DEGREE PROJECT

Automated Controller Design for a Missile Using Convex Optimization

Christoph Auenmüller

Space Engineering, masters level
2016

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

LULEÅ UNIVERSITY OF TECHNOLOGY

Automated Controller Design for a Missile Using Convex Optimization

By

Christoph Auenmüller

Faculty of Computer Science
University of Würzburg
University of Luleå

A master thesis submitted to the University of Würzburg and University of Luleå in accordance with the requirements of the degree of Master of Science. Realized at MBDA Germany.

September 2016

Written under the supervision of: Prof. Dr.-Ing. Sergio Montenegro
Prof. Thomas Gustafsson
Dr.-Ing. Klaus Kefferpütz

Co-funded by the
Erasmus+ Programme
of the European Union

Affidavit

I hereby confirm that my thesis entitled "Automated Controller Design for a Missile Using Convex Optimization" is the result of my own work. I did not receive any help or support from commercial consultants. All sources are specified in the thesis.

Furthermore, I confirm that this thesis has not yet been submitted as part of another examination process neither in identical nor in similar form.

Place, Date

Signature

Abstract

The focus of the present master thesis is the automation of an existing controller design for a missile using two aerodynamic actuating systems. The motivation is to evaluate more missile concepts in a shorter period of time.
The option used is trimming and linearization of a highly nonlinear missile at specific conditions. According to these conditions, either a two-dimensional operating point grid defined by Mach number and height or three-dimensional operating point grid defined by Mach number, height and angle of attack is generated for the whole operating range of the missile. The controllers are designed at these points using convex optimization. The convex set defines the pole placement area which is constrained by linear matrix inequalities according to the dynamic behavior of the missile at the operating point conditions. These controllers describe a validity area where the missile can be stabilized. This area consists all neighboring operating points and defines therefore the grid density which can differ at specific regions of the operating range. Controlling the missile to the target makes it necessary to apply gain-scheduling in order to get the manipulated variable by interpolation of adjacent operating points. During this blending of the controllers a problem called windup can occur when an actuator is saturated. This might lead to instability in worst case but can be counteracted by a model-recovery anti-windup network which guarantees stability in the presence of saturation. This anti-windup design is automated by an affine linear parameter dependency of the grid parameters and has the same validity area like the controllers.
The whole design was successfully developed and tested in MATLAB/Simulink on missiles using one or two aerodynamic actuating systems. The controllers have a good performance at small and high acceleration steps and the anti-windup keeps the missile stable even though the actuators are saturated. Stability and robustness of the controllers and anti-windup networks was verified as well as an air defense maneuver where the missile starts at the ground and intercepts a target at high altitude was successfully simulated for different grids and missiles.

Contents

List of Figures

List of Tables

Nomenclature

Greek Symbols

Symbol	Description	Value / Unit
α	Angle of Attack	rad
β, η, α	Anti-Windup Design Parameters	–
Δ	Local Area of an Operating Point	–
δ	Imaginary to Real Ratio, Difference between necessary and realizable command	–
η	Deflection of Elevator	rad
γ	Minimization Parameter of the H_∞ norm	–
$\underline{\lambda}$	Lower Limitation of Pole Placement Area	–
ξ_{ms}	Mismatch States	–
ω	Rotation Rate	$\mathrm{rad\,s^{-1}}$
Θ	Pitch Angle	rad
ρ	Air Density	$\mathrm{kg\,m^{-3}}$
ϕ	Roll Angle	rad
β	Sideslip Angle	rad
θ	Total Aerodynamic Angle of Attack	rad
φ	Total Aerodynamic Roll Angle	rad
$\overline{\lambda}$	Upper Limitation of Pole Placement Area	–
ξ	Deflection of Aileron	rad
Ψ	Yaw Angle	rad
ζ	Deflection of Rudder	rad

Roman Symbols

Symbol	Description	Value / Unit

$\boldsymbol{A}$	System Matrix	–
$\boldsymbol{B}$	Input Matrix	–
$\boldsymbol{C}$	Output Matrix	–
$\boldsymbol{D}$	Feedthrough Matrix	–
d	Affiliation Parameter	–
$\overline{q}$	Dynamic Pressure	$\mathrm{N\,m^{-2}}$
g	Earth Acceleration	$9.81\,\mathrm{m\,s^{-2}}$
$\boldsymbol{F}$	Scaled Force	F
$\boldsymbol{G}(s)$	Transfer Function	–
H	Height	m
$\mathbb{I}$	Identity Matrix	–
I_{xx}	Moment of Inertia about x-Axis	$\mathrm{kg\,m^2}$
I_{yy}	Moment of Inertia about y-Axis	$\mathrm{kg\,m^2}$
I_{zz}	Moment of Inertia about z-Axis	$\mathrm{kg\,m^2}$
$\boldsymbol{k}$	Control Gain	–
$\boldsymbol{K}_{aw}$	Anti-Windup Gain	–
V	Kinematic Velocity	$\mathrm{m\,s^{-1}}$
$\boldsymbol{K}(s)$	Control Gain in State Space Representation	–
$\boldsymbol{F}(x)$	Linear Matrix Inequalities	–
$\mathcal{G}$	Lyapunov Region	–
M	Mach Number	–
m	Mass	kg
$\boldsymbol{M}$	Moments	$\mathrm{N\,m}$
J	Output Energy	J
$\boldsymbol{P}$	Positive Definite Matrix	–
q	Pitch Rate	$\mathrm{rad\,s^{-1}}$
$\boldsymbol{Q}$	Positive Definite Matrix	–

$\boldsymbol{R}$	Psitive Definite Symmetric Matrix, Objective Function	–
S	Reference Area	m^2
l	Reference Length	m
p	Roll Rate	$\mathrm{rad\,s^{-1}}$
$\boldsymbol{x}_{aw}$	Anti-Windup Network States	–
$\boldsymbol{x}_r$	Controller States	–
$\boldsymbol{x}_s$	Plant States	–
t	Scaled Time	t
$\boldsymbol{T}_{Act}$	Time Constants Actuators	$(\mathrm{mt})^{-1}$
$\boldsymbol{T}$	Transformation Matrix	–
$\boldsymbol{u}$	Manipulated Variable	–
u	Velocity in Body-Fixed System in x-direction	$\mathrm{m\,s^{-1}}$
v	Velocity in Body-Fixed System in y-direction	$\mathrm{m\,s^{-1}}$
w	Velocity in Body-Fixed System in z-direction	$\mathrm{m\,s^{-1}}$
x_{cg}	Center of Gravity	mm
x_{cp}	Pressure Point	mm
$\boldsymbol{y}$	Output Vector	–
y_a	Scaled Acceleration in y-Direction	acc
r	Yaw Rate	$\mathrm{rad\,s^{-1}}$
z_a	Scaled Acceleration in z-Direction	acc

List of Abbreviations

MRAW Model Recovery Anti-Windup 25
AW Anti-Windup 2
NW Network 2
DOF Degrees of Freedom 5
MATLAB MATrix LABoratory 1
DACS Divert Attitude Control System 1
LMI Linear Matrix Inequality VII
ROA Region Of Attraction 61
OP Operating Point 2
EoM Equations of Motion 33
NED North-East-Down 6
BF Body-Fixed 5
PPA Pole Placement Area 33
IPM Interior-Point Method 16
GEVP Generalized Eigenvalue Problem 19
SISO Single Input Single Output 19
LTI Linear Time Invariant 19
PID Proportional Integral Differential IX
FDLTI Finite-Dimensional Linear Time Invariant 58
NDI Nonlinear Dynamic Inversion 1
TF Transfer Function 39
GP Grid Parameter 52
GB Grid Bound 45
GD Grid Dimension 45

1 Introduction

Continuously improvements in the development of warfare material all over the world makes it necessary to counteract this constant threat. MBDA is specialized in the development of missile systems, especially air defense missiles, to provide safety. Such a missile has to fulfill diverse design criteria for the electrical power supply, attitude determination and control, propulsion, structure, command and data handling, thermal control and payload. For the design of the whole missile system, various simulations have to be executed in advance where various designs of the missile can be evaluated. Therefore, a simulation for missile systems with its essential parts like radar, communication and launcher was developed. An autopilot is needed in all simulations of the whole missile system. One essential module for the guidance and hence performance is the flight control system. This system is necessary for the automated control and navigation of missiles. This master thesis deals with the development and improvement of flight control algorithms in terms of an automated controller design using convex optimization in order to evaluate more missiles in a shorter period of time.

1.1 Problem Description

For the preliminary design of the missile's guidance, a program in MATrix LABoratory (MATLAB) was developed. In order to simulate the designed control system to offer valuable clues of the performance, a simulation in Simulink connected via S-Functions to the mentioned missile system simulation, developed in C++, exists as well.
The implemented control method is able to use only measured values and is applicable to actuator systems like ailerons, elevators, rudders or Divert Attitude Control System (DACS) in order to control a missile optimally over the whole flight envelope. To evaluate different concepts in a short period of time, the design of an autopilot should be automated by using the maximum possible number of measurable states and providing robustness. Since the development of this control method will be mainly formulated in a general manner, it can be also applied to small launch vehicles. Since the varying air density in the lower and higher atmosphere influence the control effectiveness, the automatically designed controllers have to guarantee a stabilized flight behavior of missiles or small rockets over the whole envelope.
The automated design is a challenging task since the missile is highly nonlinear. One possibility to deal with the nonlinearity is an approach by using a Nonlinear Dynamic Inversion (NDI) controller. Such a control method has some disadvantages according to an automated design. A NDI controller needs to know the whole state of the missile which requires the design of a state estimator/observer.

Employing a NDI design to a tail controlled missile makes it necessary to account for the non-minimum phase property of the system [25]. This is done by defining a new controlled variable which is a complex task. Another possibility is the linearization of the missile in various operating ranges. The advantage in the sense of automation is that automated tools for the design of linear systems exist. Additionally, the knowledge of the whole system state is not necessary due to the availability of output feedback controllers. A controller has a specific validity area in the operating range which leads to the necessity of several controllers for the whole operating range of the missile. Using a suitable method for blending among the controllers makes it possible to control the missile during its mission. Subsequent, the main problems of the thesis accounted in [3] will be briefly described:

1.1.1 Automated Adaption of the Operating Point Grid

A nonlinear system like a missile has to be linearized at specific operating points also called trim points in order to apply for example well known controller design- and analysis methods for linear systems to nonlinear systems. The current grid is divided in height and Mach number which influence the control effectiveness and will be enhanced by the possibility of linearizing additionally over the angle of attack. If the grid is too coarse, the approximation of the nonlinear flight dynamics will not be sufficiently precise. Degradation of the performance and even instability might occur. Linear matrix inequalities allow for an automated evaluation of the grid. A negative result would lead to a denser grid, avoiding the mentioned problems. Detailed information regarding this problem are given in section 2.2.1.

1.1.2 Controller Design

When the system is linearized at specific Operating Point (OP)s, controllers at these points have to be designed to ensure a stable behavior of the system during the flight. These controllers are applied to the diverse actuating systems and have to provide different quality criteria. Convex optimization using LMIs offers the possibility of an automated design of the controllers ensuring the controllers to fulfill predefined requirements. This automated design is discussed in more detail in section 2.3 and later in 3.3.

1.1.3 Design of an Anti-Windup Network

The controller of the plant should be highly dynamic but is constrained by limits of the actuator. When the necessary manipulated variable is higher than actually physically realizable, instability can occur. A possibility to counteract such problems is called anti-windup. An automated design of an Anti-Windup (AW) Network (NW) can be developed in order to stabilize a system during saturation of an actuator. We will also discuss the fact, that blending between controllers requires windup countermeasures. A more detailed description of anti-windup will be presented in section 2.4.

1.1.4 Implementation and Design Aspects

Currently the implementation is constrained by a grid consisting of 2x3 grid points. In order to simulate the whole envelope, adjacent operating points have to be determined instantaneous online while retrieving the particular control algorithm. The implementation should be independent of the dimension of the grid, hence the user should be able to define the dimension, their variables and bounds.
The parameters should be set with respect to the current flight conditions in order to have stable flight behavior in lower and also higher atmospheric layers. Therefore, the ability of lateral acceleration has to be investigated.

The algorithms, controller design and anti-windup control will be developed in MATLAB/Simulink.

1.2 Presupposed Knowledge

This master thesis presupposes a fundamental knowledge in control theory as imparted by the book [12] or similar books which impart basics in control. Furthermore, basic knowledge in aerodynamics as well as flight mechanics are recommended for understanding this thesis.

1.3 Outline

This work is organized as in the following described.
In the beginning, background theory for the control of missiles is presented. The existing missile simulation, linearization methods as well as the prove of stability according to Lyapunov [28] will be briefly described. Furthermore, convex optimization using LMI constraints, the problem of windup and counteracting methods as well as possibilities of controllers for the stabilization of the missile will be addressed.
Afterwards, the realization of automating the AW NW development, grid generation, controller design and their implementation in the existing program in MATLAB/Simulink and link to the missile simulation will be shown.
Finally, the results will be presented and discussed and an overview of upcoming work and suggestions for improvements are given.

Remark 1. *Due to secrecy reasons on the part of MBDA Germany, the units of acceleration* $[\mathrm{m/s^2}] \cdot k_1 = [acc]$*, force* $[\mathrm{N}] \cdot k_2 = [F]$ *and the time* $[\mathrm{s}] \cdot k_3 = [t]$ *are distorted by scaling factors.*

2 Background Theory for the Control of Missiles

2.1 Missile Model

The control of a technical system requires a detailed mathematical copy of it. This can be realized by the related differential equations which describe its motion. Therefore, all exterior forces and moments which act on the missile are needed in order to get all equations of motion, representing the six Degrees of Freedom (DOF) behavior of the missile. The forces and moments are caused by gravity, aerodynamics and the missile's thrust.
Subsequently, the equations of motion and the different actuating systems will be presented and in more detail described according to [7], [18], [19].

2.1.1 Equations of Motion (EoM)

As mentioned above, the missile has six DOFs which will be described by three vector equations. More specific, the translation in the kinematic system, the rotational motion and the differential equations of the attitude.
The translational behavior can be expressed by

$$\dot{V} = \frac{1}{m} \cdot \left(F_{X,t} \cdot \cos\alpha \cdot \cos\beta + F_{Y,t} \cdot \cos\beta + F_{Z,t} \cdot \sin\alpha \cdot \cos\beta\right), \tag{2.1}$$

$$\dot{\alpha} = \frac{1}{V \cdot \cos\beta \cdot m} \cdot (F_{Z,t} \cdot \cos\alpha - F_{X,t} \cdot \sin\alpha) + q - \tan\beta \cdot (p \cdot \cos\alpha + r \cdot \sin\alpha), \tag{2.2}$$

$$\dot{\beta} = \frac{1}{V \cdot m} \cdot (F_{Y,t} \cdot \cos\beta - F_{X,t} \cdot \cos\alpha \cdot \sin\beta - F_{Z,t} \cdot \sin\alpha \cdot \sin\beta) + p \cdot \sin\alpha - r \cdot \cos\alpha, \tag{2.3}$$

where $\dot{V}$ describes the time derivative of the kinematic velocity, $\dot{\alpha}$ and $\dot{\beta}$ the time derivatives of the kinematic angle of attack and sideslip angle. Furthermore, m is the missile's mass and p, q, r the roll rate, pitch rate and yaw rate respectively. The three forces in x-,y- and z-direction ($F_{X,t}$, $F_{Y,t}$ and $F_{Z,t}$) resulting from the principle of linear momentum and describe the equilibrium of forces in the Body-Fixed (BF) frame (see figure 2.1)

$${}_B\vec{\boldsymbol{F}}_t = \sum {}_B\vec{\boldsymbol{F}} = \begin{bmatrix} F_{X,t} \\ F_{Y,t} \\ F_{Z,t} \end{bmatrix} = \begin{bmatrix} F_{X,A} + F_{X,p} + F_{X,g} \\ F_{Y,A} + F_{Y,DACS} + F_{Y,g} \\ F_{Z,A} + F_{Z,DACS} + F_{Z,g} \end{bmatrix}. \tag{2.4}$$

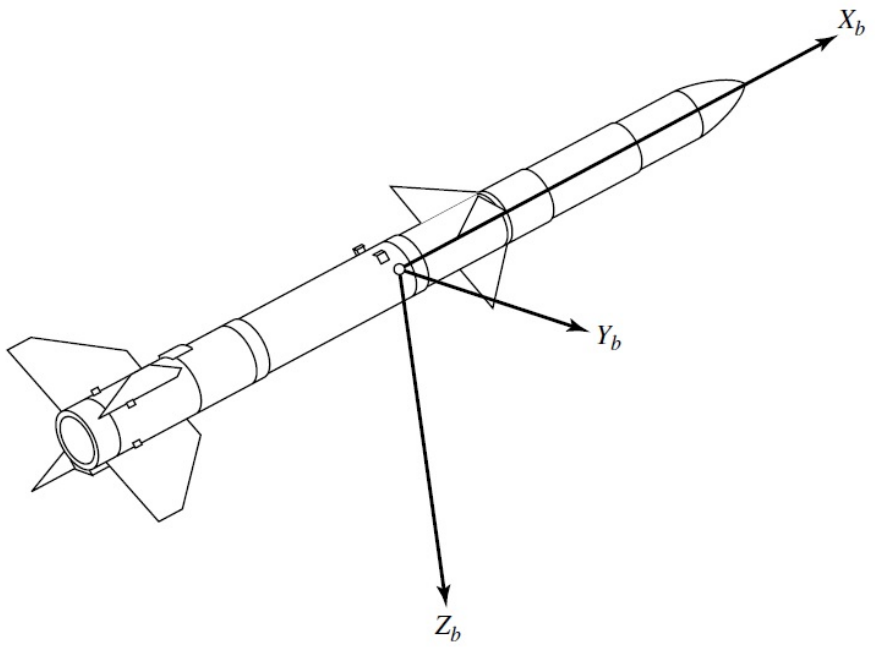

Figure 2.1: Illustration of the body-fixed frame of a missile [19]

The first column of the final matrix shown in equation (2.4) summarizes all aerodynamic forces, the second column the forces generated by the main engine $F_{X,p}$ and by the system for lateral thrusting maneuvers (Divert Attitude Control System DACS) $F_{Y,DACS}$ and $F_{Z,DACS}$ and the third column summarizes all forces caused by gravity.
In the North-East-Down (NED) coordinate system the Earth's acceleration due to gravity acts always in nadir, hence in z-direction but in the Body-Fixed frame the Earth's acceleration acting on the missile depends on the current attitude. Therefore, the gravity forces acting on the missile can be described with the roll angle ϕ and the pitch angle Θ

$$ {}_B\vec{\boldsymbol{F}}_g = \begin{bmatrix} F_{X,g} \\ F_{Y,g} \\ F_{Z,g} \end{bmatrix} = \begin{bmatrix} -m \cdot g \cdot \sin\Theta \\ m \cdot g \cdot \sin\phi \cdot \cos\Theta \\ m \cdot g \cdot \cos\phi \cdot \cos\Theta \end{bmatrix}. \tag{2.5} $$

The aerodynamic forces can be defined by

$$ {}_B\vec{\boldsymbol{F}}_A = \begin{bmatrix} F_{X,A} \\ F_{Y,A} \\ F_{Z,A} \end{bmatrix} = S \cdot \overline{q} \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \tag{2.6} $$

where C_X, C_Y and C_Z are dimensionless coefficients or rather derivatives which represent forces and moments acting on the missile. This method is quite often used in simulations in order to simplify the complex and time consuming exact calculation of these forces and moments under the assumption of neglecting wind.

