal value of the parameter. Since quantities such ass mass, intertia and geomtric distances are easier to determine offline than for example aerodynamic coefficients, the considered corresponding uncertainties are chosen to be smaller. In the real application the onboard plant model is always perfectly known, whereas the real plant remains uncertain. Applying the uncertainties to the real plant for the robustness analysis discovers any problems with the total closed loop. These problems also include actual controllability problems with the real plant. If for example the uncertainties in the effectivenesses of the main propellers leads to small real effectivenesses, the aircraft might physically not be able to hover. Since this thesis mainly analyses robustness due to uncertainties in order to compare different controller approaches, the uncertainties are applied in another way here. Here, the uncertainties are included in the estimated model dynamics. The real plant remains unchanged whereas the parameter deviations only apply to the onboard plant model. This approach keeps the actual capabilities and the controllability of the plant constant and does not lead to problems of actual physical constrollability of the plant. Consequently only the robustness of the different controllers with respect to errors in the OBPM is considered.

Parameter Deviation		
Parameter	Unit	Standard Deviation σ
I_{yy}	kgm^2	0.05
$\left(r^{GR}\right)_B$	m	0.01
$\left(r^{GR}\right)_B$	m	0.01
ρ	kg/m^3	0.1
$c_{L\alpha,wing}$	-	0.1
$c_{L\alpha,elev}$	-	0.1
$c_{D0,wing}$	-	0.1
$c_{D0,wing}$	-	0.1
$c_{D0,wing}$	-	0.1
k_{wing}	-	0.1
k_{elev}	-	0.01
k_{P_1}	kgm/rad^2	10^{-5}
k_{P_2}	kgm/rad^2	$2 \cdot 10^{-5}$
k_{P_3}	kgm/rad^2	$2 \cdot 10^{-5}$

Table 4.2: Expected Standard Deviation of Parameters

Table 3.1 specifies the uncertainty due to the sensor cut. In addition to the general white noise error for the measurements a bias like error and a time delay are considered for robustness analysis.

The uncertainty in the actuator dynamics lies within the eigenfrequency and the damping of the actuator models. For the same reasons as for the parametric uncertainties in the equation of motion, the uncertainties in the actuator dynamics are applied in the onboard estimated actuator model. Unlike for the parametric uncertainties just worst case behavior is performed here. The modelled actuator eigenfrequency deviation is $\pm 20\%$ of the nominal value. The effect of the damping will be neglected here.

4.3 Online Parameter Estimation

Besides the ability to handle nonlinear system dynamics, online parameter estimation is one of the main advantages of the extended Kalman filter. This section focusses on determining a reasonable parameter set for online estimation. Therefore, an impact analysis of estimation errors on the baseline controller is performed. The results from this analysis are the desired parameters that are to be estimated in an ideal situation. Since not all parameters will be observable at the same time and the filter should not contain too many states, a reasonable tradeoff has to be found between desired parameter estimation and feasible parameter estimation. For this reason the section about observability analysis performs an analysis on how feasible different parameter estimation schemes are.

4.3.1 Impact of Parameters on Controller Performance

This section determines the impact of onboard plant model parameters on the total closed loop performance. Therefore, the parameters are deviated one by one in the positive and negative direction and the baseline closed loop model is simulated

$$\begin{aligned} p_i^+ &= p_i^{nom} + \delta_i^+ \sigma_{p_i} \\ p_i^- &= p_i^{nom} - \delta_i^- \sigma_{p_i} \end{aligned} \tag{4.44}$$

where p_i^+ denotes the positive deviated ith parameter, p_i^{nom} the nominal parameter value and δ_i^+ the positive perturbation. The deviations are increased in both directions until the closed loop error measure e reaches the critical value of one. Let $\delta_i^{+,crit}$ denote the critical positive deviation leading to $e = 1$.

The gaussian distributions from table 4.2 determine the probability for δ_i to be in the acceptale range $[\delta_i^{-,crit} ... \delta_i^{+,crit}]$. In this acceptable range the closed loop scalar measure e is guaranteed to be smaller than one.

$$p_i^{failure} = 1 - \frac{1}{2}\left[\operatorname{erf}\left(\frac{\delta_i^{+,crit}}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{\delta_i^{-,crit}}{\sqrt{2}}\right)\right] \tag{4.45}$$

Equation (4.45) gives the failure probability, that means the probability for an estimation error in p_i to cause failure with $e \geq 1$. Table 4.3 summarizes the results of the described analysis for all

considered parameters. The entry δ_i^{crit} denotes the maximum absolute value of the positive and negative deviation.

Parameter	δ_{crit}	p^{crit}	**Priority**
u_{wind}	1.9	5.7%	A
C_{T,P_2}	2	2.3%	B
C_{T,P_3}	2.1	1.8%	B
w_{wind}	2.7	0.3%	C
$C_{L\alpha,wing}$	3.6	$\leq 10^{-3}$	D
$C_{L\alpha,elev}$	4.2	$\leq 10^{-3}$	D
I_{yy}	5	$\leq 10^{-3}$	D
C_{T,P_1}	5	$\leq 10^{-3}$	D
ρ	7.9	$\leq 10^{-3}$	E
$C_{D0,elev}$	8	$\leq 10^{-3}$	E
r_x^{GR}	9	$\leq 10^{-3}$	E
k_{wing}	13.8	$\leq 10^{-3}$	E
r_y^{GR}	26.3	$\leq 10^{-3}$	E
m	55.1	$\leq 10^{-3}$	D
$C_{D0,wing}$	61.3	$\leq 10^{-3}$	E
$C_{D0,body}$	251.7	$\leq 10^{-3}$	E
k_{elev}	562.7	$\leq 10^{-3}$	E

Table 4.3: Parameter Impact Analysis

The priorities in table 4.3 group the different parameters according to their impact for the controller performance. Priority A means a high impact on controller performance and thus the corresponding parameter would be a very important candidate for an parameter to estimate. As the table shows, the most promising parameter to estimate for improving the controller performance is the horizontal wind velocity. The next category are the effectivenesses of the two main propellers. The last possible candidate for a parameter to estimate is the vertical wind velocity with an overall failure probability of 0.3%. All other parameters are not considered to be desired candidates for the online parameter estimation from here on, since their impact on controller performance is small.

4.3.2 Observability of Parameters

This subsection determines the observability of above parameters and combinations and determines a parameter estimation strategy. The four candidates for parameter estimation from the

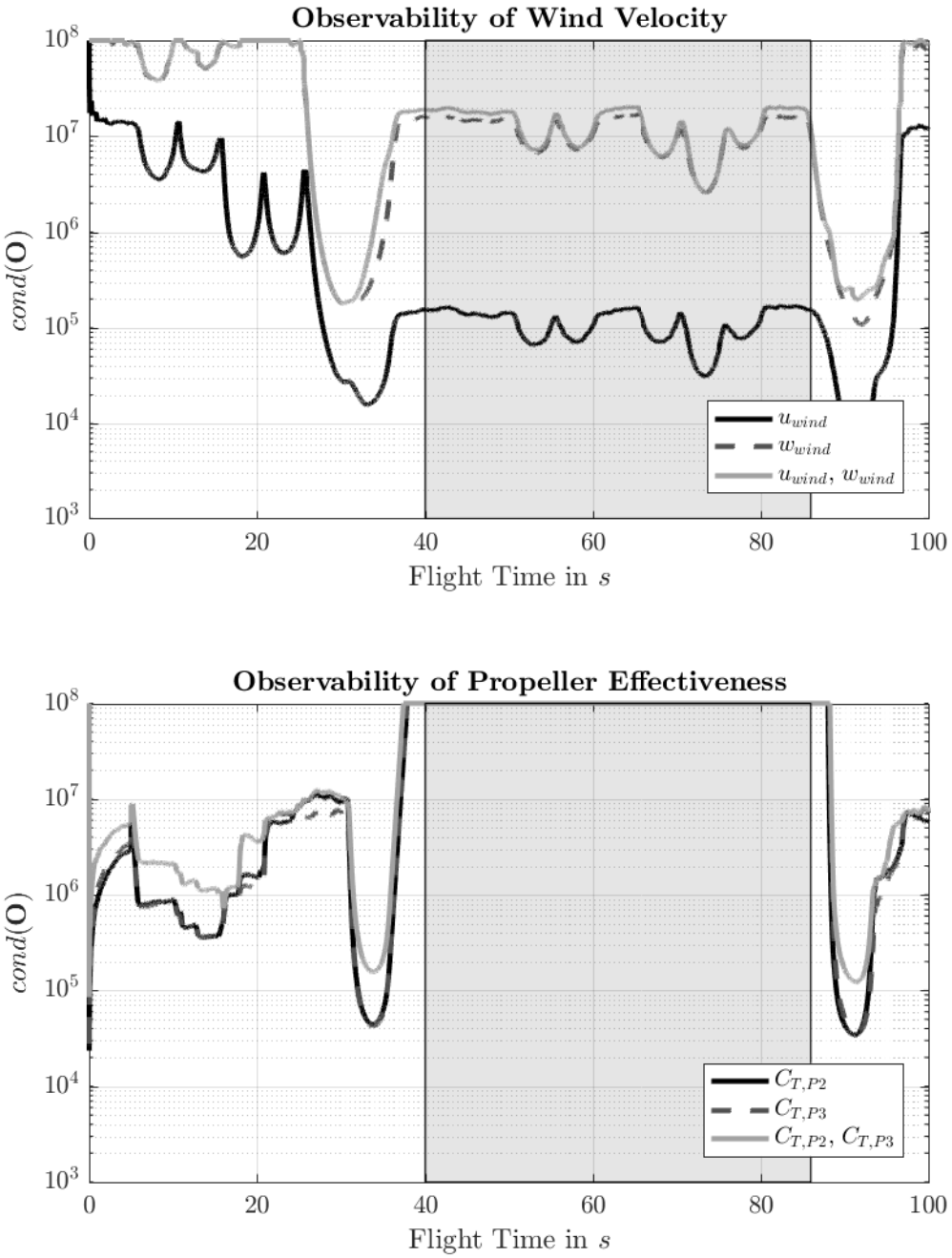

Figure 4.5: Observability of different parameter estimation schemes

last subsection are u_{wind}, w_{wind}, C_{T,P_2} and C_{T,P_3}.

The chosen observability measure for this application is the condition of the observability gramian [**?**]. The observability gramian for the discrete extended Kalman filter follows [**?**]

$$\boldsymbol{O} = \sum_{k=k_1}^{k=k_2} \left(\boldsymbol{I} + \Delta t \boldsymbol{A}_k^{extended^T} \right) \boldsymbol{C}_k^{extended} \boldsymbol{C}_k^{extended^T} \left(\boldsymbol{I} + \Delta t \boldsymbol{A}_k^{extended} \right) \Delta t \tag{4.46}$$

where $\boldsymbol{O}$ denotes the observability gramian, k the discrete time index, and the matrices $\boldsymbol{A}_k^{extended}$, $\boldsymbol{C}_k^{extended}$ the matrices for the extended Kalman filter given in equation (2.28).