The reference area is defined by S and the dynamic pressure by $\overline{q}$. The dynamic pressure can be determined by

$$\overline{q} = \frac{\rho}{2} \cdot \left[\sqrt{u^2 + v^2 + w^2}\right]^2 \tag{2.7}$$

where u, v, w are the velocity components in the body-fixed frame. These can be used to calculate the angle of attack

$$\alpha = \arctan\left(\frac{w}{u}\right) \tag{2.8}$$

and sideslip angle

$$\beta = \arctan\left(\frac{v}{\sqrt{u^2 + w^2}}\right). \tag{2.9}$$

The aerodynamics of the missile are described in a so-called aeroballistic system [27]. Therefore, the total aerodynamic angle of attack θ and total aerodynamic roll angle φ will be used (see figure 2.2). They can be calculated by

$$\theta = \arccos(\cos(\alpha)\cos(\beta)), \tag{2.10}$$

$$\varphi = \arctan\left(\frac{\tan\beta}{\sin\alpha}\right). \tag{2.11}$$

The derivatives depend on many variables (e.g. the angle of attack, sideslip angle, manipulating variables and also Mach number) and illustrate the high nonlinearity of the missile. These derivatives from equation (2.6) can be calculated by

$$C_X = C_{X00} + C_{X40}\vartheta^4 + C_{Xet2} \cdot \left(\eta^2 + \zeta^2 + 2 \cdot \xi^2\right) + C_{Xet11} \cdot (\alpha\eta - \beta\zeta) + \Delta C_X, \tag{2.12}$$

$$C_Y = C_{ZPol}\sin\varphi + C_{YPol}\cos\varphi - C_\zeta\zeta, \tag{2.13}$$

$$C_Z = C_{ZPol}\cos\varphi - C_{YPol}\sin\varphi - C_\zeta\eta, \tag{2.14}$$

where $C_\zeta = C_\eta$ due to symmetry and the manipulating variables ξ, η, ζ depend on the number of aerodynamic actuating systems. The equations of motion are described exemplary for one aerodynamic actuating system at the rear of the missile. In case of two aerodynamic actuating systems, the equations where the

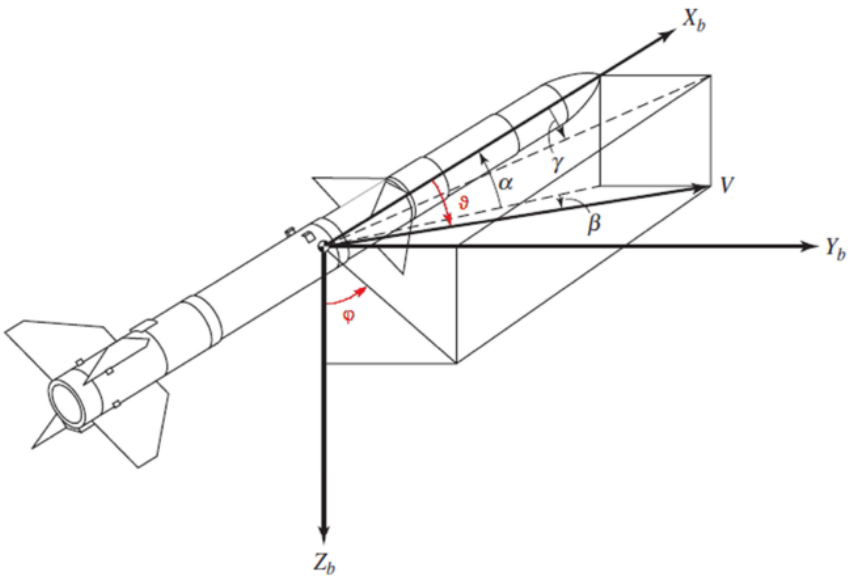

Figure 2.2: Figure 2.1 enhanced by the aerodynamic angles [19]

manipulating variables are considered have to be enhanced by an elevator, rudder and aileron at the missile's front. The whole derivation can be seen in [7].
Under the assumptions that the missile is rotationally symmetric and rotates about its principle axes of inertia it can be concluded that no deviation moments occur and $I_{yy} = I_{zz}$. The rotational behavior of the missile can then be expressed by

$$\dot{\vec{\omega}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} \cdot M_{X,t} \\ \frac{1}{I_{yy}} \cdot [M_{Y,t} + (I_{yy} - I_{xx}) \cdot p \cdot r] \\ \frac{1}{I_{yy}} \cdot [M_{Z,t} + (I_{xx} - I_{yy}) \cdot p \cdot q] \end{bmatrix} \tag{2.15}$$

where

$$\begin{bmatrix} M_{X,t} \\ M_{Y,t} \\ M_{Z,t} \end{bmatrix} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} + \begin{bmatrix} 0 \\ M_{Y,DACS} \\ M_{Z,DACS} \end{bmatrix}. \tag{2.16}$$

The moments $M_{Y,DACS}$ and $M_{Z,DACS}$ would result from the lateral thrusters. The aerodynamic moments L, M and N are defined as

$${}_B\vec{\boldsymbol{M}}_A = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = S \cdot \overline{q} \cdot l \begin{bmatrix} C_L \\ C_M \\ C_N \end{bmatrix} \tag{2.17}$$

where the derivatives are defined as stated in [19]

$$C_L = C_{L21} \vartheta^2 \sin(4\varphi) + C_{L\xi} \xi + C_{Lp} \frac{l}{V} p, \tag{2.18}$$

$$C_M = C_{MPol}\cos\varphi + C_{NPol}\sin\varphi + C_{M\eta}\eta + C_Z\frac{x_{cp}-x_{cg}}{l} + C_{Mq}\frac{l}{V}q, \quad (2.19)$$

$$C_N = -C_{MPol}\sin\varphi + C_{NPol}\cos\varphi + C_{M\eta}\zeta - C_\gamma\frac{x_{cp}-x_{cg}}{l} + C_{Mq}\frac{l}{V}r. \quad (2.20)$$

In equations (2.18) - (2.20), x_{cp} defines the pressure point, x_{cg} the center of gravity and l the reference length. The other derivatives $C_{\{.\}}$ are lodged in tables and can be interpolated by the Mach number. They are described in [7], [19].
The last equations, which are necessary to define the behavior of a missile, describe its attitude

$$\begin{bmatrix}\dot{\phi}\\ \dot{\Theta}\\ \dot{\Psi}\end{bmatrix} = \begin{bmatrix}1 & \sin\phi\tan\Theta & \cos\phi\tan\Theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi(\cos\Theta)^{-1} & \cos\phi(\cos\Theta)^{-1}\end{bmatrix}\cdot\begin{bmatrix}p\\ q\\ r\end{bmatrix}. \quad (2.21)$$

As you can see in equation (2.21), the attitude of a missile is described by the derivatives of the Euler angles where ϕ defines the roll angle, Θ the pitch angle and Ψ the yaw angle.

2.1.2 Actuating Systems

For the control of missiles, the most commonly used actuating systems are aerodynamic control surfaces or lateral thrusters. These are the elevator, rudder and aileron. When talking about these systems in this thesis, the dynamics are meant. The elevator's dynamics are described by the angle of attack α and the pitch rate q, the rudder's dynamics by the sideslip angle β and the yaw rate r and the aileron is described by the roll angle ϕ and the roll rate p.
In reality, there is always a time lag between a command and the execution of the command by the actuating system. This behavior can be approximated by a PT-1 element in the controller design.
Furthermore, actuating systems are restricted. Either by a specific maximum deflection $u_{\max}$ of a control surface or by a maximal amount of thrust of a thrusting system. This master thesis deals exclusively with the control of missiles by aerodynamic actuating systems. Especially with control surfaces at the front and back of the missile's structure. When the saturation limit $u_{\max}$ is exceeded, the missile is possibly not able to perform desired lateral accelerations and might become unstable. To counteract such problems, a so-called anti-windup network (AW NW) can be employed. More information will follow in section 2.4.

2.1.3 Missile Simulation

The missile simulation was developed by MBDA Germany to analyze the behavior of a missile system during its mission. The representation of the missile system

or rather air defense system consists beside the missile of radars and filter. The overall performance depends on all components of the system. Sometimes one has to balance between the performance of the missile and the radar and filter as well as the actuator and seeker requirements. This simulation can be used for the development and design of flight control algorithms. It is implemented in C++ and has a modular setting in order to simulate various missiles.
This simulation is connected to Simulink via S-functions. First, initial conditions of the missile are defined and transmitted to the missile simulation. Afterwards, the behavior of the missile will be computed by the equations of motion, which has been described in section 2.1.1. The output of the missile simulation are for example the actual accelerations in all directions, the current position in the NED frame or the current attitude.
A missile has four different phases which can be simulated. See figure 2.3 where an ideal thrust profile is illustrated.

1. Boost Phase where the missile will be accelerated

2. Sustain Phase for second smaller but longer thrusting if desired

3. Midcourse Phase for maximization of range

4. Endgame where the missile is very close to the target and highly dynamic maneuvers are necessary

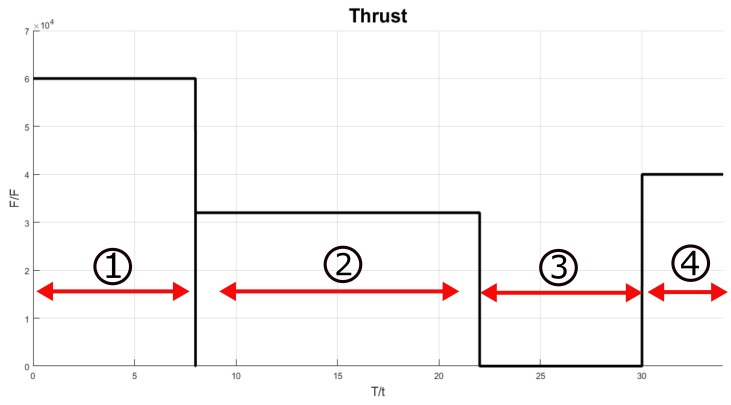

Figure 2.3: Ideal thrust profile of boost phase 1, sustain phase 2, midcourse phase 3 and endgame 4

For all four phases, initial conditions are given to simulate the correct behavior of the missile. In order to design appropriate controllers for all phases, the highly dynamic missile has to be linearized since it is due to perturbations and time dependent parameters nonlinear.

2.2 Linearization

The equations of motion which describe the behavior of a system are in general nonlinear due to uncertainties, perturbations or time varying parameters. Hence, instead of the linear state-space model

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t), \qquad \boldsymbol{x}(0) = x_0 \tag{2.22}$$
$$y(t) = \boldsymbol{c}'\boldsymbol{x}(t) + du(t) \tag{2.23}$$

or rather the linear differential equation

$$\frac{\mathrm{d}^n y}{\mathrm{d}t^n} + a_{n-1}\frac{\mathrm{d}^{n-1} y}{\mathrm{d}t^{n-1}} + ... + a_1\dot{y} + a_0 y(t) = b_q\frac{\mathrm{d}^q u}{\mathrm{d}t^q} + ... + b_1\dot{u} + b_0 u(t) \tag{2.24}$$

the system can be described with a nonlinear state-space model of the form [12]

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), u(t)), \qquad \boldsymbol{x}(0) = x_0 \tag{2.25}$$
$$y = h(\boldsymbol{x}(t), u(t)). \tag{2.26}$$

In the following, all time dependencies of the signals will be neglected in the notation due to clarity. Furthermore it should be said that state-space model (2.23) is not always equivalent to the transfer function of (2.24). Only if the system is fully observable and controllable, otherwise the order of the transfer function is smaller compared to the state-space model.
In spite of the real nonlinear behavior, linear model approximations of the missile are used for the controller design. This is because of the objective of ensuring a stable behavior at a specific OP. The missile flies through the surrounding area at that OP when the control loop works. Hence, it is sufficient that the model defines the missile's motion in the surrounding area of that OP. Therefore, the goal is to define an approximated version of the nonlinear model described by equations (2.25) and (2.26). This approximation should be valid in a local area that linear design methods can be used for the controller design.

$$\Delta\dot{\boldsymbol{x}}_i = \boldsymbol{A}_i \cdot \Delta\boldsymbol{x}_i + \boldsymbol{B}_i \cdot \Delta\boldsymbol{u}_i \tag{2.27}$$
$$\Delta\boldsymbol{y}_i = \boldsymbol{C}_i \cdot \Delta\boldsymbol{x}_i + \boldsymbol{D}_i \cdot \Delta\boldsymbol{u}_i \tag{2.28}$$

where

$\boldsymbol{A}_i$ = System matrix at an OP "i"
$\boldsymbol{B}_i$ = Input matrix
$\boldsymbol{C}_i$ = Output matrix
$\boldsymbol{D}_i$ = Feedthrough matrix
$\boldsymbol{x}_i$ = State vector (see table 2.1)
$\boldsymbol{u}_i$ = Control vector (see table 2.2)
$\boldsymbol{y}_i$ = Output vector
Δ = Implies the surrounding area of an OP where the model should be valid

The deviation from the reference point ($\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}$ and $\overline{\boldsymbol{y}} = h(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}})$) where the missile is linearized is defined as

$$\Delta \boldsymbol{x} = \boldsymbol{x} - \overline{\boldsymbol{x}}, \tag{2.29}$$
$$\Delta \boldsymbol{u} = \boldsymbol{u} - \overline{\boldsymbol{u}}, \tag{2.30}$$
$$\Delta \boldsymbol{y} = \boldsymbol{y} - \overline{\boldsymbol{y}}. \tag{2.31}$$

The linearization is exemplary shown in figure 2.4. The characteristic curve of a nonlinear static system is stated by

$$\boldsymbol{y}(t) = n(\boldsymbol{u}(t)). \tag{2.32}$$

The requirement for the OP in this example is

$$\overline{\boldsymbol{y}} = n(\overline{\boldsymbol{u}}) \tag{2.33}$$

which means that OP $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{y}})$ is situated on the nonlinear characteristic curve.

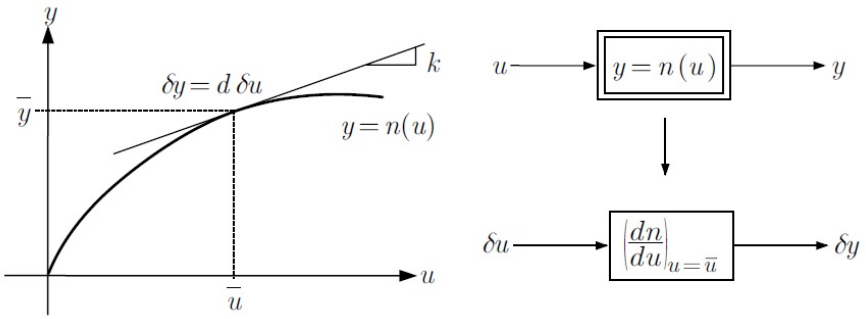

Figure 2.4: Linearization example of a static nonlinearity [12]

It is recommended to substitute the missile into linear models at steady states since the expected changes at these reference points are very small ($\dot{\boldsymbol{x}} \approx 0$) which leads to an enlargement of Δ and hence the scope of the missile. The requirement for the reference points is that the lateral acceleration is zero, i.e. $\overline{\boldsymbol{y}} = \overline{\boldsymbol{a}} = h(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}) = 0$. To fulfill this requirement, a so-called trimming of the missile is necessary. The trimming method and its implementation is described in section 2.2.1.