The condition of the observability gramian is calculated for the different parameter sets. The time interval of the integration of the gramian is one second or 200 controller steps respectively.

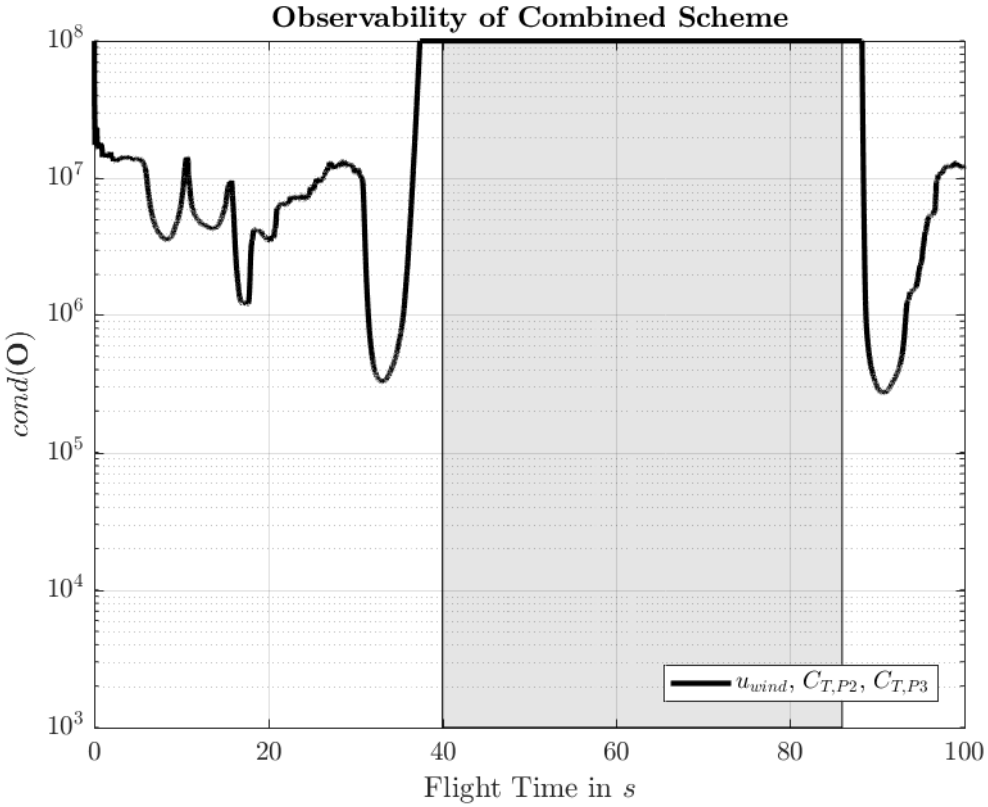

Figure 4.6: Observability of a combined parameter estimation scheme

Figure 4.5 shows the condition of the observability gramian over the baseline simulation. The required matrices for the extended Kalman filter are obtained with the onboard plant model. Obviously, the effictivenesses of the main propellers are not observable in wingborne flight (shaded region in figure 4.5), since the main propellers are shut down. In hover and during transition/braking, the effectivenesses are equally observable independently as well as together. During hover and low speeds especially the vertical wind speed is inadequately observable. The horizontal wind speed is observable over all flight phases, the best observability is found in wingborne flight. The reason for the high values of the condition is the difference between

state and parameter estimation. While states are measured directly, the parameters have to be measured by state derivatives, which is a state difference over the sampletime of the controller. Since the sampletime of the controller is quite small $(0.005\ s)$, a small error in a state measurement causes bigger errors in the parameter estimation. Consequently the overall condition of the observability will be high.

Because of the dependency of the observability of the different parameters on the current flight state (wingborne or hover) a hybrid parameter estimation approach is proposed. The hybrid parameter approach estimates the propeller effictiveness in hover mode and switches to wind estimation in wingborne flight. Since the horizontal wind velocity also was observable during hover, figure 4.6 shows the observability of a possible combined estimation of horizontal wind and main propeller effectiveness during hover. The observability of this combined estimation scheme during hover is worse than the pure estimation of the main propeller effectiveness. For this reason, only the propeller effectiveness will be estimated during hover from now on.

4.4 Practical Considerations

This section focusses on a short summary of practical requirements. The main practical requirments are the ability to turn on and off the filter during operation. This is required for takeoff and landing, since groundcontact is a very discontinuous problem which is not modelled in the onboard plant model. Another concern is the diagnostics of the model filter. What happens when an estimated parameter reaches extreme values and the filter obviously is not operating stable? The last subsection focusses on the computational burden of the filter.

4.4.1 Filter Logics

For a practical implementation in a real aircraft, a possible flight management system shall be able to manage the model filter. One very important thing to mention here is to turn off the filter for take off and landing, since here, groundcontacts completely change the dynamics of the vehicle, which makes the complete model filtering approach invalid. As it might be an option to estimate different parameters during different flight phases, one shall also be able to turn on and off individual channels for parameter estimation. Another scenario that requires a shutdown of the model filter is, if the model filter shows undesired behavior, e.g. the covariances or the state estimates turn out to be high. The next subsection therefore focusses on diagnostics of errorneos behavior of the filter.

4.4.2 Filter Diagnostics

The filter shall include a diagnostic unit which is capable to output flags concerning validity of signals and possible error flags. Possible flags include flags for high values of estimated parameters, high differences between measurements and estimated states as well as high values for estimated covariances of the filter. The required actions for possible errors or malfunction of the model filter shall not be done by the filter itself but from the outside, e.g. a flight management

system. The flightmanagement system uses the output flags from the filter and uses the implemented filter switches to perform the corresponding actions.

4.4.3 Computational Cost

Since the considered application in this thesis is just two-dimensional and the model filter is mainly set up for a first analysis, the computational cost is not the main interest for this application. Nevertheless, it is planned for future works to extend the modelfilter for real three-dimensional applications. For this reason reducing the computational cost is an important aspect in designing a feasible model filter. Therefore, some simplifications and ideas will be implemented and investigated that address the computational cost issue.

Chapter 5

Implementation of the Model Filter

This chapter focusses on the implementation of the extended Kalman filter. Therefore, a first overview of the filter is given, then the detailed kalman filter equations for propagation and correction are introduced. The following section then defines process and measurment noise covariances. The final part of this chapter puts special emphasis on reducing computational effort and introduces two ideas for reducing it.

5.1 Overall Strategy and Ideas

The used filter is an extended Kalman filter that estimates states and parameters. The estimated states are the four states of the real vehicle. The estimated parameters are horizontal and vertical wind as well as the effectivenesses of the main propellers.

$$\boldsymbol{p} = \begin{bmatrix} u_{wind} \\ w_{wind} \\ C_{TP_1} \\ C_{TP_2} \end{bmatrix} \tag{5.1}$$

The parameter estimation scheme is hybrid, meaning that during wingborne wind is estimated while during hover the propeller effectivenesses are estimated. The Kalman filter does not estimate total parameter values but deviations from the assumed nominal value. In order to avoid highly differing orders of magnitude for the different parameters and covariances, the parameters are normalized with their respective standard deviations from table 4.2.

$$p_i = \sigma_{p_i} \hat{p}_i \tag{5.2}$$

An estimated unit parameter deviation means a deviation of one standard deviation for the real parameter.

For the prediction of the states the state time derivative data from the onboard plant model is used and integrated with a euler forward scheme. The required derivative information for the extended Kalman filter approach is also obtained by the onboard plant model.

5.2 Kalman Filter Equations

This section focusses on the implementation of the Kalman filter equations. The extended Kalman Filter is implemented in SIMULINK. Implementing the Kalman filter equations in an computationally efficient manner is a special focus of this section.

5.2.1 Propagation

This subsection summarizes the propagation equations for the extended Kalman filter. The propagation of the states follows an euler forward integration of the OBPM dynamics, the parameters reamin constant:

$$\begin{aligned} \boldsymbol{x}_{k+1|k} &= \boldsymbol{x}_{k|k} + \dot{\boldsymbol{x}}_{OBPM}\Delta t \\ \boldsymbol{p}_{k+1|k} &= \boldsymbol{p}_{k|k} \end{aligned} \tag{5.3}$$

The matrix products in the covariance propagation often require high computational effort. Exploiting the structure of the model dynamics and the state transition matrix helps reducing this effort, which is applied in the following equations. The linear approximation for the exponential of the transition matrix leads to:

$$\phi = \boldsymbol{I} + \boldsymbol{A}\Delta t \tag{5.4}$$

with $\boldsymbol{A}$ being the linear system dynamics matrix. Inserting the extended system matrix for the parameter estimation with an EKF (equation (2.25)) results in

$$\phi = \begin{bmatrix} \boldsymbol{I} + \Delta t \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}|_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{p}}_k} & \Delta t \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}|_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{p}}_k} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \phi_x & \phi_p \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{5.5}$$

The covariance $\boldsymbol{P}$ can also be divided in the same manner in states and parameters:

$$\boldsymbol{P}_{k-1,k-1} = \begin{bmatrix} \boldsymbol{P}^{xx}_{k-1,k-1} & \boldsymbol{P}^{xp}_{k-1,k-1} \\ \boldsymbol{P}^{px}_{k-1,k-1} & \boldsymbol{P}^{pp}_{k-1,k-1} \end{bmatrix} \tag{5.6}$$

Dividing the process noise in the same manner and inserting equations (5.5) and (5.6) into the covariance propagation equation (2.8) leads to

$$\boldsymbol{P}_{k,k-1} = \begin{bmatrix} \phi^x & \phi^p \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{P}^{xx}_{k-1,k-1} & \boldsymbol{P}^{xp}_{k-1,k-1} \\ \boldsymbol{P}^{px}_{k-1,k-1} & \boldsymbol{P}^{pp}_{k-1,k-1} \end{bmatrix} \begin{bmatrix} \phi^{xT} & \boldsymbol{0} \\ \phi^{pT} & \boldsymbol{I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{Q}^{xx}_k & \boldsymbol{Q}^{xp}_k \\ \boldsymbol{Q}^{px}_k & \boldsymbol{Q}^{pp}_k \end{bmatrix} \tag{5.7}$$

By using the symmetry of the covariances the covariance propagation can be performed on the trhee submatrices $\boldsymbol{P}^{xx}$, $\boldsymbol{P}^{xp}$ and $\boldsymbol{P}^{pp}$. The propagation for the parameter covariance follows

$$\boldsymbol{P}^{pp}_{k,k-1} = \boldsymbol{P}^{pp}_{k-1,k-1} + \boldsymbol{Q}^{pp}_k \tag{5.8}$$

Extracting the propagation for the crosscovariance between states and parameters leads to

$$\boldsymbol{P}^{xp}_{k,k-1} = \phi^x \boldsymbol{P}^{xp}_{k-1,k-1} + \phi^p \boldsymbol{P}^{pp}_{k-1,k-1} + \boldsymbol{Q}^{xp}_k \tag{5.9}$$

The state covariance propagation reads

$$\boldsymbol{P}^{xx}_{k,k-1} = \left(\phi^x \boldsymbol{P}^{xx}_{k-1,k-1} + \phi^p \boldsymbol{P}^{px}_{k-1,k-1}\right) \phi^{xT} + \left(\phi^x \boldsymbol{P}^{xp}_{k-1,k-1} + \phi^p \boldsymbol{P}^{pp}_{k-1,k-1}\right) \phi^{pT} + \boldsymbol{Q}^{xx}_k \tag{5.10}$$

Note that the term in the second bracket of equation (5.10) equals the term in equation (5.9). Exploiting this can further decrease the required computational cost.

5.2.2 Correction

For the correction the same division into states and parameters is used as for the propagation equations. Since for the applied extended Kalman filter the estimated states and the measurements coincide and the parameters are not measured the measurement matrix $\boldsymbol{H}$ equals

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \end{bmatrix} \tag{5.11}$$

Using the same strucuture as for the covariance propagation and inserting in equation (2.10) leads to the Kalman gain

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{P}^{xx}_{k-1,k-1} & \boldsymbol{P}^{xp}_{k-1,k-1} \\ \boldsymbol{P}^{px}_{k-1,k-1} & \boldsymbol{P}^{pp}_{k-1,k-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \mathbf{0} \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{P}^{xx}_{k,k-1} & \boldsymbol{P}^{xp}_{k,k-1} \\ \boldsymbol{P}^{px}_{k,k-1} & \boldsymbol{P}^{pp}_{k,k-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \mathbf{0} \end{bmatrix} + \boldsymbol{R} \right)^{-1} \tag{5.12}$$

Simplifying above equation and dividing the Kalman gain into a state and a parameter part results in

$$\begin{bmatrix} \boldsymbol{K}^x \\ \boldsymbol{K}^p \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}^{xx}_{k,k-1} \\ \boldsymbol{P}^{px}_{k,k-1} \end{bmatrix} \left(\boldsymbol{P}^{xx}_{k,k-1} + \boldsymbol{R} \right)^{-1} \tag{5.13}$$

Inserting this into the covariance correction equation (2.11) leads to

$$\boldsymbol{P}_{k,k} = \left(\boldsymbol{I} - \begin{bmatrix} \boldsymbol{K}^x \\ \boldsymbol{K}^p \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \end{bmatrix} \right) \boldsymbol{P}_{k,k-1} \tag{5.14}$$

As for the covariance propagation the division into submatrices helps to reduce the computational workload. The state covariance correction reads

$$\boldsymbol{P}^{xx}_{k,k} = \left(\boldsymbol{I} - \boldsymbol{K}^x \right) \boldsymbol{P}^{xx}_{k,k-1} \tag{5.15}$$

The state parameter cross covariance correction equation is

$$\boldsymbol{P}^{xp}_{k,k} = \left(\boldsymbol{I} - \boldsymbol{K}^x \right) \boldsymbol{P}^{xp}_{k,k-1} \tag{5.16}$$

And finally, the parameter covariance correction follows

$$\boldsymbol{P}^{pp}_{k,k} = \boldsymbol{P}^{pp}_{k,k-1} - \boldsymbol{K}^p \boldsymbol{P}^{xp}_{k,k-1} \tag{5.17}$$

The state and parameter correction follows as

$$\begin{aligned} \boldsymbol{x}_{k+1|k+1} &= \boldsymbol{x}_{k+1|k} + \boldsymbol{K}^x_k \left(\boldsymbol{x}_{meas} - \boldsymbol{x}_{k+1|k} \right) \\ \boldsymbol{p}_{k+1|k+1} &= \boldsymbol{p}_{k+1|k} + \boldsymbol{K}^p_k \left(\boldsymbol{x}_{meas} - \boldsymbol{x}_{k+1|k} \right) \end{aligned} \tag{5.18}$$

5.3 Measurement and Process Noise Covariance

Measurement and process noise coavariances are the parameters to tune the performance of Kalman filters. When it comes to optimal estimation, keeping the process noise and measurement noise as close to the noise covariances in reality is the preferred option [**?**]. Since this thesis puts more emphasis on the total closed loop performance and not the estimation performance of the filter itself, at least the process noise covariance is seen as tuning parameter and will not reflect the real noise characteristics of the system. The following subsections define the choice of measurement and process noise covariance.

5.3.1 Measurement Noise Covariance R

In this application the measurement noise corresponds to the modelled sensor covariances. In a real application, the measurements will be output of an inertial navigation system (INS), since there are no sensors for kinematic velocities or euler angles. Since most inertial navigation systems also use an extended Kalman filter that also estimates a state covariance, this state covariance is a reasonable candidate for a measurement noise covariance for the model filter. For this simplified application, let the discrete time measurement noise covariance equal the covariance of the sensors:

$$\boldsymbol{R} = \begin{bmatrix} \sigma^2_{WN,u} & 0 & 0 & 0 \\ 0 & \sigma^2_{WN,w} & 0 & 0 \\ 0 & 0 & \sigma^2_{WN,\theta} & 0 \\ 0 & 0 & 0 & \sigma^2_{WN,q} \end{bmatrix} \tag{5.19}$$

5.3.2 Measurement Noise Covariance Q

This subsection focusses on defining the process noise covariance $\boldsymbol{Q}$. Unlike the measurement noise, which is chosen to match the noise characteristics of the sensors completely, the process noise covariance will be left open as a tuning parameter. Therefore, the process noise covariance will be separated into a constant part and a varying part. The varying part depends on states and controls and takes account for flight state dependent uncertainties.

Constant Part

The most general approach for setting up a tunable constant process covariance for the states just takes into account the symmetry, leading to ten tuning parameters:

$$\boldsymbol{Q}_{xx,const} = \begin{bmatrix} \mu_1 & \mu_5 & \mu_6 & \mu_7 \\ \mu_5 & \mu_2 & \mu_8 & \mu_9 \\ \mu_6 & \mu_8 & \mu_3 & \mu_{10} \\ \mu_7 & \mu_9 & \mu_{10} & \mu_4 \end{bmatrix} \Delta t \tag{5.20}$$

with the tuning vector $\boldsymbol{\mu}$ and the sample time of the controller and Kalman filter Δt. It is very important to note, that covariances are always positive definite. This is an additional condition for the choice of $\boldsymbol{\mu}$. The additional process noise for the parameter estimation follow

$$\begin{aligned} \boldsymbol{Q}_{pp} &= \begin{bmatrix} \mu_{11} & 0 & 0 & 0 \\ 0 & \mu_{12} & 0 & 0 \\ 0 & 0 & \mu_{13} & 0 \\ 0 & 0 & 0 & \mu_{14} \end{bmatrix} \Delta t \\ \boldsymbol{Q}_{xp} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Delta t \end{aligned} \tag{5.21}$$

The process noise in the parameters $\boldsymbol{Q}_{pp}$ does not have a physical meaning. Since parameters usually are supposed to be constants, there is no change over time and the time derivative equals exactly zero without any uncertainty. The first reason for including this covriance is to avoid convergence of the estimated parameter covariances to zero. The second reason is to be able to handle a wind estimation as parameter estimation as well. No parameter process noise in the wind estimation would lead to very slow reaction to changes in the wind. Since wind is not a normal constant parameter but changes over time there has to be a parameter process noise.

For the sake of simplification the cross process noise covariance between states and parameters is chosen to be zero. The parameter process noise is chosen to be a diagonal. In order for the full process noise covariance to be positive definite the diagonal elements of $\boldsymbol{Q}_{pp}$ have to be greater than one and $\boldsymbol{Q}_{xx}$ has to be positive definite.

Varying Part

So far, only a constant white process noise is represented by the above choice of process noise. On of the main uncertainties of the system, the uncertainty due to parametric uncertainties in the equations of motion do not follow this assumption of a constant white noise. First of all, the parametric uncertainties depend on the states and controls of the system. For example a high uncertainty in the effictiveness of a propeller leads to a higher uncertainty in the states deriviatives if the propeller is used more than when it is almost turned off. Second of all, a parametric uncertainty does not lead to white noise characteristics. A parametric uncertainty is a bias like uncertainty that leads to highly time correlated noise in the system dynamics.

An approach to deal with the second problem is to extend the states by the uncertain parameters and performs a parameter estimation. Since the number of uncertain parameters is quite high and there is no reasonable observability of all the parameters in this application, this approach is not feasible. A promising way to deal with the bias like errors in the system dynamics is to not estimate all the uncertain parameters but to include the uncertainty caused by them in the estimated covariances [**?**].

Since the modelling of the process noise does not focus on full statistical consistency here, the fact that parametric uncertainties cause bias like errors and not white noise errors is neglected here. The process noise part corresponding to the parametric uncertainties is modelled as a classic white noise process noise.

The problem that the uncertainties are not constant but change over time can be dealt with a linearization. Performing a linearization of the time derivatives of the states with respect to the paramters leads to

$$\tilde{\dot{\boldsymbol{x}}} = \boldsymbol{J}\boldsymbol{p} \tag{5.22}$$

where $\tilde{\dot{\boldsymbol{x}}}$ denotes the error in the state derivatives due to parametric uncertainties $\boldsymbol{p}$ and $\boldsymbol{J}$ the corresponding Jacobian. Taking the covariance of above equation and discretizing leads to the varying process noise due to parametric uncertainties:

$$\boldsymbol{Q}_{xx,varying} = \boldsymbol{J}\,\mathrm{cov}\,(\boldsymbol{p})\boldsymbol{J}^T\Delta t \tag{5.23}$$

The covariance of the parameters is given by table 4.2. The following part derives the entries of the jacobian $\boldsymbol{J}$.

The folllowing derivation of the jacobian neglects the uncertainty in the center of gravity, the mass and the inertia.