The linearization about a trim state to get the substituted models is done by a Taylor expansion. Using equations (**??**) and (2.31) leads to the following deviation:

$$\Delta \dot{\boldsymbol{x}} = \dot{\boldsymbol{x}} - \dot{\overline{\boldsymbol{x}}} \tag{2.34}$$
$$= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{f}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}) \tag{2.35}$$
$$= \boldsymbol{f}(\overline{\boldsymbol{x}} + \Delta \boldsymbol{x}, \overline{\boldsymbol{u}} + \Delta \boldsymbol{u}) - \boldsymbol{f}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}) \tag{2.36}$$

Table 2.1: States of a missile

State	Description	Unit
V	Kinematic velocity	m/s
α	Angle of attack in kinematic system	rad
β	Sideslip angle in kinematic system	rad
p	Roll rate in body-fixed system	rad/s
q	Pitch rate in body-fixed system	rad/s
r	Yaw rate in body-fixed system	rad/s
ϕ	Roll angle between BF and NED	rad
Θ	Pitch angle between BF and NED	rad
Ψ	Yaw angle between BF and NED	rad

Table 2.2: Control parameter of a missile

Control Parameter	Description	Unit
ξ_f	Deflection of front aileron	rad
η_f	Deflection of front elevator	rad
ζ_f	Deflection of front rudder	rad
ξ_b	Deflection of back aileron	rad
η_b	Deflection of back elevator	rad
ζ_b	Deflection of back rudder	rad

This yields the following Taylor expansion

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{f}(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}}) + \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\overline{\boldsymbol{x}}\\ \boldsymbol{u}=\overline{\boldsymbol{u}}}} \cdot \Delta\boldsymbol{x} + \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\overline{\boldsymbol{x}}\\ \boldsymbol{u}=\overline{\boldsymbol{u}}}} \cdot \Delta\boldsymbol{u} + \boldsymbol{O}^2(\boldsymbol{x},\boldsymbol{u}) - \boldsymbol{f}(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}}). \tag{2.37}$$

Since the missile is highly dynamic, it can be linearized only in a small surrounding area of the OP. Therefore, $\Delta\boldsymbol{x}(t)$ and $\Delta\boldsymbol{u}(t)$ are relatively small and terms of higher order $\boldsymbol{O}^2$ can be neglected compared to terms of first rank. Equation (2.37) can the be written as

$$\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\overline{\boldsymbol{x}}\\ \boldsymbol{u}=\overline{\boldsymbol{u}}}} \cdot \Delta\boldsymbol{x} + \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\overline{\boldsymbol{x}}\\ \boldsymbol{u}=\overline{\boldsymbol{u}}}} \cdot \Delta\boldsymbol{u}. \tag{2.38}$$

The two differential quotients from equation (2.38) define derivatives of the states and control parameters and describe therefore the system matrix $\boldsymbol{A}$ (equation (2.39)) and input matrix $\boldsymbol{B}$ (equation (2.40)) and are called Jacobi-matrices [12].

$$\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\overline{\boldsymbol{x}}\\ \boldsymbol{u}=\overline{\boldsymbol{u}}}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{\substack{\boldsymbol{x}=\overline{\boldsymbol{x}}\\ \boldsymbol{u}=\overline{\boldsymbol{u}}}} = \boldsymbol{A} \tag{2.39}$$

$$\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\overline{\boldsymbol{x}}\\ \boldsymbol{u}=\overline{\boldsymbol{u}}}} = \left[\begin{array}{cccc} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_n} \end{array}\right]_{\substack{\boldsymbol{x}=\overline{\boldsymbol{x}}\\ \boldsymbol{u}=\overline{\boldsymbol{u}}}} = \boldsymbol{B} \tag{2.40}$$

This yields the linear relation as already stated in equation (2.28):

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{A}\cdot\Delta\boldsymbol{x} + \boldsymbol{B}\cdot\Delta\boldsymbol{u} \tag{2.41}$$

$$\tag{2.42}$$

The output matrix $\boldsymbol{C}$ and feedthrough matrix $\boldsymbol{D}$ can be determined with the same procedure but using a Taylor expansion of function $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})$.

2.2.1 Trimming

Trimming is used to determine trim variables $(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}})$ which define a subset of state derivatives. This state can be determined when the derivatives of the states are zero or approximately zero. The trim variables are a set of variables, wisely chosen from the states and control parameters shown in tables 2.1 and 2.2. For the numerical minimization of the derivatives of the states, a cost function has to be determined for the sum of these derivatives. To find the minimal cost function for the definition of the desired trim variables, an optimization algorithm can be used, which is provided by MATLAB and is called `fminsearch`. This function uses the gradient-based Nelder-Mead Simplex Algorithm [4].
A steady-state of a missile is for example a "straight and level" flight where the missile will not experience lateral acceleration steps or the height is increased or decreased. The associated trim variables are the angle of attack α and the deflection of the elevators η_f and η_b. It is sufficient to define only the elevator at the back as a trim variable. For a higher performance, the elevator at the front can be called in.
These variables have to be determined in order to eliminate the derivatives of the Euler angles, the rotation rates the angle of attack and the sideslip angle. Since the missile will be accelerated when the booster is burning and decelerated by the aerodynamic drag when no thrust is present, the velocity can not be controlled. Hence, the derivative of the velocity will not be zero but is handled as steady-state [3], [7].

Consider the previously mentioned case of straight and level flight where the trim variables are set to α and η_b. The requirement is

$$\overline{\boldsymbol{y}} = \overline{\boldsymbol{a}} = \boldsymbol{0} = \boldsymbol{h}(\overline{\boldsymbol{\alpha}},\overline{\boldsymbol{\eta_b}}) \tag{2.43}$$

which is fulfilled by

$$\dot{\boldsymbol{x}}_i = \boldsymbol{f}(\overline{\boldsymbol{\alpha}},\overline{\boldsymbol{\eta_b}}) = \boldsymbol{0}, \forall i. \tag{2.44}$$

The requirement of the lateral accelerations, states and control inputs for such an example are summarized in table 2.3.

Table 2.3: States and control parameters for a straight and level flight

Parameter	Description	Characteristic
	Requirement	
a	Pre-defined	$y_a = 0$, $z_a = 0$
	States	
V	Handled as steady-state	$V \neq 0$, $\dot{V} = 0$
α	Trim variable/Wanted	$\alpha \neq 0$, $\dot{\alpha} = 0$
β	No requirement/Arbitrary	$\beta \in \mathbb{R}$, $\dot{\beta} = 0$
p	No requirement/Arbitrary	$p \in \mathbb{R}$, $\dot{p} = 0$
q	No requirement/Arbitrary	$q \in \mathbb{R}$, $\dot{q} = 0$
r	No requirement/Arbitrary	$r \in \mathbb{R}$, $\dot{e} = 0$
ϕ	No requirement/Arbitrary	$\phi \in \mathbb{R}$, $\dot{\phi} \in \mathbb{R}$
Θ	No requirement/Arbitrary	$\Theta \in \mathbb{R}$, $\dot{\Theta} \in \mathbb{R}$
Ψ	No requirement/Arbitrary	$\Psi \in \mathbb{R}$, $\dot{\Psi} \in \mathbb{R}$
	Control Inputs	
ξ_f	Set to zero	$\xi_f = 0$
η_f	Set to zero	$\eta_f = 0$
ζ_f	Set to zero	$\zeta_f = 0$
ξ_b	Set to zero	$\xi_b = 0$
η_b	Trim variable/Wanted	$\eta_b \neq 0$
ζ_b	Set to zero	$\zeta_b = 0$

2.2.2 Operating Point Grid

The missile is linearized at specific operating points, spread over the whole flight envelope of the missile. As described in section 2.2.1, the missile can be linearized for straight and level flights. To increase the performance one can trim also over the angle of attack, the center of gravity or dynamic pressure. The present master thesis deals exclusively with the linearization over the Mach number, height and dependent on the user input, linearization over the angle of attack is possible.
The bounds of the grid are set by the operating range of the missile. Within these bounds, the operating points are separated regular or irregular. At these points, controllers are designed. Since the surrounding area of an OP where the linearized model is valid is not equal at all operating points, irregular grids have the advantage of a smaller number of operating points and therefore less controllers. Regular grids adjust the distance between the points to the smallest valid surrounding area of an OP, present in the grid.

2.3 Convex Optimization

Convex optimization is a special class of mathematical optimization problems [22]. These have the essential characteristic that every local optimum is also a global optimum[9]. The goal is to minimize the objective function under certain constraints. These constraints are stated as equalities or inequalities as exemplary shown in the following general optimization problem (2.45)

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, ..., m \end{array} \tag{2.45}$$

where

$$\begin{array}{ll} \boldsymbol{x} = (x_1, ..., x_n) & = \text{Optimization variable} \\ f_0 : \boldsymbol{R}^n \rightarrow \boldsymbol{R} & = \text{Objective function} \\ f_i : \boldsymbol{R}^n \rightarrow \boldsymbol{R}, i = 1, ..., m & = \text{(Inequality) constraint functions} \\ b_1, ..., b_m & = \text{Limits of the constraints.} \end{array}$$

If a vector $\boldsymbol{x}^*$ has the smallest objective value among all other vectors that satisfy the constraints, vector $\boldsymbol{x}^*$ is called optimal. This means that we have for any vector $\boldsymbol{z}$ with $f_1(z) \leq b_1, ..., f_m(\boldsymbol{z}) \leq b_m$, $f_0(\boldsymbol{z}) \geq f_0(\boldsymbol{x}^*)$.
If the optimization problem is convex, the objective- and constraint functions are convex [21]

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \tag{2.46}$$

if $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.
To give a better understanding of a convex quantity, a convex and nonconvex set are illustrated in figure 2.5. A convex set contains a line segment which is an affine set between any two points in the set

$$x_1, x_2 \in C, 0 \leq \theta \leq 1 \Longrightarrow \theta x_1 + (1 - \theta) x_2 \in C. \tag{2.47}$$

No analytical formulas but very reliable and efficient algorithms exist to solve convex optimization problems. In practice the Interior-Point Method (IPM) works very well and solve the problem to a specified accuracy. Furthermore, convex optimization problems are often difficult to recognize. However, many tricks exist for the transformation of a problem in a convex form. LMIs are one option of convex optimization problems that arise in system and control theory [20]. The next section is intended to explain linear matrix inequality constraints and their computational rules used in convex optimization.

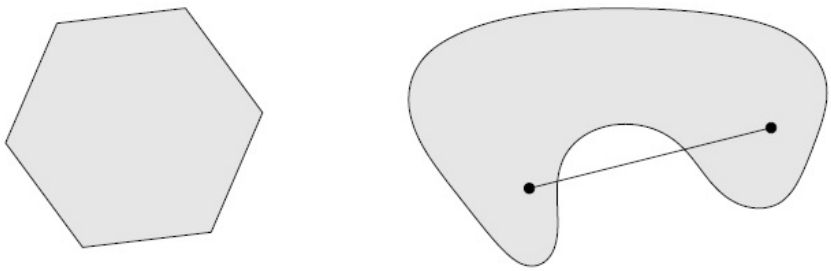

Figure 2.5: Convex set (left) and nonconvex set (right) [21]

2.3.1 Linear Matrix Inequalities (LMIs)

Since optimization problems employing LMIs are convex, every local optimum is at the same time a global optimum. Therefore, local search methods can be applied to find the global optimum. In control theory convex optimization problems can be constrained by linear matrix inequalities. Such a LMI can be stated as [9], [20]

$$\boldsymbol{F}(\boldsymbol{x}) \equiv \boldsymbol{F}_0 + \sum_{i=1}^{m} x_i \boldsymbol{F}_i \succ 0 \tag{2.48}$$

where the symmetric matrices $\boldsymbol{F}_i \in \mathbb{R}^{nxn}$ are given and the wanted vector $x \in \mathbb{R}^m$ contains the variables. The notation $\boldsymbol{F}(\boldsymbol{x}) \succ 0$ means that the matrix $\boldsymbol{F}(\boldsymbol{x})$ is symmetric and positive definite. Hence, the following conditions have to be fulfilled:

- $\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x} > 0$ for all $\boldsymbol{x} \in \mathbb{R}^n \backslash \{\boldsymbol{0}\}$
- $\lambda_i > 0$ for all eigenvalues $\lambda_i, i = 1, ..., n$
- All first minors $\boldsymbol{A}_i$ apply $\det(\boldsymbol{A}_i) > 0$

The name "Linear Matrix Inequality" has been established because of the parameter x_i which occurs only in a linear manner when the inequality is written as a matrix. But a LMI is only a system of linear inequalities when the matrices $\boldsymbol{F}_i(\boldsymbol{x})$ are diagonal.
The set $\{\boldsymbol{x}|\boldsymbol{F}(\boldsymbol{x}) > 0\}$ is convex since the LMI (2.48) is a convex constraint on $\boldsymbol{x}$. This expression of the LMI is quite general. But it gets more clearly when using stability in the sense of Lyapunov [28]. The autonomous system $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ is stable when the following inequalities are fulfilled

$$\boldsymbol{A}'\boldsymbol{R} + \boldsymbol{R}\boldsymbol{A} \prec 0, \tag{2.49}$$

$$\boldsymbol{R} \succ 0 \tag{2.50}$$

where $\boldsymbol{R}$ describes a positive definite symmetric matrix. Since LMIs have the characteristic of being convex, they are solvable very efficiently with numerical methods.

2.3.1.1 Multiple LMIs

Another property of LMIs is that multiple linear matrix inequalities can be summarized. For example the LMIs $\boldsymbol{A} \succ 0$, and $\boldsymbol{B} \succ 0$ can be summarized by

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix} \succ 0. \tag{2.51}$$

This can be proved taking a closer look on the eigenvalues. Since the matrix $\boldsymbol{C}$ is diagonal, it has the same eigenvalues like $\boldsymbol{A}$ and $\boldsymbol{B}$ and hence the definiteness.

2.3.1.2 Uniqueness

Furthermore, LMIs are not unique, i.e. the same solution can be characterized by different LMIs. For example if the matrix $\boldsymbol{M}$ is regular, the following two LMIs are equivalent to each other.

$$\boldsymbol{A} \succ 0 \tag{2.52}$$
$$\boldsymbol{M}'\boldsymbol{A}\boldsymbol{M} \succ 0 \tag{2.53}$$

2.3.1.3 Schur-Complement

The Schur-complement yields a set of linear matrix inequalities out of convex nonlinear matrix inequalities. The Schur-complement indicates that the inequalities

$$\begin{aligned} \boldsymbol{R} &\succ 0 \\ \boldsymbol{Q} - \boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}' &\succ 0 \end{aligned} \tag{2.54}$$

are equivalent to the LMI

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}' & \boldsymbol{R} \end{bmatrix} \succ 0 \tag{2.55}$$

where $\boldsymbol{R}$ and $\boldsymbol{Q}$ have to be symmetric.

2.3.1.4 Substitution

Sometimes a nonlinear problem exist like for example the controller design for linear plants. It is necessary to convert it into a linear matrix inequality by a clever substitution. For a closed control loop, the system matrix can be written as

$$\hat{\boldsymbol{A}} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}. \tag{2.56}$$

Hence, the Lyapunov inequality in the variables $\boldsymbol{K}$ and $\boldsymbol{R}$ for the controller design is then

$$\begin{aligned} \boldsymbol{R} &\succ 0 \\ (\boldsymbol{A}-\boldsymbol{BK})'\boldsymbol{R}+\boldsymbol{R}(\boldsymbol{A}-\boldsymbol{BK}) &\prec 0. \end{aligned} \tag{2.57}$$

Since terms of $\boldsymbol{RBK}$ exist, equation (2.57) is no LMI in $\boldsymbol{R}$ and $\boldsymbol{K}$. Introducing a new regular matrix $\boldsymbol{Q}=\boldsymbol{R}^{-1}$ as well as the variable $\boldsymbol{W}=\boldsymbol{KQ}$, equation (2.57) becomes a LMI in $\boldsymbol{Q}$ and $\boldsymbol{W}$

$$\begin{aligned} \boldsymbol{Q} &\succ 0 \\ \boldsymbol{QA}'+\boldsymbol{AQ}-\boldsymbol{BW}-\boldsymbol{W}'\boldsymbol{B}^T &\prec 0. \end{aligned} \tag{2.58}$$

2.3.1.5 Solution Methods

Two main algorithms exist for solving LMIs. The IPM and the ellipsoid method. Easier to understand is the ellipsoid method but more efficient is the IPM. A detailed explanation of these algorithms can be seen in [8] and [14].
For the solution of the LMIs, two solvers will be used in this thesis.