The 'lowest' uncertainty that even affects the kinematics of the system is the wind. The derivatives of the aerodynamic flight path angle with respect to the wind components follows according to equation (3.15)

$$\begin{aligned}\frac{\delta\gamma_A^G}{\delta\left(u_W^G\right)_O^E} &= \frac{-\left(w_A^G\right)_O^E}{\left[\left(u_A^G\right)_O^E\right]^2+\left[\left(w_A^G\right)_O^E\right]^2} = -\frac{\sin\gamma_A}{\|v_A\|}\\ \frac{\delta\gamma_A^G}{\delta\left(w_W^G\right)_O^E} &= \frac{\left(u_A^G\right)_O^E}{\left[\left(u_A^G\right)_O^E\right]^2+\left[\left(w_A^G\right)_O^E\right]^2} = \frac{\cos\gamma_A}{\|v_A\|}\end{aligned} \tag{5.24}$$

As for the aerodynamic flight path angle, wind also has an influence on the aerodynamic angle of attack

$$\begin{aligned}\frac{\delta\alpha_A^G}{\delta\left(u_W^G\right)_O^E} &= -\frac{\delta\gamma_A^G}{\delta\left(u_W^G\right)_O^E} = \frac{\sin\gamma_A}{\|v_A\|}\\ \frac{\delta\alpha_A^G}{\delta\left(w_W^G\right)_O^E} &= -\frac{\delta\gamma_A^G}{\delta\left(w_W^G\right)_O^E} = -\frac{\cos\gamma_A}{\|v_A\|}\end{aligned} \tag{5.25}$$

Wind also affects the dynamic pressure by changing the overall aerodynmic speed

$$\begin{aligned}\frac{\delta\bar{q}}{\delta\left(u_W^G\right)_O^E} &= -\rho\left(u_A^G\right)_O^E\\ \frac{\delta\bar{q}}{\delta\left(w_W^G\right)_O^E} &= -\rho\left(w_A^G\right)_O^E\\ \frac{\delta\bar{q}}{\delta\rho} &= \frac{\bar{q}}{\rho}\end{aligned} \tag{5.26}$$

The next part in the system dynamics to be analyzed are the aerodynamic lift and drag coefficients. For the sake of simplification the derivative of the $C_{L\alpha}$ - blending with respect to the angle of attack is neglected:

$$\frac{\delta\lambda_{C_L}}{\delta\alpha} \approx 0 \tag{5.27}$$

This assumption is valid for wide ranges of the angle of attack, since the blending factor λ_{C_L} is almost constant one or zero. Only for the sharp stall transition region this assumption causes errors. The derivatives of the lift coefficient of the wing then follow

$$\begin{aligned}\frac{\delta C_{L,wing}}{\delta\left(u_W^G\right)_O^E} &= C_{L\alpha,wing}\frac{\left(w_A^G\right)_O^E}{\left[\left(u_A^G\right)_O^E\right]^2+\left[\left(w_A^G\right)_O^E\right]^2} = C_{L\alpha,wing}\frac{\sin\gamma_A}{\|v_A\|}\\ \frac{\delta C_{L,wing}}{\delta\left(w_W^G\right)_O^E} &= C_{L\alpha,wing}\frac{-\left(u_A^G\right)_O^E}{\left[\left(u_A^G\right)_O^E\right]^2+\left[\left(w_A^G\right)_O^E\right]^2} = -C_{L\alpha,wing}\frac{\cos\gamma_A}{\|v_A\|}\\ \frac{\delta C_{L,wing}}{\delta C_{L\alpha,wing}} &= \alpha_A\lambda_{C_L}\end{aligned} \tag{5.28}$$

The derivatives of the aerodynamic drag coefficent of the wing follow accordingly to the lift coefficient:

$$\begin{aligned}
\frac{\delta C_{D,wing}}{\delta \left(u_W^G\right)_O^E} &= 2k_{wing}C_{L\alpha,wing}^2\alpha_A\frac{\sin\gamma_A}{\|v_A\|} \\
\frac{\delta C_{D,wing}}{\delta \left(w_W^G\right)_O^E} &= -2k_{wing}C_{L\alpha,wing}^2\alpha_A\frac{\cos\gamma_A}{\|v_A\|} \\
\frac{\delta C_{D,wing}}{\delta C_{L\alpha,wing}} &= 2k_{wing}C_{L\alpha,wing}\alpha_A^2\lambda_{C_L}^2 \\
\frac{\delta C_{D,wing}}{\delta k_{wing}} &= C_{L\alpha,wing}^2\alpha_A^2\lambda_{C_L}^2 \\
\frac{\delta C_{D,wing}}{\delta C_{D0,wing}} &= 1
\end{aligned} \tag{5.29}$$

The sensitivity of the horizontal aerodynamic forces of the wing is

$$\begin{aligned}
\frac{\delta\left(X_{A,wing}^R\right)_O}{\delta\bar{q}} &= \frac{\left(X_{A,wing}^R\right)_O}{\bar{q}} \\
\frac{\delta\left(X_{A,wing}^R\right)_O}{\delta C_{L,wing}} &= -\bar{q}S_{wing}\sin\gamma_A \\
\frac{\delta\left(X_{A,wing}^R\right)_O}{\delta C_{D,wing}} &= -\bar{q}S_{wing}\cos\gamma_A \\
\frac{\delta\left(X_{A,wing}^R\right)_O}{\delta\gamma_A} &= \bar{q}S_{wing}\left(\sin\gamma_A C_{D,wing} - \cos\gamma_A C_{L,wing}\right)
\end{aligned} \tag{5.30}$$

and the sensitivity of the vertical force of the wing is

$$\begin{aligned}
\frac{\delta\left(Z_{A,wing}^R\right)_O}{\delta\bar{q}} &= \frac{\left(Z_{A,wing}^R\right)_O}{\bar{q}} \\
\frac{\delta\left(Z_{A,wing}^R\right)_O}{\delta C_{L,wing}} &= -\bar{q}S_{wing}\cos\gamma_A \\
\frac{\delta\left(Z_{A,wing}^R\right)_O}{\delta C_{D,wing}} &= \bar{q}S_{wing}\sin\gamma_A \\
\frac{\delta\left(Z_{A,wing}^R\right)_O}{\delta\gamma_A} &= \bar{q}S_{wing}\left(\cos\gamma_A C_{D,wing} + \sin\gamma_A C_{L,wing}\right)
\end{aligned} \tag{5.31}$$

The propeller forces depend on uncertainties in the effectiveness of the propellers. The sensitivity analysis for the horizontal component leads to

$$\begin{aligned}
\frac{\delta\left(X_{A,prop}^R\right)_O}{\delta C_{T,P1}} &= \omega_1^2\cos\theta \\
\frac{\delta\left(X_{A,prop}^R\right)_O}{\delta C_{T,P2}} &= -\omega_2^2\sin\theta \\
\frac{\delta\left(X_{A,prop}^R\right)_O}{\delta C_{T,P3}} &= -\omega_3^2\sin\theta
\end{aligned} \tag{5.32}$$

The vertical force component sensitivities follow accordingly:

$$\begin{aligned}
\frac{\delta\left(Z^R_{A,prop}\right)_O}{\delta C_{T,P1}} &= -\omega_1^2 \sin\theta \\
\frac{\delta\left(Z^R_{A,prop}\right)_O}{\delta C_{T,P2}} &= -\omega_2^2 \cos\theta \\
\frac{\delta\left(Z^R_{A,prop}\right)_O}{\delta C_{T,P3}} &= -\omega_3^2 \cos\theta
\end{aligned} \tag{5.33}$$

The sensitivity of the resulting moment only shows in the effectivenesses of the two main propellers, since the front propeller does not have a lever arm with respect to the center of gravity.

$$\begin{aligned}
\frac{\delta\left(M^R_{A,prop}\right)_O}{\delta C_{T,P2}} &= \left(r_x^{RP_2}\right)_B \omega_2^2 \\
\frac{\delta\left(M^R_{A,prop}\right)_O}{\delta C_{T,P3}} &= \left(r_x^{RP_3}\right)_B \omega_3^2
\end{aligned} \tag{5.34}$$

Finally, the sensitivies of the elevator follow. The uncertainties of the lift coefficient of the elevator follow

$$\begin{aligned}
\frac{\delta C_{L,elev}}{\delta\left(u_W^G\right)_O^E} &= C_{L\alpha,elev}\frac{\sin\gamma_A}{\|v_A\|} \\
\frac{\delta C_{L,elev}}{\delta\left(w_W^G\right)_O^E} &= -C_{L\alpha,elev}\frac{\cos\gamma_A}{\|v_A\|} \\
\frac{\delta C_{L,elev}}{\delta C_{L\alpha,elev}} &= (\alpha_A+\eta)\lambda_{C_L}
\end{aligned} \tag{5.35}$$

The drag coefficients' sensitivity of the elevator is

$$\begin{aligned}
\frac{\delta C_{D,elev}}{\delta\left(u_W^G\right)_O^E} &= 2k_{elev}C^2_{L\alpha,elev}\left(\alpha_A+\eta\right)\frac{\sin\gamma_A}{\|v_A\|} \\
\frac{\delta C_{D,elev}}{\delta\left(w_W^G\right)_O^E} &= -2k_{elev}C^2_{L\alpha,elev}\left(\alpha_A+\eta\right)\frac{\cos\gamma_A}{\|v_A\|} \\
\frac{\delta C_{D,elev}}{\delta C_{L\alpha,elev}} &= 2k_{elev}C_{L\alpha,elev}\left(\alpha_A+\eta\right)^2\lambda^2_{C_L} \\
\frac{\delta C_{D,elev}}{\delta k_{elev}} &= C^2_{L\alpha,elev}\left(\alpha_A+\eta\right)^2\lambda^2_{C_L} \\
\frac{\delta C_{D,elev}}{\delta C_{D0,elev}} &= 1
\end{aligned} \tag{5.36}$$

The pitch moment of the elevator then follows.

$$\begin{aligned}
\frac{\delta\left(M^R_{A,elev}\right)_O}{\delta\bar{q}} &= \frac{\left(M^R_{A,elev}\right)_O}{\bar{q}} \\
\frac{\delta\left(M^R_{A,elev}\right)_O}{\delta C_{L,elev}} &= \left(r^{RE}_x\right)_B \bar{q} S_{elev} \cos\alpha_A \\
\frac{\delta\left(M^R_{A,elev}\right)_O}{\delta C_{D,elev}} &= \left(r^{RE}_x\right)_B \bar{q} S_{elev} \sin\alpha_A \\
\frac{\delta\left(M^R_{A,elev}\right)_O}{\delta\alpha_A} &= \left(r^{RE}_x\right)_B \bar{q} S_{elev} \left(-\sin\alpha_A C_{L,elev} + \cos\alpha_A C_{D,elev}\right)
\end{aligned} \tag{5.37}$$

For the sake of space consumption the forces of the elevator are not given here but follow acccordingly to the forces and moments of the wing.

Since the body drag has the exact same influence as the zero drag of the wing the corresponding uncertainty can simply be shifted towards the wing zero drag.

With the help of the chain rule of differentiation the above mentioned sensitivites are now gathered in the Jacobian

$$\boldsymbol{J} = \frac{\delta\dot{\boldsymbol{x}}}{\delta\boldsymbol{p}} \tag{5.38}$$

The entries of the jacobian are given in the appendix.

A way to verify the performed sensitivity analysis and the performed linearizations with neglections of higher order terms is to perform a Monte Carlo analysis of the parametric uncertainties. The system dynamics can be evaluated for normally distributed parameter deviations according to table 4.2. In the next step, the covariance of the time derivative errors of the states can be evaluated. Figure 5.1 shows the results for the diagonal of the error covariance of a numeric Monte Carlo analyiss with 10000 system dynamics evaluations. It also shows the analyitcal distributions to the state time derivatives. The system dynamics are evaluated along the baseline trajectory during transition.The first thing to mention is that the analytical sensitivity analysis and the Monte Carlo analysis match quite well here. This means the made simplifications and the linearization are reasonable.

Figure 5.2 shows the exact same quantities as figure 5.1 but is evaluated during the braking process. The first obvious observation from both maneuvers is that the main contributions of the uncertainties are within the wind and the effectiveness of the propellers. The next thing to mention is that the uncertainties in the translational acceleration channels are higher in higher velocities in wingborne flight than they are for hover. This is intuitive since for higher velocities the influence of aerodynamic effects increases. Since the dynamic pressure is quadratic in the aerodynamic velocity, wind has a higher influence than for lower velocities. The only uncertainty that decreases in wingborne flight is the uncertainty due to the main propellers as they are shut down. This uncertainty is compensated by an aerodynamic uncertainty as for wingborne flight aerodynamic surfaces have to create lift.

The uncertainty in the pitch acceleration shows a completely opposite effect compared to the translational channels. In hover there is a high uncertainty in the pitch channel because

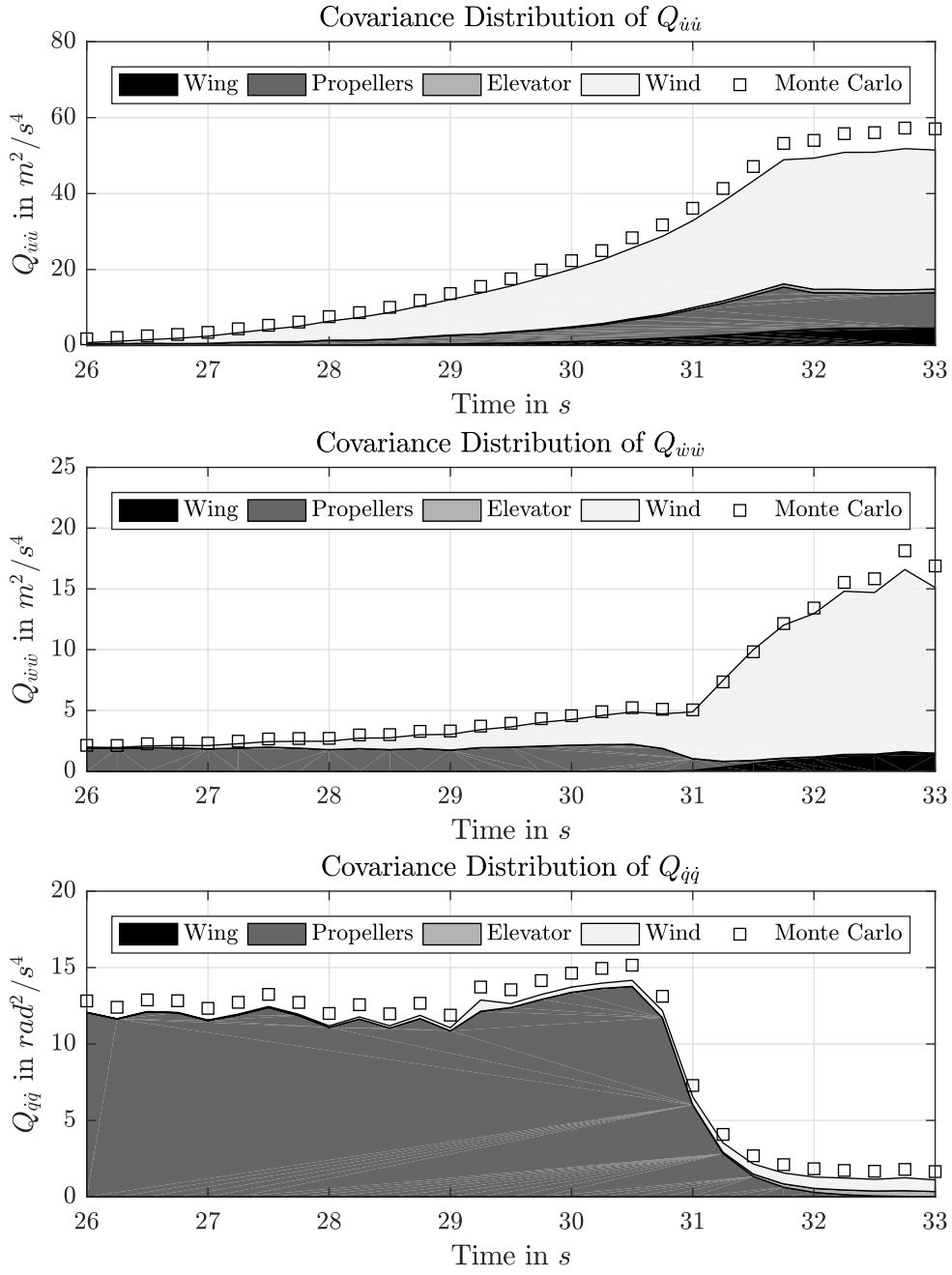

Figure 5.1: Distribution of parametric uncertainty for the state time derivatives during the transition

of the individual uncertainties in the main propellers. For wingborne flight the main propellers turn off and the moments are desired pitch moments are solely achieved by the elevator. Since in this model neither the front propeller nor the wing produce moments the trim in the pitch for airborne flight also means that the elevator will only produce marginal pitch moments. For this reason uncertainties in the elevator like for example $C_{L\alpha,elev}$ will not cause high uncertainties in the overall pitch acceleration.

Since the filter performs a parameter estimation for the wind and the main propeller effective-

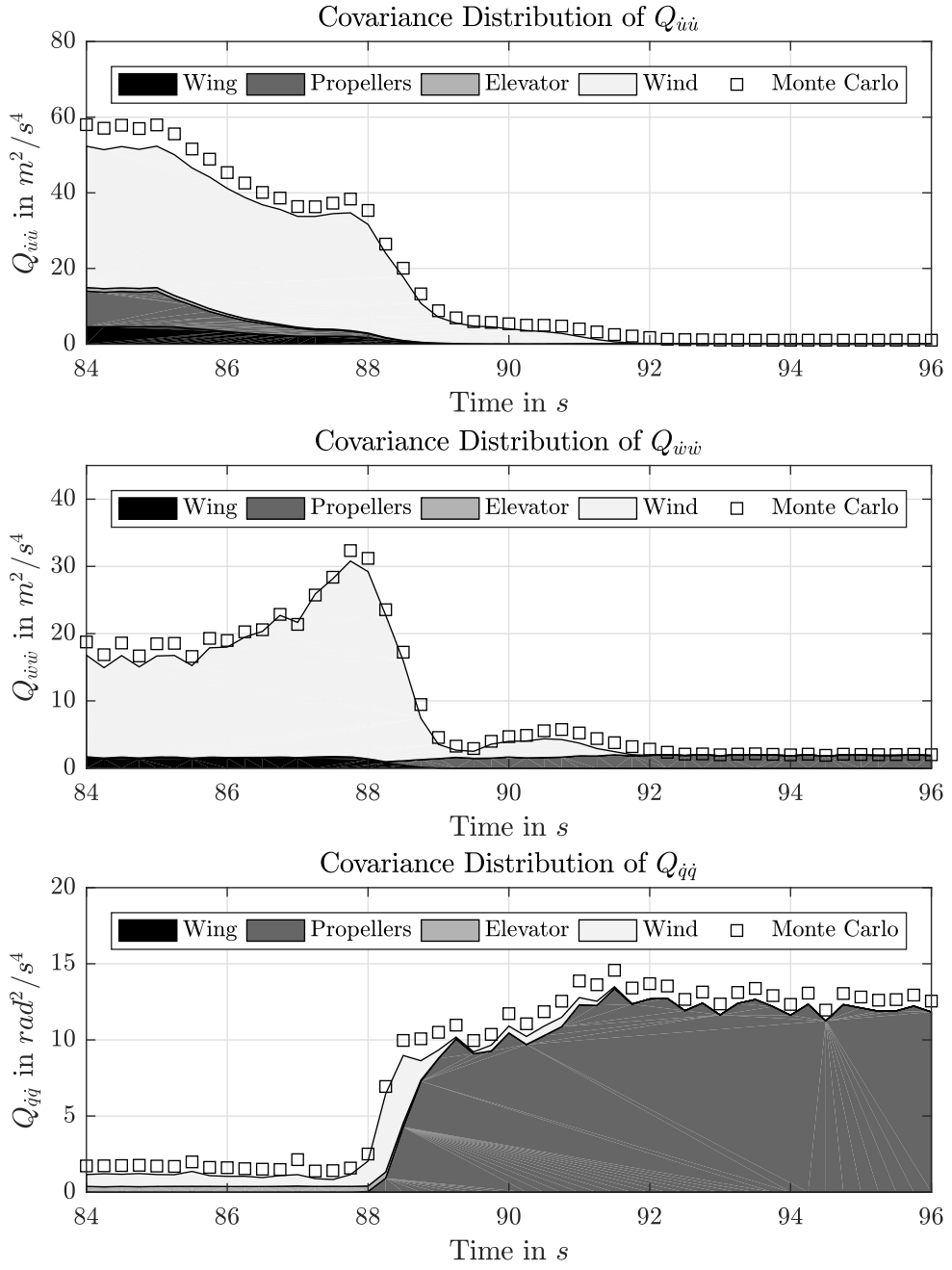

Figure 5.2: Distribution of parametric uncertainty for the state time derivatives during braking

ness the filter already considers the bias like uncertainties for these parameters. For this reason figure 5.3 and figure 5.4 show the parametric uncertainties without considering the effects of the estimated parameters. As the figures show, the main uncertainties lie within the combined zero drag of wing and body C_{D0}, the aerodynamic lift derivative of the wing $C_{L\alpha,wing}$, the air density ρ and the effectiveness of the front propeller C_{T,P_1}. For the translational acceleration the results of the Monte Carlo analysis and the analytical sensitivity results are very similar. This means the linearization and simplification to only consider the uncertainty due to the four aforementioned parameters is reasonable. However, for the pitch channel there is a constant deviation between