- LMI Lab which is part of the Robust Control Toolbox of MATLAB
- SDPT3 which is a MATLAB package available free of charge for semidefinite-quadratic-linear programming [17]

In general, both solvers can be applied to three different kinds of problems:

- Solvability or rather feasibility
- Eigenvalue problem
- Generalized Eigenvalue Problem (GEVP)

The main difference between these solvers is that the LMI Lab is numerical more robust compared to the SDPT3 solver but the SDPT3 solver is much faster than the LMI Lab. Both, the LMI Lab and the SDPT3 solver, have the disadvantage of a difficult notation in MATLAB. But another MATLAB toolbox available free of charge makes it more user-friendly. It is called YALMIP [11] and works as an interface for the LMI Lab and the SDPT3 solver among others. By using this interface, LMIs can be written in their original matrix notation.

2.3.2 LMIs in Control Engineering

This section is intended to present some possible applications of LMI optimization problems in control engineering. For simplification, only Single Input Single Output (SISO) Linear Time Invariant (LTI) systems of the following form will be investigated

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax}+\boldsymbol{b}u, \tag{2.59}$$

$$\boldsymbol{y} = \boldsymbol{c}'\boldsymbol{x}. \tag{2.60}$$

2.3.2.1 Stability of the Open Loop

For the investigation of asymptotic stability of system (2.60) in the open loop, i.e. $u = 0$, one can ask for the existence of a Lyapunov function $v = \boldsymbol{x}'\boldsymbol{R}\boldsymbol{x}$ for which the following condition is fulfilled

$$\dot{v} = \boldsymbol{x}'(\boldsymbol{A}^T\boldsymbol{R} + \boldsymbol{R}\boldsymbol{A})\boldsymbol{x} < 0. \tag{2.61}$$

Formulating this problem as a LMI one has to ask for the existence of a symmetric matrix $\boldsymbol{R}$ which fulfills

$$\begin{aligned} \boldsymbol{R} &\succ 0, \\ \boldsymbol{A}'\boldsymbol{R} + \boldsymbol{R}\boldsymbol{A} &\prec 0. \end{aligned} \tag{2.62}$$

Multiplying these inequalities from left to right with $\boldsymbol{R}^{-1}$ and defining the new variable $\boldsymbol{Q} = \boldsymbol{R}^{-1}$ leads to the following representation which is not necessary in this case but will be useful for the case of stability investigation of the closed loop

$$\begin{aligned} \boldsymbol{Q} &\succ 0, \\ \boldsymbol{Q}\boldsymbol{A}' + \boldsymbol{A}\boldsymbol{Q} &\prec 0. \end{aligned} \tag{2.63}$$

This example should have demonstrated the feasibility problem, mentioned in section 2.3.1.5. All feasible solutions are of equal value since the task here is only to find a feasible matrix $\boldsymbol{R}$ or $\boldsymbol{Q}$ which satisfies the inequality.

2.3.2.2 Stability of the Closed Loop

Now, a linear state feedback controller $u = -\boldsymbol{k}'\boldsymbol{x}$ is used and the variables are now vector $\boldsymbol{k}$ and matrix $\boldsymbol{R}$ which has to satisfy the inequalities stated in equation (2.64)

$$\begin{aligned} \boldsymbol{R} &\succ 0 \\ (\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k})'\boldsymbol{R} + \boldsymbol{R}(\boldsymbol{A} - \boldsymbol{b}\boldsymbol{k}) &\prec 0. \end{aligned} \tag{2.64}$$

Due to the product of $\boldsymbol{R}\boldsymbol{b}\boldsymbol{k}'$ one has to employ substitutions as explained in section 2.3.1.4. This method leads to the following LMI in the matrix variables $\boldsymbol{Q}$ and $\boldsymbol{w}$

$$\begin{aligned} \boldsymbol{Q} &\succ 0 \\ \boldsymbol{Q}\boldsymbol{A}' + \boldsymbol{A}\boldsymbol{Q} - \boldsymbol{b}\boldsymbol{w} - \boldsymbol{w}'\boldsymbol{b}^T &\prec 0. \end{aligned} \tag{2.65}$$

The controller can then be calculated from the solution of LMI (2.65) by $\boldsymbol{k}^T = \boldsymbol{w}^T\boldsymbol{Q}^{-1}$. The validation of an existing stabilizing controller verifies stability of the system but still no optimization is present and hence no statements on the performance can be made.

2.3.2.3 Actuator Limitation

In general, every completely controllable LTI system can be stabilized. When adding the constraint of for example a limitation of the actuator, linear methods are no longer able to guaranty stability [9]. One approach is then to find a controller which is able to stabilize a specific initial condition $\boldsymbol{x}_0$ without exceeding a defined actuating limitation u_{max}. The manipulated variable $u(t)$ will be defined along an arbitrary trajectory

$$\max_t |\boldsymbol{u}(t)| = \max_t |\boldsymbol{k}'\boldsymbol{x}(t)|. \tag{2.66}$$

Investigating a Lyapunov region $\mathcal{G}$ which contains $\boldsymbol{x}_0$, contains also the whole trajectory which makes the time dependency obsolete

$$\mathcal{L} = \{\boldsymbol{x} : |\boldsymbol{k}'\boldsymbol{x}| \leq \boldsymbol{u}_{\max}\} = \{\boldsymbol{x} : \boldsymbol{x}'\boldsymbol{k}\boldsymbol{k}'\boldsymbol{x} \leq \boldsymbol{u}_{\max}^2\},\ \boldsymbol{x}(0) \in \mathcal{G} \tag{2.67}$$

where $\mathcal{G} = \{\boldsymbol{x}|\boldsymbol{x}^T\boldsymbol{R}\boldsymbol{x} \leq 1\}$ for LTI systems. The whole derivation can be seen in [9]. The solution for the stated problem of finding a controller which is able to stabilize a specific initial condition x_0 without exceeding a defined actuating limitation $u_{\max}$ is in LMI notation

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{w} \\ \boldsymbol{w}' & \boldsymbol{u}_{max}^2 \end{bmatrix} \succeq 0. \tag{2.68}$$

2.3.2.4 Inclusion of an Initial Region

The limitations are satisfied locally in a Lyapunov region $\mathcal{G}$. Therefore, it has to be investigated whether this region contains the set of all possible initial conditions $\mathcal{X}_0$. This is quite simple when $\mathcal{X}_0$ is a convex polytope (for example a rectangle) or an ellipse. In case of a convex polytope, it is sufficient to verify whether its vertices are part of $\mathcal{G}$. This can be expressed using as many LMIs as the polytope has vertices. They have the form like following stated

$$1 - \boldsymbol{x}_0^T\boldsymbol{R}\boldsymbol{x}_0 \geq 0 \tag{2.69}$$

$$\Leftrightarrow \begin{bmatrix} 1 & \boldsymbol{x}_0^T \\ \boldsymbol{x}_0 & \boldsymbol{R}^{-1} \end{bmatrix} \succeq 0 \tag{2.70}$$

$$\Leftrightarrow \begin{bmatrix} 1 & \boldsymbol{x}_0^T \\ \boldsymbol{x}_0 & \boldsymbol{Q} \end{bmatrix} \succeq 0. \tag{2.71}$$

If $\mathcal{X}_0$ is an ellipse $\mathcal{E}(\mathcal{X}, 1)$, the LMI is

$$\boldsymbol{R} - \boldsymbol{X} \geq 0 \tag{2.72}$$

$$\Leftrightarrow \begin{bmatrix} \boldsymbol{Q} & \mathbb{I} \\ \mathbb{I} & \boldsymbol{X} \end{bmatrix} \succeq 0 \tag{2.73}$$

$$\tag{2.74}$$

where $\mathbb{I}$ is the identity matrix.

2.3.2.5 Output Energy

The controller design which makes use of convex optimization needs also an objective function. All possible controllers $\boldsymbol{k}$ and possible matrices $\boldsymbol{Q}$ are constrained by equations (2.71), (2.68) and 2.65 in such a way that $\boldsymbol{k}$ is a valid controller which stabilizes all $\boldsymbol{x}_0 \in \mathcal{X}_0$. In case no controller which can hold these constraints exist this quantity is empty. Remedies are either increasing u_{max} or decreasing $\mathcal{X}_0$. If the quantity of possible controllers is not empty, one has to select the best controller. This evaluation can be done with the help of a convex function that the whole optimization problem stays convex.
A good measure for the control accuracy is the output energy

$$J = \int_0^\infty y^2 \mathrm{d}t \qquad \text{where } y = \boldsymbol{c}^T\boldsymbol{x}. \tag{2.75}$$

The goal is to minimize the upper limit of J for all $\boldsymbol{x}$ of the Lyapunov region

$$\mathcal{G} = \{\boldsymbol{x}^T\boldsymbol{R}\boldsymbol{x} \leq 1\}. \tag{2.76}$$

By introducing a performance parameter $\gamma > 0$ the control accuracy is

$$J \leq \gamma \boldsymbol{x}_0^T\boldsymbol{R}\boldsymbol{x}_0 \leq \gamma. \tag{2.77}$$

The whole optimization problem can then be written as

$$\begin{aligned}
&\text{minimize } \gamma \\
&\text{subject to } \boldsymbol{Q} \succ 0 \\
&\text{Stability and Performance:} \\
&\begin{bmatrix} \boldsymbol{A}\boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{A}^T - \boldsymbol{b}\boldsymbol{w}^T - \boldsymbol{w}\boldsymbol{b}^T & \boldsymbol{Q}\boldsymbol{c} \\ \boldsymbol{c}^T\boldsymbol{Q} & -\gamma \end{bmatrix} \prec 0 \\
&\text{Inclusion of Initial Region:} \\
&\begin{bmatrix} 1 & \boldsymbol{x}_0^T \\ \boldsymbol{x}_0 & \boldsymbol{Q} \end{bmatrix} \succeq 0 \\
&\text{Actuator Limitation:} \\
&\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{w} \\ \boldsymbol{w}^T & u_{max}^2 \end{bmatrix} \succeq 0
\end{aligned} \tag{2.78}$$

The whole derivation can be found again in [9].

2.4 Anti-Windup

This section is intended to describe the basics of anti-windup problems and how to solve them. A detailed explanation of the anti-windup method which was used in the course of this thesis will be presented in section 3.6. The information used to describe anti-windup are mainly taken from the doctoral theses of Dr. Klaus Kefferpütz [9] and Dr. Andreas Ortseifen [15].
The controller of a control loop calculates a manipulated variable in order to influence a system as desired. The commanded manipulated variable will be realized by an actuator which transposes the variable into a physical quantity. In reality, no perfect actuator exists. The actuator has its own dynamic and is limited in the actuating range. This was already explained for missiles in section 2.1.2. The actuator dynamics can be taken into account in the controller design but the limits of the manipulating variables need more attention [9].

2.4.1 General Windup Problem

The classic problem of windup is connected to controllers with an integral part like a PID controller. Figure 2.6 illustrates a control loop with a saturation block which defines the limits of the actuating range. The characteristic curve of the saturation is defined by

$$u_s = \mathrm{sat}(u_a) = \begin{cases} u_a & \text{if } u_{\min} \leq u_a \leq u_{\max}, \\ u_{\min} & \text{if } u_a < u_{\min}, \\ u_{\max} & \text{if } u_a > u_{\max}. \end{cases} \tag{2.79}$$

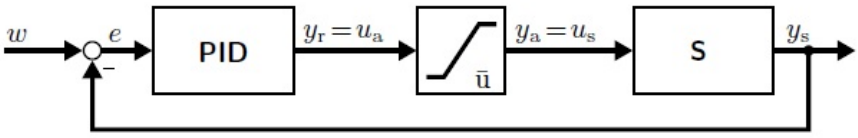

Figure 2.6: Control loop with a PID controller, a linear plant **S** and an actuator with limitations in the actuating range [15]

The controller is defined by

$$u_a(t) = K_p e(t) + K_i \int_0^l e(\tau)\mathrm{d}\tau + K_d \frac{\mathrm{d}}{\mathrm{d}t} e(t) \tag{2.80}$$

where $e(t) = w(t) - y_s(t)$ defines the error between reference variable $w(t)$ and control variable $y_s(t)$. As long as this control error remains small, the control loop works linear, i.e. unsaturated and the commanded manipulated variable remains small too and holds $u_s = u_a$. If the deviation from the reference variable is too

large, $u_a > u_{\max}$ or $u_a < u_{\min}$ and the saturation block, defined in equation (2.79), delivers $u_a = u_{\max}$ or $u_a = u_{\min}$. The behavior of the control loop is now nonlinear, therefore the control error will not be reduced which leads to an increase of u_a, resulting from the integration of the error which is part of the control law stated in equation (2.80). This behavior is obviously not useful since the manipulated variable reached its limit. The integral part increases until the manipulated variable changes sign. Therefore, the high integral part has to be reduced. The problem of an increasing integral part defines the original windup problem and can lead to a dramatic degradation of control accuracy and in worst case instability.
As time goes by, the term "windup" was generalized that all undesired behavior of a system induced by actuating limits is referred to as windup. In particular, windup can also occur exclusively induced by the controller where no integral part is present. This happens when the controller dynamics are marginal stable or unstable. In case of saturation the constant manipulating variable ($u_s = u_{\min}$ or $u_s = u_{\max}$) is applied to the plant and changes of the plant output y_s will not affect the plant input u_s since the controller has no feedback here. The result is an increase of the controller states which leads to a big overshoot, oscillations and instability.

2.4.2 Anti-Windup Methods

A frequently used approach is designing the controller without consideration of actuator limits until a desired behavior of the unsaturated control loop occurs. This is done by preventing the occurrence of windup by a conservative design of controller gains. The two big resulting disadvantages are firstly oversizing of the actuator and hence loss of performance since the complete actuating range will not be used and secondly unexpected conditions which have not been predicted can lead to a nonlinear saturated behavior.
In the last decades, many methods have been developed where the controller was designed under the explicit consideration of actuating limits. The disadvantage is that classic linear design methods can not be used any more [9]. Other widely spread methods are called Anti-Windup methods. It is convenient here to design the controller in a first step without the consideration of actuating limits. This controller has a desired performance in the absence of saturation. Afterwards, the controller will be extended by a so called AW NW. A general anti-windup network can influence the controller states and outputs. In our case, the controller outputs and inputs will be influenced. Such a network has the following goals:

- In the absence of saturation, the anti-windup network will not influence the behavior of the control loop
- In case of saturation, stability should be guaranteed
- After saturation, the unlimited system behavior should be recovered as good as possible

For clarification see figure 2.7. Here, the differences between a system using a simple anti-windup network, a system which uses no anti-windup and a linear system, i.e. which has no saturation limits, is shown. The controller remains fixed. The arbitrary asymptotic stable system was in this example limited to ± 5. The result is obviously a much smaller overshoot of the system using anti-windup compared to the system which is not. Compared to the unlimited linear system, the others are not that dynamic and need more time to reach the set-point value (reference).

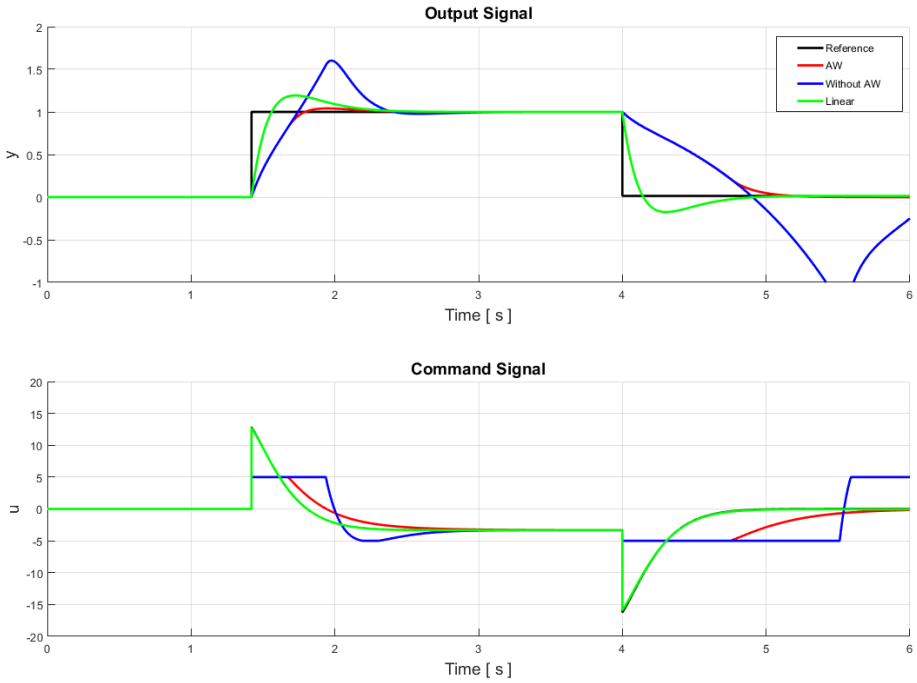

Figure 2.7: Comparison between a system using a simple anti-windup network (—), a system which uses no anti-windup (—) and an unsaturated system (—)

A simple anti-windup network has the big disadvantage that it does not guaranty a good performance or robustness. A specific kind of anti-windup is the so called Model Recovery Anti-Windup (MRAW) (see figure 2.8) which will be discussed in the next section.