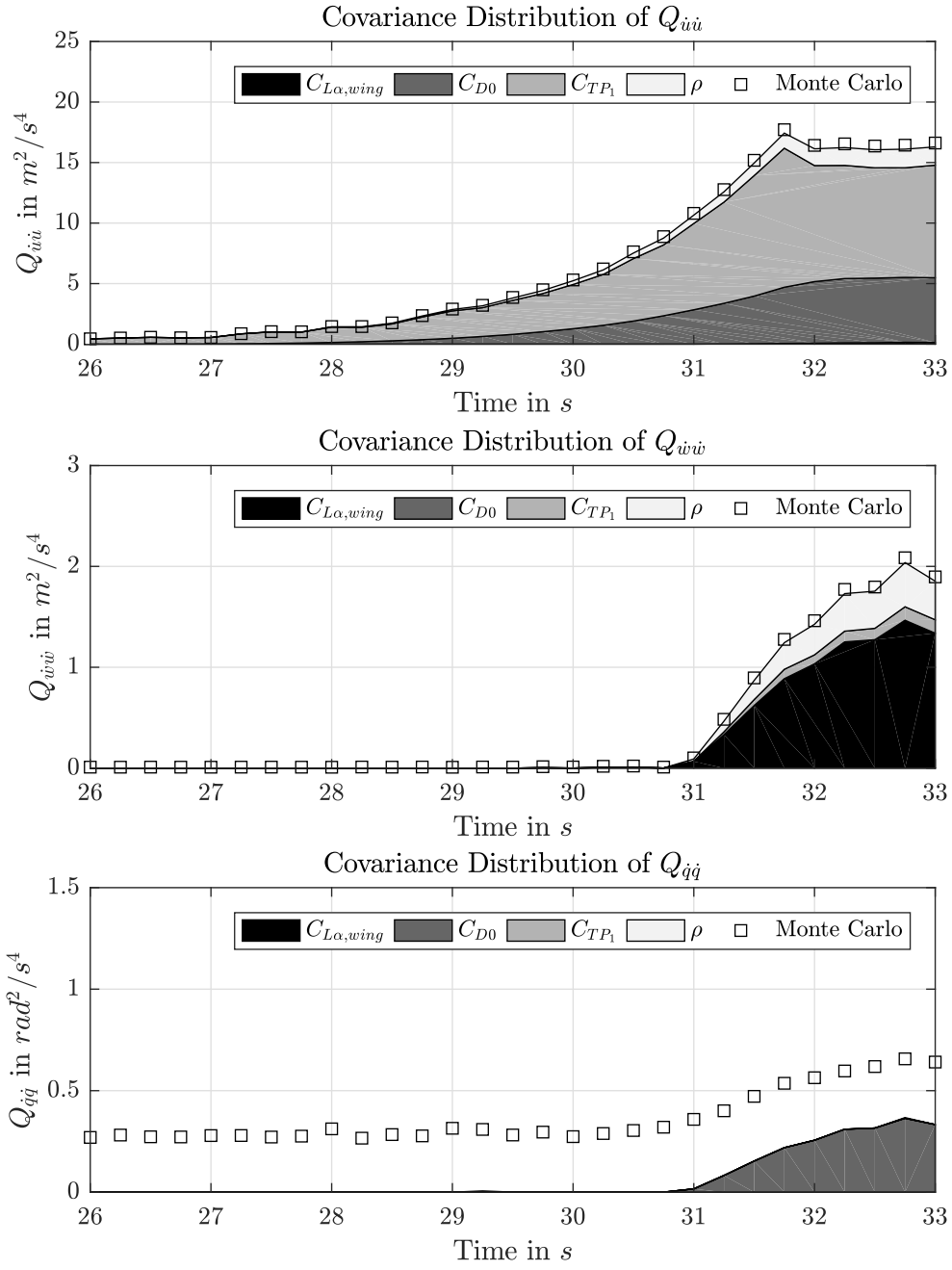

Figure 5.3: Distribution of parametric uncertainty for the state time derivatives during the transition. The uncertainty due to the estimated parameters (wind and effectiveness of the main propellers) is neglected here.

the full uncertainty Monte Carlo analysis and the simplified analytical uncertainty. The reason for this is the neglected uncertainty in the position of the center of gravity. Since this error seems to be constant and not state dependent anyway it can be enclosed in the tuning process of the constant process noise matrix $\boldsymbol{Q}_{xx}$.

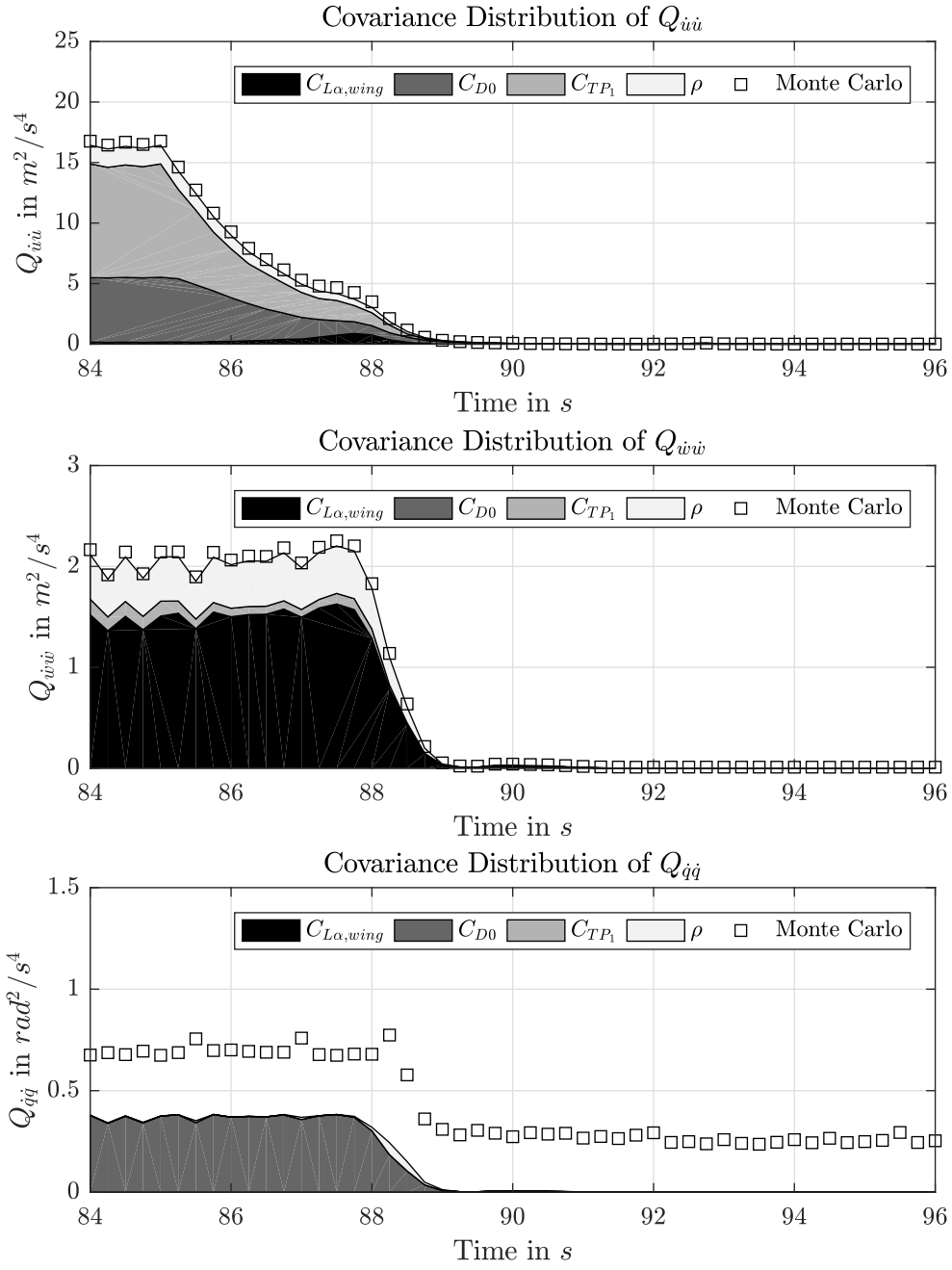

Figure 5.4: Distribution of parametric uncertainty for the state time derivatives during braking. The uncertainty due to the estimated parameters (wind and effectiveness of the main propellers) is neglected here.

With these results we can assemble the varying process noise as:

$$\boldsymbol{Q}_{xx,varying} = \mu_{15}\boldsymbol{J}_{varying}\boldsymbol{\Lambda}\boldsymbol{J}_{varying}^{T} \tag{5.39}$$

with the tuning parameter μ_{15} and the matrices

$$\boldsymbol{J}_{varying} = \begin{bmatrix} \frac{\delta \dot{u}}{\delta C_{L\alpha}} & \frac{\delta \dot{u}}{\delta C_{D0}} & \frac{\delta \dot{u}}{\delta C_{T,P_1}} & \frac{\delta \dot{u}}{\delta \rho} \\ \frac{\delta \dot{w}}{\delta C_{L\alpha}} & \frac{\delta \dot{w}}{\delta C_{D0}} & \frac{\delta \dot{w}}{\delta C_{T,P_1}} & \frac{\delta \dot{w}}{\delta \rho} \\ 0 & 0 & 0 & 0 \\ \frac{\delta \dot{q}}{\delta C_{L\alpha}} & \frac{\delta \dot{q}}{\delta C_{D0}} & \frac{\delta \dot{q}}{\delta C_{T,P_1}} & \frac{\delta \dot{q}}{\delta \rho} \end{bmatrix}$$
$$\boldsymbol{\Lambda} = \begin{bmatrix} \mathrm{cov}(C_{L\alpha}) & 0 & 0 & 0 \\ 0 & \mathrm{cov}(C_{D0}) & 0 & 0 \\ 0 & 0 & \mathrm{cov}(C_{T,P_1}) & 0 \\ 0 & 0 & 0 & \mathrm{cov}(\rho) \end{bmatrix} \tag{5.40}$$

5.4 Filter Switches and Logics

The switches for the filter and the different parameter channels are implemented within the propagation. When the filter is set to inactive, the propagation covariances are set to very high values which leads to a feedthrough of the measurements in the correction. The propagated covariances for the inactive filter are:

$$\boldsymbol{P}_{k+1|k}^{xx,inactive} = \begin{bmatrix} 10^9 & 0 & 0 & 0 \\ 0 & 10^9 & 0 & 0 \\ 0 & 0 & 10^9 & 0 \\ 0 & 0 & 0 & 10^9 \end{bmatrix}$$
$$\boldsymbol{P}_{k+1|k}^{pp,inactive} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.41}$$
$$\boldsymbol{P}_{k+1|k}^{xp,inactive} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The propagated covariance for the states is chosen high in order to almost exlusively use the measurements in order to determine the estimated states. When the filter is inactive, the parameter deviations are set to zero. With the assumption of the standard deviations in table 4.2 and the corresponding normalization of the estimated parameters, the expected parameter covariance in the case of an inactive filter equals one. The estimated cross coavariance between states and parameters is set to zero when the filter is inactive. When the cross covariance equals zero, the measurement update will not allocate any correction into the parameters.