2.4.2.1 Model Recovery Anti-Windup (MRAW)

Model-recovery anti-windup methods use a model of the plant to avoid windup. They have a structure like the external anti-windup network illustrated in figure 2.8. The dynamics of such a network are mainly defined by the model of the plant

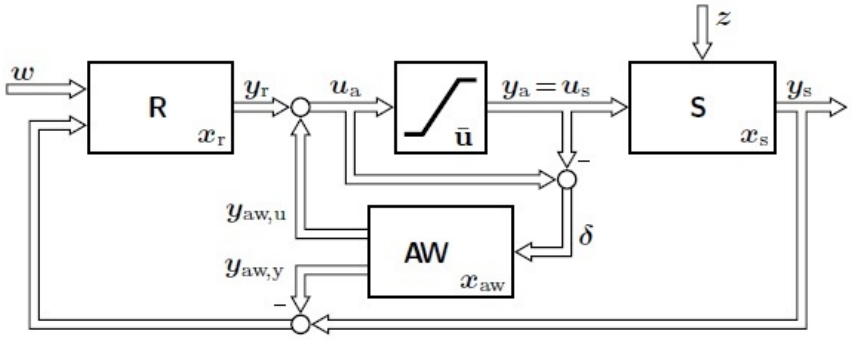

Figure 2.8: Illustration of an external anti-windup network (MRAW) which influences the controller input and output [15]

$$\dot{\boldsymbol{x}}_{aw} = \boldsymbol{A}_s \boldsymbol{x}_{aw} + \boldsymbol{B}_{s,u} \boldsymbol{k}_{aw}(\boldsymbol{x}_{aw}) + \boldsymbol{B}_{s,u}\delta, \tag{2.81}$$

$$\boldsymbol{y}_{aw,y} = -\boldsymbol{C}_s \boldsymbol{x}_{aw}, \tag{2.82}$$

$$\boldsymbol{y}_{aw,u} = -\boldsymbol{K}_{aw} \boldsymbol{x}_{aw} \tag{2.83}$$

where $\boldsymbol{x}_{aw} \in \mathbb{R}^{n_s}$ and n_s is the order of the system which is the same as the anti-windup's order. Index s is now used for the plant's matrices. Only $\boldsymbol{K}_{aw}$ has to be defined in the anti-windup design. These formulations lead to the following advantages:

- Nonlinear AW: Nonlinear anti-windup methods, which ensures a good performance, can be applied due to the function $\boldsymbol{K}_{aw}$
- Simple Design: The anti-windup design is reduced to a simple state space controller design for the model of the plant
- Good Quality Criteria: The mismatch between the unlimited and limited system behavior can be directly influenced in the anti-windup design

The network shown in equations (2.81) - (2.83) works as described subsequently

1. The AW system (equation (2.81)) influences the control loop only if a difference $\boldsymbol{\delta} = \boldsymbol{u} - \text{sat}(\boldsymbol{u}) \neq 0$ under the constraint of $\boldsymbol{x}_{aw}(0) = \boldsymbol{0}$ occurs
   - $\boldsymbol{u}$ is the necessary manipulated variable
   - $\text{sat}(\boldsymbol{u})$ the actual realizable manipulated variable
2. Equation (2.82) leads to a reduction of the measure $\boldsymbol{y}_s$ to avoid an uncontrolled integration of the controller states
3. If the plant is unstable, $\boldsymbol{x}_{aw}$ will increase even though $\delta = 0$ after saturation. The feedback via $\boldsymbol{K}_{aw}$ (equation (2.83)) yields a fast recovery ($\boldsymbol{x}_{aw} = 0$), i.e. the closed loop system (2.81) will be stabilized after the impulse $\boldsymbol{\delta}$ is dispersed

The whole state space model of the overall system consisting of the plant, controller and the anti-windup control loop is defined by

$$\dot{\boldsymbol{x}}_s = \boldsymbol{A}_s\boldsymbol{x}_s + \boldsymbol{B}_{s,u}\boldsymbol{u}_s + \boldsymbol{B}_{s,z}\boldsymbol{z}, \tag{2.84}$$
$$\dot{\boldsymbol{x}}_r = \boldsymbol{A}_r\boldsymbol{x}_r + \boldsymbol{B}_{r,u}\boldsymbol{C}_s(\boldsymbol{x}_s + \boldsymbol{x}_{aw}) + \boldsymbol{B}_{r,w}\boldsymbol{w}, \tag{2.85}$$
$$\dot{\boldsymbol{x}}_{aw} = \boldsymbol{A}_s\boldsymbol{x}_{aw} + \boldsymbol{B}_{s,u}\boldsymbol{y}_r - \boldsymbol{B}_{s,u}\boldsymbol{u}_s, \tag{2.86}$$
$$\boldsymbol{y}_r = \boldsymbol{C}_r\boldsymbol{x}_r + \boldsymbol{D}_{r,u}\boldsymbol{C}_s(\boldsymbol{x}_s + \boldsymbol{x}_{aw}) + \boldsymbol{D}_{r,w}\boldsymbol{w}, \tag{2.87}$$
$$\boldsymbol{y}_{aw,y} = -\boldsymbol{C}_s\boldsymbol{x}_{aw}, \tag{2.88}$$
$$\boldsymbol{y}_{aw,u} = -\boldsymbol{K}_{aw}\boldsymbol{x}_{aw}, \tag{2.89}$$
$$\boldsymbol{y}_s = \boldsymbol{C}_s\boldsymbol{x}_s \tag{2.90}$$

where $\boldsymbol{u}_s = \mathrm{sat}_u(\boldsymbol{y}_r + \boldsymbol{y}_{aw,u})$, $\boldsymbol{w}$ is the reference signal, $\boldsymbol{z}$ defines a disturbance and $\boldsymbol{x}_r$, $\boldsymbol{x}_s$ and $\boldsymbol{x}_{aw}$ are the states of the controller, plant and anti-windup system, respectively. These states can be summarized in a single vector

$$\boldsymbol{x}'_{awrk} = \begin{bmatrix}\boldsymbol{x}'_s & \boldsymbol{x}'_r & \boldsymbol{x}'_{aw}\end{bmatrix}. \tag{2.91}$$

Using a coordinate transformation, the representation of the system can be simplified which allows the minimization of the mismatch between the unlimited control loop and limited control loop with an anti-windup network. This representation is called "Mismatch-Representation". For the design of the model recovery anti-windup, i.e. the design of the anti-windup gain $\boldsymbol{K}_{aw}$, one has to pay attention for the mismatch between the limited control loop during saturation to the behavior of the unlimited control loop. Due to the linear transformation of the state variables by a regular matrix $\boldsymbol{T}$, the system can be easily analyzed. This transformation can be stated as

$$\boldsymbol{\xi}_{ms} = \boldsymbol{T}\boldsymbol{x}_{awrk} = \begin{bmatrix}\boldsymbol{\xi}_s \\ \boldsymbol{\xi}_r \\ \boldsymbol{\xi}_{aw}\end{bmatrix} = \begin{bmatrix}\mathbb{I} & 0 & \mathbb{I} \\ 0 & \mathbb{I} & 0 \\ 0 & 0 & -\mathbb{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_s \\ \boldsymbol{x}_r \\ \boldsymbol{x}_{aw}\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}_s + \boldsymbol{x}_{aw} \\ \boldsymbol{x}_r \\ -\boldsymbol{x}_{aw}\end{bmatrix} \tag{2.92}$$

where $\boldsymbol{\xi}_{ms}$ defines the mismatch states and $\boldsymbol{T} \in \mathbb{R}^{(2n_s+n_r)\times(2n_s+n_r)}$. This equation (2.92) transforms equations (2.84) - (2.90) to the mismatch representation (see figure 2.9)

$$\dot{\boldsymbol{\xi}}_s = \boldsymbol{A}_s\boldsymbol{\xi}_s + \boldsymbol{B}_{s,u}\boldsymbol{y}_r + \boldsymbol{B}_{s,z}\boldsymbol{z}, \tag{2.93}$$
$$\dot{\boldsymbol{\xi}}_r = \boldsymbol{A}_r\boldsymbol{\xi}_r + \boldsymbol{B}_{r,u}\boldsymbol{C}_s\boldsymbol{\xi}_s + \boldsymbol{B}_{r,w}\boldsymbol{w}, \tag{2.94}$$
$$\dot{\boldsymbol{\xi}}_{aw} = \boldsymbol{A}_s\boldsymbol{\xi}_{aw} + \boldsymbol{B}_{s,u}(\mathrm{sat}_u(\boldsymbol{y}_r + \boldsymbol{y}_{aw,u}) - \boldsymbol{y}_r), \tag{2.95}$$
$$\boldsymbol{y}_r = \boldsymbol{C}_r\boldsymbol{\xi}_r + \boldsymbol{D}_{r,u}\boldsymbol{C}_s\boldsymbol{\xi}_s + \boldsymbol{D}_{r,w}\boldsymbol{w}, \tag{2.96}$$
$$\boldsymbol{y}_{aw,y} = \boldsymbol{C}_s\boldsymbol{\xi}_{aw}, \tag{2.97}$$
$$\boldsymbol{y}_{aw,u} = -\boldsymbol{K}_{aw}(-\boldsymbol{\xi}_{aw}), \tag{2.98}$$
$$\boldsymbol{y}_s = \boldsymbol{C}_s(\boldsymbol{\xi}_s + \boldsymbol{\xi}_{aw}) = \boldsymbol{y}_{s,ub} + \boldsymbol{y}_{aw,y}. \tag{2.99}$$

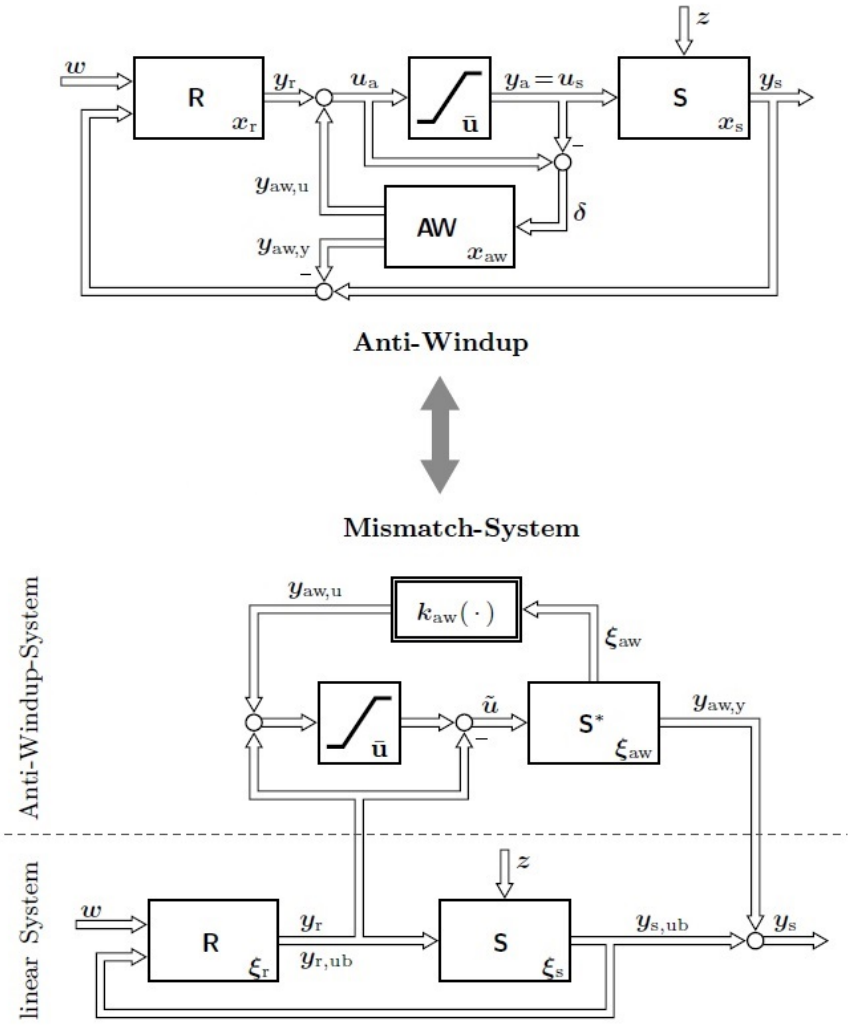

Figure 2.9: Transformation of a model recovery anti-windup control loop in general representation to the mismatch representation [15]

The advantage of this transformation or rather the mismatch representation is the decomposition of the anti-windup network in two well arranged subsystems. The first linear subsystem describes the unlimited control loop by equations (2.93), (2.94) and (2.96) with the output $\boldsymbol{y}_{s,ub}$

$$\boldsymbol{L}: \begin{cases} \dot{\boldsymbol{\xi}}_s & = \boldsymbol{A}_s\boldsymbol{\xi}_s + \boldsymbol{B}_{s,u}\boldsymbol{y}_r + \boldsymbol{B}_{s,z}\boldsymbol{z}, \\ \dot{\boldsymbol{\xi}}_r & = \boldsymbol{A}_r\boldsymbol{\xi}_r + \boldsymbol{B}_{r,u}\boldsymbol{C}_s\boldsymbol{\xi}_s + \boldsymbol{B}_{r,w}\boldsymbol{w}, \\ \boldsymbol{y}_r & = \boldsymbol{C}_r\boldsymbol{\xi}_r + \boldsymbol{D}_{r,u}\boldsymbol{C}_s\boldsymbol{\xi}_s + \boldsymbol{D}_{r,w}\boldsymbol{w}, \\ \boldsymbol{y}_{s,ub} & = \boldsymbol{C}_s\boldsymbol{\xi}_s. \end{cases} \tag{2.100}$$

The second subsystem contains the nonlinearity due to saturation and contains the anti-windup method and can be stated by equations (2.95), (2.98) and (2.97)

which defines the mismatch to the unlimited control loop

$$\boldsymbol{AWNW}: \begin{cases} \dot{\boldsymbol{\xi}}_{aw} & = \boldsymbol{A}_s \boldsymbol{\xi}_{aw} + \boldsymbol{B}_{s,u}(\mathrm{sat}_u(\boldsymbol{y}_r + \boldsymbol{y}_{aw,u}) - \boldsymbol{y}_r), \\ \boldsymbol{y}_{aw,y} & = \boldsymbol{C}_s \boldsymbol{\xi}_{aw}, \\ \boldsymbol{y}_{aw,u} & = -\boldsymbol{K}_{aw}(-\boldsymbol{\xi}_{aw}). \end{cases} \tag{2.101}$$

Since the missile is linearized at various operating points, a method called gain-scheduling is used to compute the manipulated variable. This is done by interpolation of this variable delivered by all neighboring OPs. During controller blending of adjacent OPs, anti-windup occurs even though the system is unlimited. This is due to the fact that only a fraction of each manipulated variable of the neighboring OPs is applied. The problem will be described in detail in section 3.5.

2.5 Controller Possibilities

Various controllers can be employed to stabilize a missile and guide it to their target. For the consideration of a proper controller many influences have to be taken into account. These influences depend on the aim of the missile. For example if the missile should be used to intercept objects in great heights, a DACS is needed in order to ensure high agility in the upper thin atmosphere. Such systems need a dynamic output feedback controller with a servo control [3] since the response to setpoint changes can be influenced by the prefilter and the fault response by the output feedback separately. When considering height in the lower atmosphere ($H \leq 15\,\mathrm{km}$), no DACS and hence no servo control is needed. Other properties of the missile and its mission which are important for the choice of controllers are the ability to hit static or dynamic targets, the load capacity, maximum velocities or which states of the missile can be measured. Two different kinds of controllers are necessary to control the three axes of the missiles which have been chosen for the automated controller design in this thesis.

2.5.1 State Feedback

A state feedback controller (see figure 2.10) computes the manipulated variable $\boldsymbol{u}$ by the use of a proportional controller $\boldsymbol{K}_x$ which influences the current state vector $\boldsymbol{x}$. The problem of state feedback controllers is that all states have to be measured which is not always possible in reality. For simple specifications for the controller and the states are measurable, it is a good, simple and reliable choice for the control. Since the roll angle ϕ of the missile should be zero during the flight, a state feedback controller is a good choice to counteract disturbances on the roll axis and to reach quickly $\phi = 0$.

When more states have to be measured and this is not possible, a dynamic output feedback controller should be chosen [2].

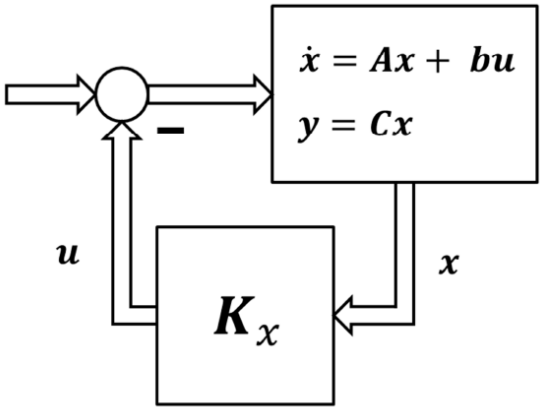

Figure 2.10: Block diagram of a state feedback controller [3]

2.5.2 Dynamic Output Feedback

A controller with a dynamic output feedback consists of a dynamic system whose inputs are only the measurable states of the plant (see figure 2.11). The order of the controller is the same as the plant's order to apply a LMI based design [10]. Due to the fact that the dynamics of the controller can be unstable, one is more free in the design compared to other controllers e.g. state feedback controller combined with an observer which has stable dynamics. A big advantage is that the poles can be placed according to the agility of the system. The transfer function of the dynamic output feedback controller is

$$\boldsymbol{G}(s) = \boldsymbol{C}_r(s\boldsymbol{I} - \boldsymbol{A}_r)^{-1}\boldsymbol{B}_r + \boldsymbol{D}_r. \tag{2.102}$$

Which can be expressed in shorthand notation as

$$\boldsymbol{G}(s) = \left(\frac{\boldsymbol{A}_r|\boldsymbol{B}_r}{\boldsymbol{C}_r|\boldsymbol{D}_r}\right). \tag{2.103}$$

For the stabilization of the pitch and yaw axis and also the execution of lateral acceleration commands, dynamic output feedback controllers will be used. The design of these controllers for the missile will be presented in section 3.3.

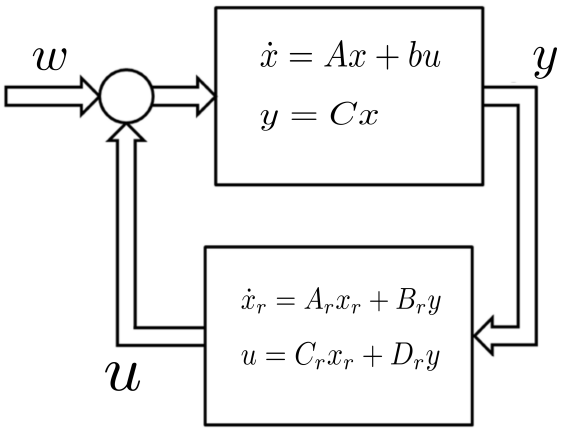

Figure 2.11: Block diagram of a dynamic output feedback controller [3]

3 Automation

This chapter is intended to present the approach to the problem of automated controller design, the whole procedure of the automation as well as some brief description of the algorithms which have been developed in order to achieve the goals which has been specified in section 1.1.
The whole automated controller design using convex optimization was mainly applied to one specific missile and small areas of the operating range but the scope of this thesis is to extend the concept to the whole flight envelope.

3.1 Procedure

In order to provide a better understanding of the automated controller design procedure, a step by step description of the procedure is presented below. The main tasks are also illustrated in flow chart 3.1.

1. Definition of Design Properties
   - Selection of the missile
   - Definition of the operating range, the grid dimension and trim variables
2. Linearization
   - Calculation of trim variables
     - According to the actuating systems and user input
     - Minimization of a cost function which adapts the trim variables in such a way that the derivatives, calculated by the Equations of Motion (EoM), of the states are approximately zero
   - Linearization of the missile at OPs or rather trim points
     - Jacobian linearization with trim variables (see section 2.2)
     - Definition of the state space matrices for the elevator, rudder and aileron
3. Controller Design at Operating Points
   - Design methods rely on a convex optimization problem (see section 2.3)
   - Defining a Pole Placement Area (PPA) where the eigenvalues are placed
     - Will be determined by the current flight conditions, i.e. the OPs
     - Analysis of good working pole placement areas at the bounds of the missile's operating range have to be determined in advance
4. Verification of Neighboring Operating Points

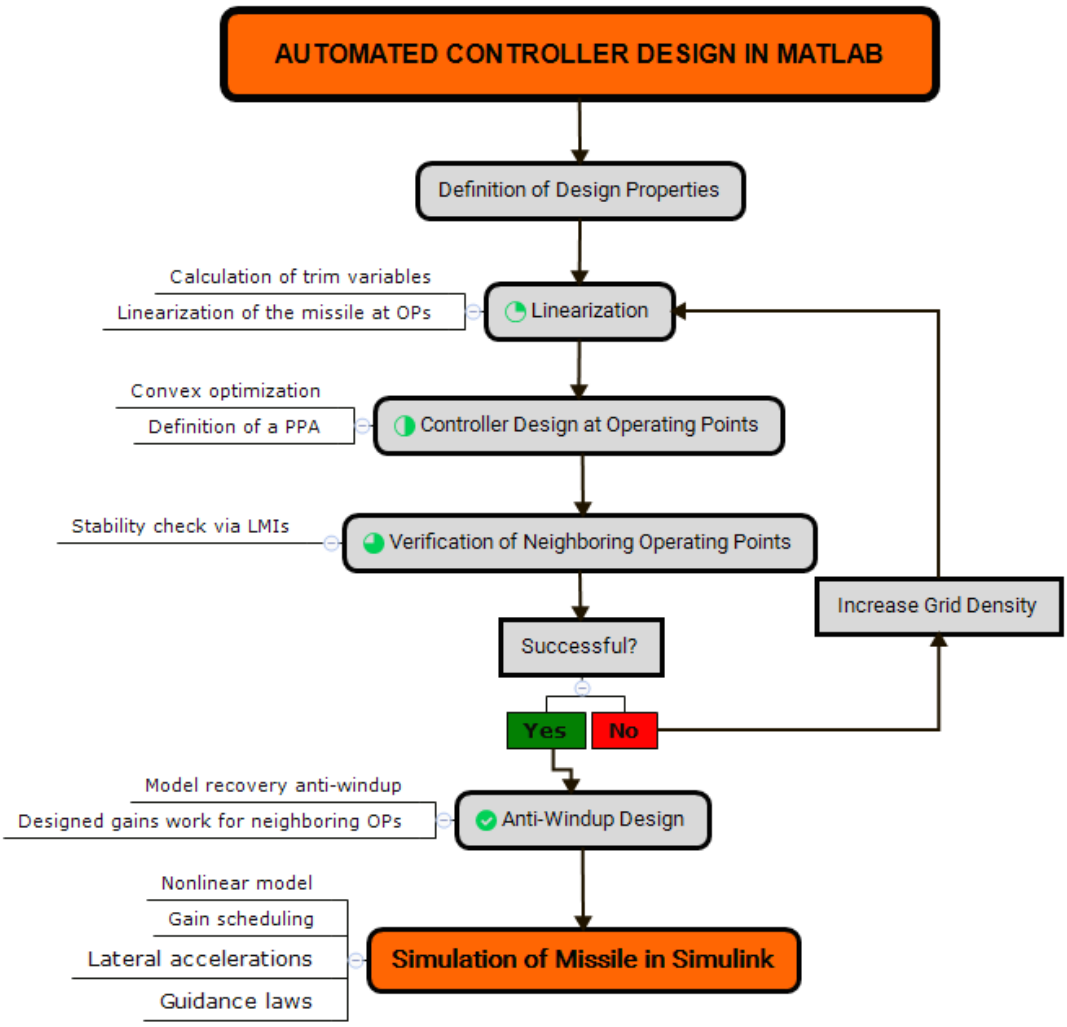

Figure 3.1: Flow chart of the whole automation process

- It will be tested via LMIs whether the controller at a specific OP stabilizes the missile until it has the conditions at the neighboring operating points
- The performance of the controller should not exceed a defined boundary
- When the verification at a specific point fails, the algorithm scales down the grid along the grid parameter (e.g. height or Mach number) where the failure occurred and starts again at point 2 linearization

5. Anti-Windup Design
   - Stability and performance criteria according to a model recovery anti-windup option presented by [13] and [15]
   - Design method rely also on a convex optimization problem and will be briefly described in section 3.6

- The designed anti-windup gains $\boldsymbol{K}_{aw}$ at each OP should also work for the neighboring operating points
  - Stability will be guaranteed by including the neighboring OPs directly in the $\boldsymbol{K}_{aw}$ design under the assumption of an affine linear parameter dependency among the linearized models at the grid points

6. Simulation of the Nonlinear Model
   - Definition of simulation properties like the scenario or initial conditions of the missile
   - Test of controllers and anti-windup network
   - Application of a gain-scheduling method which generates the manipulated variable out of the current flight condition and the related neighboring operating points
   - One Simulink model for the implementation should work for the different grid dimensions, scenarios and missiles

3.2 Existing Results

This master thesis is based on the previous work of Benedikt Bartenschlager [3]. The result of his diploma thesis "Control concept for a highly agile dual actuator interceptor" was a controller design for missiles using a DACS and/or aerodynamic actuating systems which was locally tested in small operating regions. The developed controller design achieved controllers with a good performance at specific flight conditions. The main result of the previous thesis was the identification of windup during blending and hence stability problems. The model-recovery anti-windup network was identified for the solution of these problems. The design of the anti-windup network and the verification whether adjacent OPs will be stabilized was manually executed. Hence, the validity areas of the controllers for the linear approximations of the missile were unknown.

This design was mainly used in this thesis and adapted to an automated controller design for the whole flight envelope of missiles using one or two aerodynamic actuating systems. Furthermore, the design of the anti-windup network was enhanced by another method which ensures L_2-Gain stability [13], [15]. The whole design is then applied to the particular missile and via "Gain-Scheduling" [1] stabilized and lead to the target.
Gain-Scheduling is a method used to compute the manipulated variable over the whole flight envelope. Since the missile has to be linearized over its operating range (see section 2.2), many controllers have to be designed at the trim points or rather operating points. With this method, one gets the manipulated variable determined by adjacent OPs with respect to the current flight conditions of the

missile. This method will be explained in more detail and with the help of examples illustrated in section 4.1.

3.2.1 Approach to the Automated Controller Design

The first step in getting the solution, stated in section 1.1, is familiarizing with the problem of anti-windup. In the course of this approach, a simple model was developed to demonstrate the differences between the already existing simple anti-windup method which made only use of placing the poles and the model-recovery anti-windup method described by [13] and [15] which considers $u_{\max}$ in the anti-windup network design. Afterwards, this enhanced anti-windup method was applied to the linearized z-axis of a missile, i.e. the elevator. As soon as this simulation worked, it was verified whether the designed anti-windup gains work for the "real" nonlinear missile model. For this verification, the already existing controller design [3] was used to get good working controllers at specific operating points.

In order to develop an automated controller design, we started working on the operating point grid generation for the guaranty of stabilizing the missile in the area defined by four (two-dimensional grid) or eight (three-dimensional grid) OPs. As some problems occurred which lead to a grid with an enormous number of trim points and hence controllers, a first step for the generation of irregular grids was developed.

Afterwards, the existing controller design method for the aileron, elevator and rudder was extended by an algorithm which makes it possible to design good working controllers at the predefined grid points over the whole flight envelope.

The last step of this thesis was to simulate the nonlinear missile by using the corresponding controllers and anti-windup gains to ensure a stable, robust and agile control.

In the next sections the employed design methods, algorithms and ideas for the whole automation process will be explained in detail as well as their application to the missile is presented.

3.3 Controller Design Using Convex Optimization

The controllers will be designed for the flight conditions of the various operating points. Since the behavior of the missile differs according to the current flight conditions, i.e. OP, the algorithm needs to adapt the design parameters. They should lead to more dynamical controllers when the missile has either a large velocity, low altitude or low angle of attack or to controllers which are less dynamical for conditions like high altitude, large angle of attack or small Mach number. This

section is intended to describe in detail how the controllers are designed with the requirement of using convex optimization and which changes had to be applied in order to automate this design.

The state space model of the plant in general can be expressed by

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\mathrm{diag}(\boldsymbol{T}_{Act}) \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \mathrm{diag}(\boldsymbol{T}_{Act}) \end{bmatrix} \boldsymbol{u}_c \tag{3.1}$$

where $\boldsymbol{T}_{Act}$ defines the time constants of the actuators $(1/T_{aero_i}, i = 1, ..., m)$. It is necessary to define the outputs for the feedback of the lateral acceleration controllers. The outputs are the measurable values $\boldsymbol{y}_m = [\phi, p, q, r, \xi, \eta, \zeta]'$ and the controlled values are $\boldsymbol{y}_r = [a_y, a_z]'$. The measurement vector can be expressed by

$$\boldsymbol{y} = [e \quad e_I \quad \omega \quad \boldsymbol{u}]' \tag{3.2}$$

where $e = a_{\mathrm{com}} - a$ is the control error and describes the difference of commanded and actual acceleration. Due to the added integral of the error e_I, the state space model (3.1) must be extended to

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{u}} \\ \dot{e}_I \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\mathrm{diag}(\boldsymbol{T}_{Act}) & \boldsymbol{0} \\ -\boldsymbol{c}' & -\boldsymbol{d}' & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \\ e \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \mathrm{diag}(\boldsymbol{T}_{Act}) \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u}_c + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ 1 \end{bmatrix} \boldsymbol{w}. \tag{3.3}$$

Since the axes will be separated considered, the controlled value is a scalar per axis. Therefore, $y_r = a = \boldsymbol{c}'\boldsymbol{x} + \boldsymbol{d}'\boldsymbol{u}$.

3.3.1 Dynamic Output Feedback Controller

Figure 3.2 illustrates a dynamic output feedback controller where

$$\boldsymbol{P}(s) : \begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{B}_w\boldsymbol{w} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}_w\boldsymbol{w} \\ \boldsymbol{z} &= \boldsymbol{C}_z\boldsymbol{x} + \boldsymbol{D}_{zw}\boldsymbol{w} + \boldsymbol{D}_z\boldsymbol{u} \end{cases} \tag{3.4}$$

defines the plant in general form according to [5]. In this equation, $\boldsymbol{w}$ defines the reference variable and $\boldsymbol{z}$ is used for the control accuracy and performance.
Applying the extended state space representation (3.3) exemplary to the z-axis yields

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{A}_{aq} & \boldsymbol{B}_{aq} & \boldsymbol{0} \\ \boldsymbol{0} & -\mathrm{diag}(\boldsymbol{T}_{Act}) & \boldsymbol{0} \\ -\boldsymbol{C}'_{aq} & -\boldsymbol{D}_{aq} & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{0} \\ \mathrm{diag}(\boldsymbol{T}_{Act}) \\ 0 \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ 1 \end{bmatrix} \boldsymbol{w}, \tag{3.5}$$

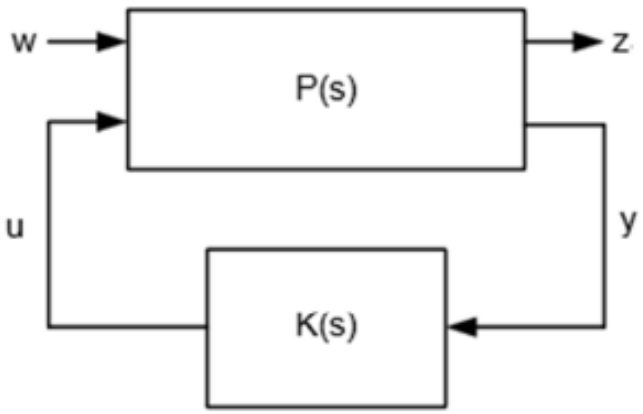

Figure 3.2: Block diagram of a dynamic output feedback controller [3]

$$\boldsymbol{z} = \begin{bmatrix} -\boldsymbol{C}'_{aq} \cdot p_1 & -\boldsymbol{D}_{aq} \cdot p_1 & 0 \\ \mathbf{0} & \operatorname{diag}(\boldsymbol{p}_2 \;\; \boldsymbol{p}_2) & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} p_1 \\ 0 \end{bmatrix} \boldsymbol{w} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \boldsymbol{u}, \tag{3.6}$$

$$\boldsymbol{y} = \begin{bmatrix} -C_{aq,1} & -C_{aq,2} & -\boldsymbol{D}_{aq} & 0 \\ 0 & 0 & \mathbf{0} & 1 \\ 0 & 1 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{0} & \mathbb{I} & \mathbf{0} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \mathbf{0} \end{bmatrix} \boldsymbol{w} \tag{3.7}$$

where p_1 and p_2 are weighting factors for the control error and the deflection of the elevator with respect to the H_∞-norm. Good working factors are $p_1 = 0.02$ and $p_2 = 3$. The dimension of the values make sense since they influence the acceleration and control surface deflection respectively. The feedback consists of the control error e, its integration e_I, the pitch rate q and the elevator at the front η_f and back η_b.
The controller $\boldsymbol{K}(s)$ can be seen in figure 3.2 and is mathematically defined by

$$\boldsymbol{K}(s) : \begin{cases} \dot{\boldsymbol{x}}_r &= \boldsymbol{A}_r \boldsymbol{x}_r + \boldsymbol{B}_r \boldsymbol{y} \\ \boldsymbol{u} &= \boldsymbol{C}_r \boldsymbol{x}_r + \boldsymbol{D}_r \boldsymbol{y} \end{cases} \tag{3.8}$$

The whole design of the dynamic output feedback controller follows [5], [2] and [10].
The closed loop system is then obtained by equations (3.4) and (3.8)

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{x}}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} + \boldsymbol{B}\boldsymbol{D}_r\boldsymbol{C} & \boldsymbol{B}\boldsymbol{C}_r \\ \boldsymbol{B}_r\boldsymbol{C} & \boldsymbol{A}_r \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_r \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}\boldsymbol{D}_r\boldsymbol{D}_w + \boldsymbol{B}_w \\ \boldsymbol{B}_r\boldsymbol{D}_w \end{bmatrix} \cdot \boldsymbol{w}, \tag{3.9}$$

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{C}_z + \boldsymbol{D}_z\boldsymbol{D}_r\boldsymbol{C} & \boldsymbol{D}_z\boldsymbol{C}_r \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_r \end{bmatrix} + \begin{bmatrix} \boldsymbol{D}_{zw} + \boldsymbol{D}_z\boldsymbol{D}_r\boldsymbol{D}_w \end{bmatrix} \cdot \boldsymbol{w}. \tag{3.10}$$