Let s denote the switch for the total fitler and $s_{p_1}...s_{p_4}$ denote the switches for the indivudual

parameter channels. The propagation of the covariances including the filter switches then follows

$$\begin{aligned}
\boldsymbol{P}_{k+1|k}^{xx,switched} &= s\boldsymbol{P}_{k+1|k}^{xx} + (1-s)\boldsymbol{P}_{k+1|k}^{xx,inactive} \\
\boldsymbol{P}_{k+1|k}^{pp,switched} &= \boldsymbol{S}^{pp} \odot \boldsymbol{P}_{k+1|k}^{pp} + \left(\mathbf{1}^{4\times4} - \boldsymbol{S}^{pp}\right) \odot \boldsymbol{P}_{k+1|k}^{pp,inactive} \\
\boldsymbol{P}_{k+1|k}^{xp,switched} &= \boldsymbol{S}^{xp} \odot \boldsymbol{P}_{k+1|k}^{xp} + \left(\mathbf{1}^{4\times4} - \boldsymbol{S}^{xp}\right) \odot \boldsymbol{P}_{k+1|k}^{xp,inactive}
\end{aligned} \tag{5.42}$$

where $\boldsymbol{P}_{k+1|k}^{xx,switched}$ denotes the propagated covariance for the correction, the matrices $\boldsymbol{S}$ switching matrices and $\mathbf{1}^{4\times4}$ a matrix with all elements equal to one. The switching matrices follow

$$\begin{aligned}
\boldsymbol{S}^{pp} &= s \begin{bmatrix} s_{p_1} & s_{p_1}s_{p_2} & s_{p_1}s_{p_3} & s_{p_1}s_{p_4} \\ s_{p_1}s_{p_2} & s_{p_2} & s_{p_2}s_{p_3} & s_{p_2}s_{p_4} \\ s_{p_1}s_{p_3} & s_{p_2}s_{p_3} & s_{p_3} & s_{p_3}s_{p_4} \\ s_{p_1}s_{p_4} & s_{p_2}s_{p_4} & s_{p_3}s_{p_4} & s_{p_4} \end{bmatrix} \\
\boldsymbol{S}^{xp} &= s \begin{bmatrix} s_{p_1} & s_{p_2} & s_{p_3} & s_{p_4} \\ s_{p_1} & s_{p_2} & s_{p_3} & s_{p_4} \\ s_{p_1} & s_{p_2} & s_{p_3} & s_{p_4} \\ s_{p_1} & s_{p_2} & s_{p_3} & s_{p_4} \end{bmatrix}
\end{aligned} \tag{5.43}$$

As long as all switches are one, which means the filter is active, equation (5.42) just uses the normal propagated covariance. If the fitler is turned off completely, meaning that $s = 0$ all switching matrices are zero and only the inactive covariances will be used in the correction step of the filter. If only single parameter channels are turned off, the switching matrices in equation (5.43) only set the corresponding rows and colums of the covariances to the inactive values. As mentioned above, for the inactive case the propagated parameters equal zero:

$$\boldsymbol{p}_{k+1|k} = \begin{bmatrix} s_{p_1} \\ s_{p_2} \\ s_{p_3} \\ s_{p_4} \end{bmatrix} \odot \boldsymbol{p}_{k|k} \tag{5.44}$$

Equations (5.42) until (5.44) are designed in a way that they do not require discrete on and off values for the switches. Any values between zero and one can be applied. Intermediate values will lead to a blending between the inactive and the normal propagated covariance. The hybrid parameter estimation takes advantage of the latter blending. In order to prevent hard switches for the parameter estimation, the already existing wingborne blending λ will be used. This leads to the following choice for the parameter switches:

$$\begin{aligned}
s_{p_1} &= \lambda \\
s_{p_2} &= \lambda \\
s_{p_3} &= 1-\lambda \\
s_{p_4} &= 1-\lambda
\end{aligned} \tag{5.45}$$

5.5 Filter Initialization

For the filter initialization the same covariances are used as for the inactive case

$$\begin{aligned}
\boldsymbol{P}_{0|0}^{xx} &= \begin{bmatrix} 10^9 & 0 & 0 & 0 \\ 0 & 10^9 & 0 & 0 \\ 0 & 0 & 10^9 & 0 \\ 0 & 0 & 0 & 10^9 \end{bmatrix} \\
\boldsymbol{P}_{0|0}^{pp} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\boldsymbol{P}_{0|0}^{xp} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned} \tag{5.46}$$

The intial states and parameter estimates are zero

$$\begin{aligned}
\boldsymbol{x}_{0|0} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\boldsymbol{p}_{0|0} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{aligned} \tag{5.47}$$

For takeoff and landing the filter has to be set to inactive from the outside. As soon as the aircraft has enough altitude to guarantee a groundcontact free flight, the filter can be turned on with the filter switch s.

5.6 Filter Diagnostics

In order to verify the filter performance online, simple comparisons are used. First of all the deviation of measured states and estimated states can be compared to a certain threshold. In the shape of this thesis this threshold is set to ten times the standard deviation of the white noise of the corresponding sensor/INS models. This leads to the flags of valid state estimates $\boldsymbol{v}_x$:

$$\boldsymbol{v}_{\hat{x}} = \left| \boldsymbol{x}_{meas,k} - \boldsymbol{x}_{k|k} \right| \leq 10 \begin{bmatrix} \sigma_{WN,u} \\ \sigma_{WN,w} \\ \sigma_{WN,\theta} \\ \sigma_{WN,q} \end{bmatrix} \tag{5.48}$$

For the parameters an estimation of a parameter deviation that is higher than three times the standard deviation of the parameter from table 4.2 is considered invalid:

$$\boldsymbol{v}_p = \left|p_{k|k}\right| \leq \begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix} \tag{5.49}$$

The covariance of the parameter estimation is considered valid as long as the corresponding standarddeviation of the diagonal does not exceed three times the assumed value from table 4.2:

$$\boldsymbol{v}_{p_{cov}} = \left|diag(\boldsymbol{P}^{pp}_{k|k})\right| \leq \begin{bmatrix} 9 \\ 9 \\ 9 \\ 9 \end{bmatrix} \tag{5.50}$$

In the shape of this thesis the filter only outputs the according diagnostic flags. There are no actions for the defined flags considered in this thesis.

5.7 Reducing Computational Cost

This sections main focus are two approaches to reduce the computational effort to perform the estimation. The first method to reduce computational cost is a more efficient numeric correction, the sequential correction. The second method neglects a part in the covariance propagation that helps reducing the number of OBPM evaluations.

5.7.1 Sequential Correction

As mentioned in section 2.1.2, the sequential correction only works for diagonally shaped or constant measurement noise covariances $\boldsymbol{R}$. Since both conditions hold for the used measurement covariance the Kalman filter can apply a sequential correction. In the first step of the sequential correction all states and covariances are set to the propagated values:

$$\begin{aligned} \boldsymbol{x}_{0,k+1} &= \boldsymbol{x}_{k|k+1} \\ \boldsymbol{p}_{0,k+1} &= \boldsymbol{p}_{k|k+1} \\ \boldsymbol{P}^{xx}_{0,k+1} &= \boldsymbol{P}^{xx}_{k|k+1} \\ \boldsymbol{P}^{xp}_{0,k+1} &= \boldsymbol{P}^{xp}_{k|k+1} \\ \boldsymbol{P}^{pp}_{0,k+1} &= \boldsymbol{P}^{pp}_{k|k+1} \end{aligned} \tag{5.51}$$

Then the sequential correction for all for measurements $i = 1...4$ follows

$$\begin{aligned} \boldsymbol{x}_{i+1,k+1} &= \boldsymbol{x}_{i,k+1} + \boldsymbol{K}^{x}_{i,k+1}\boldsymbol{S}_i\left(\boldsymbol{x}_{meas} - \boldsymbol{x}_{i,k+1}\right) \\ \boldsymbol{p}_{i+1,k+1} &= \boldsymbol{p}_{i,k+1} + \boldsymbol{K}^{p}_{i,k+1}\boldsymbol{S}_i\left(\boldsymbol{x}_{meas} - \boldsymbol{x}_{i,k+1}\right) \end{aligned} \tag{5.52}$$

with the selection vectors $\boldsymbol{S}_i$

$$\begin{aligned} S_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \\ S_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \\ S_3 &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \\ S_4 &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5.53}$$

The Kalman gain is

$$\begin{bmatrix} \boldsymbol{K}^x_{i,k+1} \\ \boldsymbol{K}^p_{i,k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}^{xx}_{i,k+1} \\ \boldsymbol{P}^{px}_{i,k+1} \end{bmatrix} \boldsymbol{S}_i^T \left[\boldsymbol{S}_i \boldsymbol{P}^{xx}_{i,k+1} \boldsymbol{S}_i^T + \boldsymbol{S}_i \boldsymbol{R} \boldsymbol{S}_i^T \right]^{-1} \tag{5.54}$$

The covariance update laws follow

$$\begin{aligned} \boldsymbol{P}^{xx}_{i+1,k+1} &= \left(\boldsymbol{I} - \boldsymbol{K}^x_{i,k+1} \boldsymbol{S}_i \right) \boldsymbol{P}^{xx}_{i,k+1} \\ \boldsymbol{P}^{xp}_{i+1,k+1} &= \left(\boldsymbol{I} - \boldsymbol{K}^x_{i,k+1} \boldsymbol{S}_i \right) \boldsymbol{P}^{xp}_{i,k+1} \\ \boldsymbol{P}^{pp}_{i+1,k+1} &= \boldsymbol{P}^p_{i,k+1} p - \boldsymbol{K}^p_{i,k+1} \boldsymbol{S}_i \boldsymbol{P}^{xp}_{i,k+1} \end{aligned} \tag{5.55}$$

In the end the corrected states and covariances are overwritten

$$\begin{aligned} \boldsymbol{x}_{k+1|k+1} &= \boldsymbol{x}_{4,k+1} \\ \boldsymbol{p}_{k+1|k+1} &= \boldsymbol{p}_{4,k+1} \\ \boldsymbol{P}^{xx}_{k+1|k+1} &= \boldsymbol{P}^{xx}_{4,k+1} \\ \boldsymbol{P}^{xp}_{k+1|k+1} &= \boldsymbol{P}^{xp}_{4,k+1} \\ \boldsymbol{P}^{pp}_{k+1|k+1} &= \boldsymbol{P}^{pp}_{4,k+1} \end{aligned} \tag{5.56}$$

The sequential correction reduces the effort of the required inversion for the correction. As the measurement matrix is diagonally shaped this approach is very efficient compared to the original full matrix inversion.