A shorthand notation of these equations (3.9) and (3.10) can be stated as

$$\left(\begin{array}{cc|c} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{D}_r\boldsymbol{C} & \boldsymbol{B}\boldsymbol{C}_r & \boldsymbol{B}\boldsymbol{D}_r\boldsymbol{D}_w+\boldsymbol{B}_w \\ \boldsymbol{B}_r\boldsymbol{C} & \boldsymbol{A}_r & \boldsymbol{B}_r\boldsymbol{D}_w \\ \hline \boldsymbol{C}_j+\boldsymbol{E}_j\boldsymbol{D}_r\boldsymbol{C} & \boldsymbol{E}_j\boldsymbol{C}_r & \boldsymbol{D}_j+\boldsymbol{E}_j\boldsymbol{D}_r\boldsymbol{D}_w \end{array}\right) = \left(\begin{array}{c|c} \boldsymbol{A}_{cl} & \boldsymbol{B}_i \\ \hline \boldsymbol{C}_i & \boldsymbol{D}_i \end{array}\right) \tag{3.11}$$

where $\boldsymbol{C}_j = \boldsymbol{C}_z$, $\boldsymbol{D}_j = \boldsymbol{D}_{zw}$, $\boldsymbol{E}_j = \boldsymbol{D}_z$ and index j defines the row of the according matrix. The Transfer Function (TF) from the reference value w_i to the performance output z_i of the closed loop system is

$$\boldsymbol{T}_{wz,i}(s) = \boldsymbol{D}_i + \boldsymbol{C}_i(s\mathbb{I} - \boldsymbol{A}_{cl})^{-1}\boldsymbol{B}_i. \tag{3.12}$$

The H_∞-norm of the transfer function (3.12) is defined by [20]

$$||\boldsymbol{G}(s)||_\infty \hat{=} \sup\{||\boldsymbol{G}(s)|| \; |\{\text{Re}\}\; s > 0\}. \tag{3.13}$$

If the H_∞-norm of the TF is minimal, the output energy is minimized, i.e. the control error caused by disturbances and measurement noise is minimized [18]. The H_∞-norm is then defined as

$$||\boldsymbol{T}_{wz,i}||_\infty < \gamma \tag{3.14}$$

and can be interpreted as a rejection of disturbances performance and enforces robustness. This criteria ensures stability of the closed loop for perturbations $\boldsymbol{w}_p = \Delta\boldsymbol{z}_p$ where the incremental gains $\Delta \leq \frac{1}{\gamma}$ [5]. The goal of the controller design using convex optimization is to minimize $\gamma > 0$. Since the controller should stabilize the plant, the closed loop has to admit a quadratic Lyapunov function

$$\boldsymbol{V}(\boldsymbol{x}_{cl}) = \boldsymbol{x}_{cl}'\boldsymbol{P}\boldsymbol{x}_{cl}, \quad \boldsymbol{P} \succ 0 \qquad \text{where } \boldsymbol{x}_{cl} = [\boldsymbol{x}'\ \boldsymbol{x}_r']' \tag{3.15}$$

such that

$$\boldsymbol{A}_{cl}'\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_{cl} \prec 0, \quad \boldsymbol{P} \succ 0. \tag{3.16}$$

According to the Bounded Real Lemma [5], system $\boldsymbol{A}_{cl}$ is stable and the H_∞-norm follows the constraint of being smaller than γ if and only if there exists a symmetric positive definite matrix $\boldsymbol{P}$ [5] defined by the LMIs

$$\begin{bmatrix} \boldsymbol{A}_{cl}'\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_{cl} & \boldsymbol{P}\boldsymbol{B}_j & \boldsymbol{C}_j' \\ \boldsymbol{B}_j'\boldsymbol{P} & -\gamma\mathbb{I} & \boldsymbol{D}_j' \\ \boldsymbol{C}_j & \boldsymbol{D}_j & -\gamma\mathbb{I} \end{bmatrix} < 0, \quad \boldsymbol{P} > 0. \tag{3.17}$$

Due to $\boldsymbol{A}_{cl}$ as stated in equation (3.11), the previous equation (3.17) defines no LMI for which reason a transformation by the new matrices

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{N} \\ \boldsymbol{N}' & \# \end{bmatrix}, \quad \boldsymbol{P}^{-1} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{M} \\ \boldsymbol{M}' & \# \end{bmatrix}, \quad \boldsymbol{P}\boldsymbol{P}^{-1} = \mathbb{I} \tag{3.18}$$

and hence new variables $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{M}$ and $\boldsymbol{N}$ has to be executed. This transformation has to be done in order to eliminate the nonlinearties which occur due to the product of the wanted variables $\boldsymbol{A}_{cl}\boldsymbol{P}$ [5]. Matrices denoted by # in equation (3.18) are not relevant and are not further considered. From $\boldsymbol{P}\boldsymbol{P}^{-1} = \mathbb{I}$ we infer that

$$\boldsymbol{P}\boldsymbol{P}^{-1} = \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{N} \\ \boldsymbol{N}' & \# \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{X} & \boldsymbol{M} \\ \boldsymbol{M}' & \# \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \boldsymbol{0} \\ \boldsymbol{0} & \mathbb{I} \end{bmatrix} \tag{3.19}$$

and therefore the constraint on $\boldsymbol{N}$ and $\boldsymbol{M}$

$$\boldsymbol{Y}\boldsymbol{X} + \boldsymbol{N}\boldsymbol{M}' = \mathbb{I} \tag{3.20}$$

holds. Furthermore, matrices

$$\boldsymbol{\pi}_1 = \begin{bmatrix} \boldsymbol{X} & \mathbb{I} \\ \boldsymbol{M}' & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{\pi}_2 = \begin{bmatrix} \mathbb{I} & \boldsymbol{Y} \\ \boldsymbol{0} & \boldsymbol{N}' \end{bmatrix}, \quad \boldsymbol{P}\boldsymbol{\pi}_1 = \boldsymbol{\pi}_2 \tag{3.21}$$

will be defined as well as new transformed controller parameters

$$\hat{\boldsymbol{A}} = \boldsymbol{N}\boldsymbol{A}_r\boldsymbol{M}' + \boldsymbol{N}\boldsymbol{B}_r\boldsymbol{C}\boldsymbol{X} + \boldsymbol{Y}\boldsymbol{B}\boldsymbol{C}_r\boldsymbol{M}' + \boldsymbol{Y}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{D}_r\boldsymbol{C})\boldsymbol{X}, \tag{3.22}$$

$$\hat{\boldsymbol{B}} = \boldsymbol{N}\boldsymbol{B}_r + \boldsymbol{Y}\boldsymbol{B}\boldsymbol{D}_r, \tag{3.23}$$

$$\hat{\boldsymbol{C}} = \boldsymbol{C}_r\boldsymbol{M}' + \boldsymbol{D}_r\boldsymbol{C}\boldsymbol{X}, \tag{3.24}$$

$$\hat{\boldsymbol{D}} = \boldsymbol{D}_r. \tag{3.25}$$

To eliminate the nonlinear parts of equation (3.13), the matrix inequality (3.17) will be transformed with the help of $\mathrm{diag}(\boldsymbol{\pi}_1, \mathbb{I}, \mathbb{I})$. Performing this transformation and using the new controller parameters stated by equations (3.22) - (3.25) yields the LMI problem of the H_∞-norm

$$\begin{bmatrix} \boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{A}'\boldsymbol{B}\hat{\boldsymbol{C}} + (\boldsymbol{B}\hat{\boldsymbol{C}})' & \hat{\boldsymbol{A}}' + (\boldsymbol{A} + \boldsymbol{B}\hat{\boldsymbol{D}}\boldsymbol{C}) & * & * \\ (\hat{\boldsymbol{A}}' + (\boldsymbol{A} + \boldsymbol{B}\hat{\boldsymbol{D}}\boldsymbol{C}))' & \boldsymbol{A}'\boldsymbol{Y} + \boldsymbol{Y}\boldsymbol{A} + \hat{\boldsymbol{B}}\boldsymbol{C} + (\hat{\boldsymbol{B}}\boldsymbol{C})' & * & * \\ (\boldsymbol{B}_w + \boldsymbol{B}\hat{\boldsymbol{D}}\boldsymbol{D}_w)' & (\boldsymbol{Y}\boldsymbol{B}_j + \hat{\boldsymbol{B}}\boldsymbol{D}_w)' & -\gamma\mathbb{I} & * \\ \boldsymbol{C}_j\boldsymbol{X} + \boldsymbol{E}_j\hat{\boldsymbol{C}} & \boldsymbol{C}_j + \boldsymbol{E}_j\hat{\boldsymbol{D}}\boldsymbol{C} & \boldsymbol{D}_j + \boldsymbol{E}_j\hat{\boldsymbol{D}}\boldsymbol{D}_w & -\gamma\mathbb{I} \end{bmatrix} < 0 \tag{3.26}$$

where $*$ replaces blocks which are readily inferred by symmetry. The symmetric matrix $\boldsymbol{P}$ has to be positive definite which leads to the constraint

$$\begin{bmatrix} \boldsymbol{X} & \mathbb{I} \\ \mathbb{I} & \boldsymbol{Y} \end{bmatrix} > 0 \tag{3.27}$$

which has to be fulfilled as well. The Lyapunov function (3.15) ensures that the eigenvalues of the closed loop are in the left-half complex plane, i.e. ensures stability. As already described in section 2.5, the dynamic output feedback controller has the big advantage that the poles of the closed loop system can be placed in a PPA. Such an area is described by a convex set and provides the possibility of a systematic design of the controller's dynamical behavior. This convex set or rather LMI region is defined by an upper and lower limit as well as by the imaginary to real ratio and can be seen in figure 3.3.

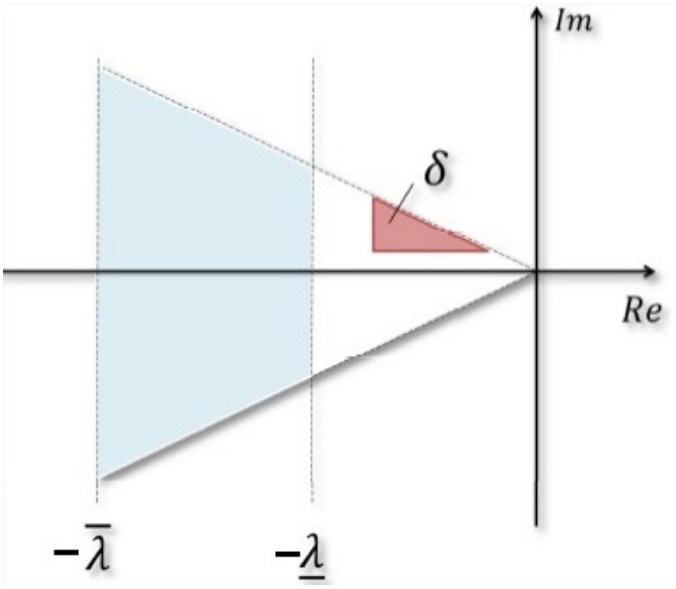

Figure 3.3: Pole placement area in the complex left-half plane [3]

The LMI constraints can be determined by transforming

$$[\boldsymbol{l}_{ij}\boldsymbol{P} + \boldsymbol{m}_{ij}\boldsymbol{A}'\boldsymbol{P} + \boldsymbol{m}_{ij}\boldsymbol{P}\boldsymbol{A}]_{ij} < 0, \quad \boldsymbol{P} > 0 \tag{3.28}$$

with $\mathrm{diag}(\boldsymbol{\pi}_1, \boldsymbol{\pi}_1, \boldsymbol{\pi}_1)$.
The LMI where the lower limit $\underline{\lambda}$ of the PPA is defined can be stated as

$$\boldsymbol{\sigma}' + \boldsymbol{\sigma} + 2 \cdot \underline{\lambda} \cdot \begin{bmatrix} \boldsymbol{X} & \mathbb{I} \\ \mathbb{I} & \boldsymbol{Y} \end{bmatrix} < 0. \tag{3.29}$$

The upper limit $\overline{\lambda}$ is ensured by

$$\boldsymbol{\sigma}' + \boldsymbol{\sigma} + 2 \cdot \overline{\lambda} \cdot \begin{bmatrix} \boldsymbol{X} & \mathbb{I} \\ \mathbb{I} & \boldsymbol{Y} \end{bmatrix} > 0. \tag{3.30}$$

The imaginary to real ratio $\delta = \frac{\mathrm{Im}\{\lambda_i\}}{\mathrm{Re}\{\lambda_i\}}$ will set the gradient of the convex set by

$$\begin{bmatrix} \delta(\boldsymbol{\sigma}' + \boldsymbol{\sigma}) & \boldsymbol{\sigma}' - \boldsymbol{\sigma} \\ -\boldsymbol{\sigma}' + \boldsymbol{\sigma} & \delta(\boldsymbol{\sigma}' + \boldsymbol{\sigma}) \end{bmatrix} < 0 \tag{3.31}$$

where

$$\boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{AX} + \boldsymbol{B}\hat{\boldsymbol{C}} & \boldsymbol{A} + \boldsymbol{B}\hat{\boldsymbol{D}}\boldsymbol{C} \\ \hat{\boldsymbol{A}} & \boldsymbol{YA} + \hat{\boldsymbol{B}}\boldsymbol{C} \end{bmatrix} \tag{3.32}$$

holds. Now, the whole convex optimization problem can be stated as

$$\begin{array}{ll} \text{minimize} & \sum \gamma_i \\ \text{subject to} & \gamma_i > 0 \text{ and (3.26), (3.27), (3.29), (3.30), (3.31).} \end{array} \tag{3.33}$$

The desired parameters are $\gamma_i, \boldsymbol{X}, \boldsymbol{Y}, \hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{D}}$. In order to derive the controller parameters, one has to redo the transformation. To this end, the QR-decomposition of

$$\boldsymbol{NM}' = \mathbb{I} - \boldsymbol{XY} \tag{3.34}$$

is applied and yields matrices $\boldsymbol{N}$ and $\boldsymbol{M}$. Now, the controller matrices can be determined by

$$\boldsymbol{D}_r = \hat{\boldsymbol{D}}, \tag{3.35}$$
$$\boldsymbol{C}_r = (\hat{\boldsymbol{C}} - \boldsymbol{D}_r\boldsymbol{CX})\boldsymbol{M}^{-'}, \tag{3.36}$$
$$\boldsymbol{B}_r = \boldsymbol{N}^{-1}(\hat{\boldsymbol{B}} - \boldsymbol{YBD}_r), \tag{3.37}$$
$$\boldsymbol{A}_r = \boldsymbol{N}^{-1}(\hat{\boldsymbol{A}} + \boldsymbol{NB}_r\boldsymbol{CX} + \boldsymbol{YBC}_r\boldsymbol{M}' - \boldsymbol{Y}(\boldsymbol{A} + \boldsymbol{BD}_r\boldsymbol{C})\boldsymbol{X})\boldsymbol{M}^{-'}. \tag{3.38}$$

The controller

$$\boldsymbol{K}(s) = \boldsymbol{D}_r + \boldsymbol{C}_r(s\mathbb{I} - \boldsymbol{A}_r)^{-1}\boldsymbol{B}_r \tag{3.39}$$

places the poles in the predefined convex set which is constrained by the LMIs (3.29), (3.30), (3.31) for $\underline{\lambda}$, $\overline{\lambda}$ and δ respectively under the requirement of $||\boldsymbol{T}_{wz,i}||_\infty < \gamma_i$.