5.7.2 Neglecting State Covariance Propagation

Unlike the sequential correction thhe idea in this subsection is a more radical intervention in the Kalman filter. Since the white noise in the measurements of the vehicle states is quite small, the state covariance will be quite small as well. For this reason the propagated uncertainty due to the effect of uncertain states on the system dynamics will be also small compared to the parametric uncertainties and overall process noise. The idea now is to neglect the porpagated state uncertainty which passed through the system dynamics. The way this is done is to set the derivative of the system dynamics with respect to the states to zero. This approach cancels the need to evaluate the gradients of the states with the onboard plant model. Consequently it reduces the number of necesserey OBPM evaluations by three, since the pitch angle still has to be perturbed as it is a virtual control. Performing the neglection in the propagation equations (see section) first of all leads to

$$\boldsymbol{\phi}^x = \boldsymbol{I} \tag{5.57}$$

This leads to simplifications in the equations for the covariances including the states:

$$\begin{aligned}\boldsymbol{P}^{xp}_{k|k-1} &= \boldsymbol{P}^{xp}_{k-1|k-1} + \boldsymbol{\phi}^p \boldsymbol{P}^{pp}_{k-1|k-1} + \boldsymbol{Q}^{xp}_k \\ \boldsymbol{P}^{xx}_{k|k-1} &= \boldsymbol{P}^{xx}_{k-1|k-1} + \boldsymbol{\phi}^p \boldsymbol{P}^{px}_{k-1|k-1} + \left(\boldsymbol{P}^{xp}_{k-1|k-1} + \boldsymbol{\phi}^p \boldsymbol{P}^{pp}_{k-1|k-1}\right) \boldsymbol{\phi}^p T + \boldsymbol{Q}^{xx}_k\end{aligned} \tag{5.58}$$

Chapter 6

Results

6.1 Nominal Performance

6.2 Robustness

6.2.1 Parametric Uncertainty Robustness

Linear Parametric Uncertainty Robustness

Nonlinear Parametric Uncertainty Robustness

6.2.2 Input Robustness

Linear Input Robustness

Nonlinear Input Robustness

6.2.3 Output Robustness

Linear Output Robustness

Nonlinear Output Robustness

6.3 Covariance Consistency

6.4 Computational Cost Reduction

Chapter 7

Resumee

Chapter 8

Future Works

Appendix

Linearization Matrices

The closed loop input matrix of the linearization follows:

$$\boldsymbol{B}_{cl} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\left[\boldsymbol{A}_{ref}\boldsymbol{C}_y^T - \left(\boldsymbol{A}_{ref}\boldsymbol{C}_{\mathrm{v}}^T\boldsymbol{S}_{\mathrm{v}} + \boldsymbol{C}_\gamma^T\tilde{\boldsymbol{B}}_\gamma\right)\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\right] \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -\left[\boldsymbol{\Omega}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\right] \\ -\left[\hat{\boldsymbol{\Omega}}_{act}\boldsymbol{S}_u\boldsymbol{T}^{-1}\boldsymbol{C}_\gamma\boldsymbol{A}_{ref}\boldsymbol{C}_y^T\right] \end{bmatrix} \tag{8.1}$$

The closed loop outptut matrix (the outputs are the pilot controlled velocities) of the linearization follows:

$$\boldsymbol{C}_{cl} = \left[\boldsymbol{C}_y\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\right] \tag{8.2}$$

Sensitivity Analysis

Let $J_{\dot{u},C_{T,P_1}}$ denote the sensitivity of $\left(\dot{u}_K^G\right)_O^{EO}$ with respect to the parameter C_{TP_1}. First of all note that all entries in the row of $\dot{\theta}$ are zero, since the strapdown equation does not depend on any parameters.

The Jacobian entries for the pitch rate derivative row follow:

$$
\begin{aligned}
J_{\dot{q},C_{T,P_2}} &= \frac{1}{I_{yy}^R}\left(r_x^{RP_2}\right)_B \omega_2^2 \\
J_{\dot{q},C_{T,P_3}} &= \frac{1}{I_{yy}^R}\left(r_x^{RP_3}\right)_B \omega_3^2 \\
J_{\dot{q},\rho} &= \frac{\dot{q}}{\rho} \\
J_{\dot{q},u_w} &= -\frac{\rho}{\bar{q}}\dot{q}\left(u_A^G\right)_O^E + \frac{\left(r_x^{RE}\right)_B}{I_{yy}^R}\frac{\bar{q}}{\|v_A\|}S_{elev}\Big[+C_{L\alpha,elev}\lambda_{C_L}\cos\alpha_A \\
&\quad + 2C_{L\alpha,elev}^2\lambda_{C_L}^2 k_{elev}\left(\alpha_A+\eta\right)\sin\alpha_A \\
&\quad - \sin\alpha_A C_{L,elev} + \cos\alpha_A C_{D,elev}\Big]\sin\gamma_A \\
J_{\dot{q},w_w} &= -\frac{\rho}{\bar{q}}\dot{q}\left(w_A^G\right)_O^E + \frac{\left(r_x^{RE}\right)_B}{I_{yy}^R}\frac{\bar{q}}{\|v_A\|}S_{elev}\Big[-C_{L\alpha,elev}\lambda_{C_L}\cos\alpha_A \\
&\quad - 2C_{L\alpha,elev}^2\lambda_{C_L}^2 k_{elev}\left(\alpha_A+\eta\right)\sin\alpha_A \\
&\quad + \sin\alpha_A C_{L,elev} - \cos\alpha_A C_{D,elev}\Big]\cos\gamma_A \\
J_{\dot{q},C_{L\alpha,elev}} &= \frac{\left(r_x^{RE}\right)_B}{I_{yy}^R}\bar{q}S_{elev}\Big[\cos\alpha_A\lambda_{C_L} + 2\sin\alpha_A k_{elev}C_{L\alpha,elev}\lambda_{C_L}^2\left(\alpha_A+\eta\right)\Big]\left(\alpha_A+\eta\right) \\
J_{\dot{q},C_{D0,elev}} &= \frac{\left(r_x^{RE}\right)_B}{I_{yy}^R}\bar{q}S_{elev}\sin\alpha_A \\
J_{\dot{q},k_{elev}} &= \frac{\left(r_x^{RE}\right)_B}{I_{yy}^R}\bar{q}S_{elev}\sin\alpha_A C_{L\alpha,elev}^2\lambda_{C_L}^2\left(\alpha_A+\eta\right)^2
\end{aligned}
\tag{8.3}
$$

All other sensitivies of the derivative of the pitch rate are zero.

$$
\begin{aligned}
J_{\dot{u},C_{T,P_1}} &= \frac{1}{m}\cos\theta\omega_2^2 \\
J_{\dot{u},C_{T,P_2}} &= -\frac{1}{m}\sin\theta\omega_2^2 \\
J_{\dot{u},C_{T,P_3}} &= -\frac{1}{m}\sin\theta\omega_3^2 \\
J_{\dot{u},\rho} &= \frac{\dot{u}}{\rho} \\
J_{\dot{u},u_w} &= -\frac{\rho}{\bar{q}}\dot{u}\left(u_A^G\right)_O^E + \frac{1}{m}\frac{\bar{q}}{\|v_A\|}S_{wing}\Big[-C_{L\alpha,wing}\lambda_{C_L}\sin\gamma_A \\
&\quad - 2C_{L\alpha,wing}^2\lambda_{C_L}^2 k_{wing}\alpha_A\cos\gamma_A \\
&\quad + \cos\gamma_A C_{L,wing} - \sin\gamma_A C_{D,wing}\Big]\sin\gamma_A \\
J_{\dot{u},w_w} &= -\frac{\rho}{\bar{q}}\dot{u}\left(w_A^G\right)_O^E - \frac{1}{m}\frac{\bar{q}}{\|v_A\|}S_{wing}\Big[C_{L\alpha,wing}\lambda_{C_L}\sin\gamma_A \\
&\quad + 2C_{L\gamma,wing}^2\lambda_{C_L}^2 k_{wing}\alpha_A\cos\gamma_A \\
&\quad - \cos\gamma_A C_{L,wing} + \sin\gamma_A C_{D,wing}\Big]\cos\gamma_A \\
J_{\dot{u},C_{L\alpha,wing}} &= -\frac{1}{m}\bar{q}S_{wing}\Big[\sin\gamma_A\lambda_{C_L} + 2\cos\gamma_A k_{wing}C_{L\alpha,wing}\lambda_{C_L}^2\alpha_A\Big]\alpha_A \\
J_{\dot{u},C_{D0,wing}} &= -\frac{1}{m}\bar{q}S_{wing}\cos\gamma_A \\
J_{\dot{u},k_{wing}} &= -\frac{1}{m}\bar{q}S_{wing}\cos\gamma_A C_{L\alpha,wing}^2\lambda_{C_L}^2\alpha_A^2
\end{aligned}
\tag{8.4}
$$

$$
\begin{aligned}
J_{\dot{w},C_{T,P_1}} &= -\frac{1}{m}\sin\theta\omega_2^2 \\
J_{\dot{w},C_{T,P_2}} &= -\frac{1}{m}\cos\theta\omega_2^2 \\
J_{\dot{w},C_{T,P_3}} &= -\frac{1}{m}\cos\theta\omega_3^2 \\
J_{\dot{w},\rho} &= \frac{\dot{w}}{\rho} \\
J_{\dot{w},u_w} &= -\frac{\rho}{\bar{q}}\dot{w}\left(u_A^G\right)_O^E + \frac{1}{m}\frac{\bar{q}}{\|v_A\|}S_{wing}\Big[-C_{L\alpha,wing}\lambda_{C_L}\cos\gamma_A \\
&\quad + 2C_{L\alpha,wing}^2\lambda_{C_L}^2 k_{wing}\alpha_A\sin\gamma_A \\
&\quad - \sin\gamma_A C_{L,wing} - \cos\gamma_A C_{D,wing}\Big]\sin\gamma_A \\
J_{\dot{w},w_w} &= -\frac{\rho}{\bar{q}}\dot{w}\left(w_A^G\right)_O^E + \frac{1}{m}\frac{\bar{q}}{\|v_A\|}S_{wing}\Big[C_{L\alpha,wing}\lambda_{C_L}\cos\gamma_A \\
&\quad - 2C_{L\gamma,wing}^2\lambda_{C_L}^2 k_{wing}\alpha_A\sin\gamma_A \\
&\quad + \sin\gamma_A C_{L,wing} + \cos\gamma_A C_{D,wing}\Big]\cos\gamma_A \\
J_{\dot{w},C_{L\alpha,wing}} &= \frac{1}{m}\bar{q}S_{wing}\Big[-\cos\gamma_A\lambda_{C_L} + 2\sin\gamma_A k_{wing}C_{L\alpha,wing}\lambda_{C_L}^2\alpha_A\Big]\alpha_A \\
J_{\dot{w},C_{D0,wing}} &= \frac{1}{m}\bar{q}S_{wing}\sin\gamma_A \\
J_{\dot{w},C_{k,wing}} &= \frac{1}{m}\bar{q}S_{wing}\sin\gamma_A C_{L\alpha,wing}^2\lambda_{C_L}^2\alpha_A^2
\end{aligned}
\tag{8.5}
$$