3.3.2 State Feedback Controller

Consider again figure 2.10 for the state feedback controller. This controller is used to control the roll angle ϕ to zero. Its design consists also the optimization of the output energy. The system for the state feedback controller is defined as

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \tag{3.40}$$
$$\boldsymbol{u} = \boldsymbol{Kx} \tag{3.41}$$

and therefore the closed loop

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}) \cdot \boldsymbol{x}. \tag{3.42}$$

The state space representation of the aileron can be written as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{A}_{pp} & \boldsymbol{B}_{pp} \\ \boldsymbol{0} & -\mathrm{diag}(\boldsymbol{T}_{Act}) \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{0} \\ \mathrm{diag}(\boldsymbol{T}_{Act}) \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ 1 \end{bmatrix} \boldsymbol{w}, \tag{3.43}$$

$$\boldsymbol{y} = \boldsymbol{x}. \tag{3.44}$$

According to the LMIs (2.49) and (2.50) presented in section 2.3.1, the statement

$$(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})'\boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}) < 0, \quad \boldsymbol{P} > 0 \tag{3.45}$$

has to hold. Since products of the desired variables $\boldsymbol{P}$ and $\boldsymbol{K}$ occur in statement (3.45), you have to perform substitution like described in section 2.3.1.4. Then we obtain the LMIs

$$\boldsymbol{Q}\boldsymbol{A}' + \boldsymbol{A}\boldsymbol{Q} + \boldsymbol{B}\boldsymbol{W} + \boldsymbol{W}'\boldsymbol{B}' < 0, \quad \boldsymbol{Q} > 0 \tag{3.46}$$

which can be used to determine a stabilizing controller $\boldsymbol{K} = \boldsymbol{W}\boldsymbol{Q}^{-1}$. The output energy can be minimized by means of equations (2.75) and (2.76) as described in section 2.3.2.5. This leads to the LMIs

$$\begin{bmatrix} \boldsymbol{Q}\boldsymbol{A}' + \boldsymbol{A}\boldsymbol{Q} + \boldsymbol{B}\boldsymbol{W} + \boldsymbol{W}'\boldsymbol{B}' & \boldsymbol{Q}\boldsymbol{C}_{zj} \\ \boldsymbol{C}_{zj}'\boldsymbol{Q} & -\gamma_j \mathbb{I} \end{bmatrix} < 0, \quad \boldsymbol{Q} > 0 \tag{3.47}$$

where $\boldsymbol{C}_{zj}$ is the j-th row of $\boldsymbol{C}_z$ and $\boldsymbol{z} = \boldsymbol{C}_z\boldsymbol{x}$ is used for the weighting of the output energy according to the states. A pole placement can be performed for the state feedback controller as described in the previous section. Applying equation (3.28) to the state feedback controller design, the design limits are enforced by

$$\begin{bmatrix} \delta(\boldsymbol{\sigma}' + \boldsymbol{\sigma}) & \boldsymbol{\sigma}' - \boldsymbol{\sigma} \\ -\boldsymbol{\sigma}' + \boldsymbol{\sigma} & \delta(\boldsymbol{\sigma}'\boldsymbol{\sigma}) \end{bmatrix} < 0, \tag{3.48}$$

$$\boldsymbol{\sigma}' + \boldsymbol{\sigma} + 2\underline{\lambda}\boldsymbol{Q} < 0, \tag{3.49}$$

$$\boldsymbol{\sigma}' + \boldsymbol{\sigma} + 2\overline{\lambda}\boldsymbol{Q} > 0, \tag{3.50}$$

where $\boldsymbol{\sigma} = \boldsymbol{A}\boldsymbol{Q} + \boldsymbol{B}\boldsymbol{W}$. Now, the whole convex optimization problem is

$$\begin{array}{ll} \text{minimize} & \sum \gamma_j, \\ \text{subject to} & \gamma_j > 0 \text{ and (3.47), (3.48), (3.49), (3.50)} \end{array} \tag{3.51}$$

where the desired variables are $\boldsymbol{Q}$, $\boldsymbol{W}$ and γ_j.

3.3.3 Automation

In order to automate this controller design, the design parameters have to be automatically adapted according to the flight conditions of the missile, i.e. the OP, where the controller should be designed. This can be done by an interpolation of the limits over the whole flight envelope. Consider the grid bounds stated in table 3.1 and the according grid illustrated in figure 3.4.

Table 3.1: User defined bounds of the operating range for an air defense missile using two aerodynamic actuating systems

Grid Parameter	Minimum	Maximum	Dimension
Mach number $[-]$	0.6	5	1
Height $[\mathrm{m}]$	0	10,000	2
Angle of Attack $[°]$	0	20	3

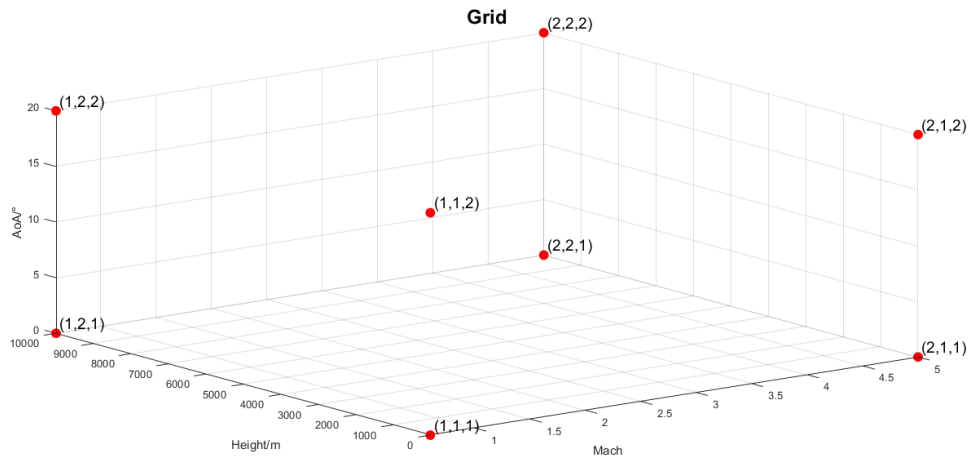

Figure 3.4: Grid bounds for an air defense missile using two aerodynamic actuating systems

One can now define the design parameters $\underline{\lambda}$, $\overline{\lambda}$ and δ for the dynamic output feedback controller and for the state feedback controller. This is exemplary shown for the dynamics of a missile's elevator in the following table 3.2.
These limits have to be determined by the user in advance. This can be done by a cautious approach. Recommended is to perform the design grid point by grid point and simulate the system with defined lateral accelerations to get the best behavior. It can be quickly seen whether the limit works or not. Either errors during the design occur or a destabilization and therefore an abort of the simulation can occur. The probable pole placement area can be restricted in advance considering the dynamical behavior and hence the agility of the missile at the grid bound where the design limit should be valid. As one can clearly see in table 3.2, the PPA and so the eigenvalues for the controller at OP (2,1,1) will be fastest from the imaginary axis in the left-half complex plane. This is due to the fact that the

Table 3.2: Design Limits for an air defense missile using two aerodynamic actuating systems at its grid bounds according to figure 3.4

Operating Points	Lower Limit	Upper Limit
(1,1,1) $\Rightarrow$ λ_{111}	10	25
(2,1,1) $\Rightarrow$ λ_{211}	20	70
(1,2,1) $\Rightarrow$ λ_{121}	6	15
(2,2,1) $\Rightarrow$ λ_{221}	10	30
(1,1,2) $\Rightarrow$ λ_{112}	6	10
(2,1,2) $\Rightarrow$ λ_{212}	10	25
(1,2,2) $\Rightarrow$ λ_{122}	3	8
(2,2,2) $\Rightarrow$ λ_{222}	8	16

missile will have its most dynamical behavior at this point. Good working PPAs for high agility is between 20 to 100 on the real axis and 3 to 15 for expected less agile behavior. These areas depend mostly on the missile's dynamical behavior.
After the determination of the design limits for the PPA at the grid bounds, the values will be interpolated according to the OP which is situated within the operating range. The interpolation formula is defined as

$$\lambda = \frac{\sum_{i=1}^{p} d_{gd} \cdot \lambda_i}{\sum_{i=1}^{p} d_{gd}} \tag{3.52}$$

where index gd is the Grid Dimension (GD), p is the number of neighboring OPs, i.e. 4 for two dimensions and 8 for a three-dimensional grid. The dimensionless parameter d is defined as $0 \leq d \leq 1$ and describes the affiliation to a specific point and hence the weighting of a value. This parameter d_{gd} can be calculated by

$$d_{gd} = 1 - \frac{OP - GB_{\min}}{GB_{\max} - GB_{\min}} \tag{3.53}$$

where OP defines the current operating point and GB the Grid Bound (GB). It describes for example the weighting along the Mach number of the design limit at GB (1,1,1) for $M = 2$. The design parameter for a GD can be calculated by

$$\lambda_{gd} = d_{gd}\lambda_1 + (1 - d_{gd})\lambda_2. \tag{3.54}$$

Now, consider an OP at $M = 3$, $H = 4\,\text{km}$ and $AoA = 15\,^\circ$. The upper limit will be subsequent exemplary calculated for this point by using the limits of the grid bounds stated in table 3.2. First, the affiliation parameters with respect to point (1,1,1) are determined. We obtain

$$d_M = 1 - \frac{3 - 0.6}{5 - 0.6} = 0.\overline{45}, \tag{3.55}$$

$$d_H = 1 - \frac{4\,\text{km} - 0\,\text{km}}{10\,\text{km} - 0\,\text{km}} = 0.6, \tag{3.56}$$

$$d_{AoA} = 1 - \frac{15\,^\circ - 0\,^\circ}{20\,^\circ - 0\,^\circ} = 0.25. \tag{3.57}$$

The upper limit of each dimension will be calculated and afterwards added to get the overall upper limit for the OP. The order does not matter. We can start by calculating the upper limit along the height at the different Mach and angle of attack points. For clarification memorize again figure 3.4. The limits will be calculated for the OP pairs shown in table 3.3.

Table 3.3: Point pairs over which the upper limit for the pole placement area will be calculated

Case	Height Point Pairs	Mach	Angle of Attack
1	[(1,1,1); (1,2,1)]	$M = 0.6$	$0\,^\circ$
2	[(2,1,1); (2,2,1)]	$M = 5$	$0\,^\circ$
3	[(1,1,2); (1,2,2)]	$M = 0.6$	$20\,^\circ$
4	[(2,1,2); (2,2,2)]	$M = 5$	$20\,^\circ$

The limits can the be calculated by

$$\lambda_1 = d_H\lambda_{111} + (1 - d_H)\lambda_{121} = 0.6 \cdot 25 + (1 - 0.6) \cdot 15 = 21, \tag{3.58}$$
$$\lambda_2 = d_H\lambda_{211} + (1 - d_H)\lambda_{221} = 0.6 \cdot 70 + (1 - 0.6) \cdot 30 = 54, \tag{3.59}$$
$$\lambda_3 = d_H\lambda_{112} + (1 - d_H)\lambda_{122} = 0.6 \cdot 10 + (1 - 0.6) \cdot 8 = 9.2, \tag{3.60}$$
$$\lambda_4 = d_H\lambda_{212} + (1 - d_H)\lambda_{222} = 0.6 \cdot 25 + (1 - 0.6) \cdot 16 = 21.4. \tag{3.61}$$
$$\tag{3.62}$$

Afterwards, these limits have to be interpolated over the other dimensions Mach number and angle of attack. First over the Mach number

$$\lambda_a = d_M\lambda_1 + (1 - d_M)\lambda_2 = 0.\overline{45} \cdot 21 + (1 - 0.\overline{45}) \cdot 54 = 39.15, \tag{3.63}$$
$$\lambda_b = d_M\lambda_3 + (1 - d_M)\lambda_4 = 0.\overline{45} \cdot 9.2 + (1 - 0.\overline{45}) \cdot 21.4 = 15.91. \tag{3.64}$$
$$\tag{3.65}$$

And finally along the angle of attack

$$\overline{\lambda} = d_{AoA}\lambda_a + (1 - d_{AoA})\lambda_b = 0.25 \cdot 39.15 + (1 - 0.25) \cdot 15.91 = 21.72. \tag{3.66}$$

This is done for every OP and automatizes therefore the design of the controllers according to their dynamical behavior over the missile's operating range.

3.4 Operating Point Grid Generation

This section is intended to describe the properties of the operating point grid and how the automated grid generation works in detail.
The operating point grid is essential for a stable flight of the missile during the whole mission. Its density depends on the performance of the controllers as well

as the behavior of the missile at specific velocities, height or angles of attack. If we have a poor controller performance and hence a small validity area, the grid gets denser as long as the validity area includes also all adjacent operating points. The density of the grid is proportional to the computational time. Therefore, the aim of the grid generation is to be as dense as necessary and as good as possible. One option is to define regular grids (see figure 3.5) or irregular grids (see figure 3.6).

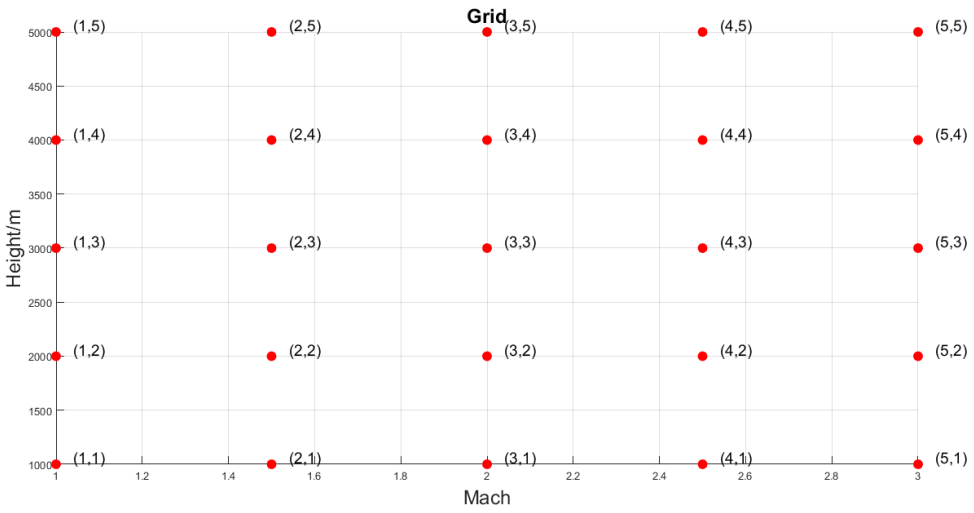

Figure 3.5: Regular grid

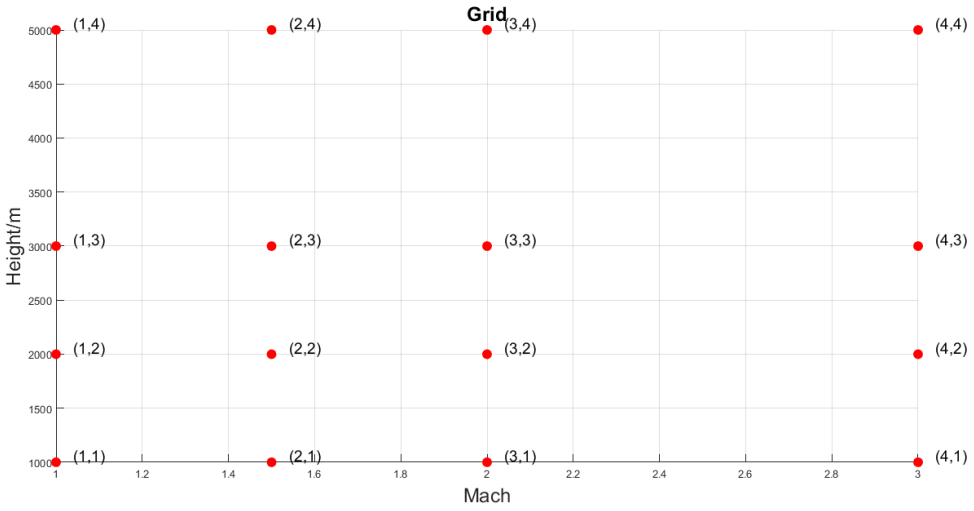

Figure 3.6: Irregular grid with less points than the regular grid from figure 3.5 but the same coverage of operating range

Regular grids have the advantage of being easy to implement but the disadvantage of much more operating points than irregular grids. Irregular grids are more

complex to implement but can save a lot of time needed for the controller design. In the beginning of this thesis, exclusively regular grids have been developed but in the end more complex irregular grids have been generated in order to overcome some problems and to reduce computing time.

3.4.1 Properties

It is necessary to define in advance an initial grid. The dimension is defined by the Mach number and height (two-dimensional - see figure 3.7) and by the angle of attack (three-dimensional - see figure 3.8). Since it can be very time consuming to design controllers over three grid dimensions due to the high number of OPs, this can be optional considered in order to get a better performance of the controllers.

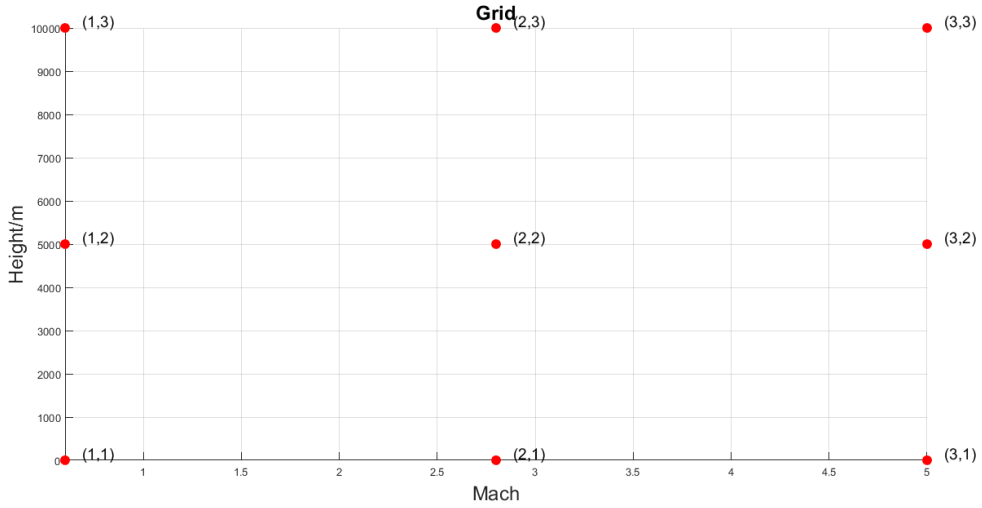

Figure 3.7: Very coarse two-dimensional grid of a missile's flight envelope

Furthermore, the operating range of the missile has to be defined in advance (see table 3.4 where the whole operating range is defined).

Table 3.4: Exemplary operating range for an air defense missile using two aerodynamic actuating systems

Grid Parameter	Minimum	Maximum
Mach number [−]	0.6	5
Height [m]	0	10,000
Angle of Attack [°]	0	20

It is furthermore possible to define in advance iteration steps between the maximum and minimum values of a grid parameter. This is helpful to get faster a working grid and a clue about the performance of the missile. Contrariwise is the disadvantage that the grid is at the beginning denser defined than maybe actually needed since defining a specific distance between the OPs along a grid parameter

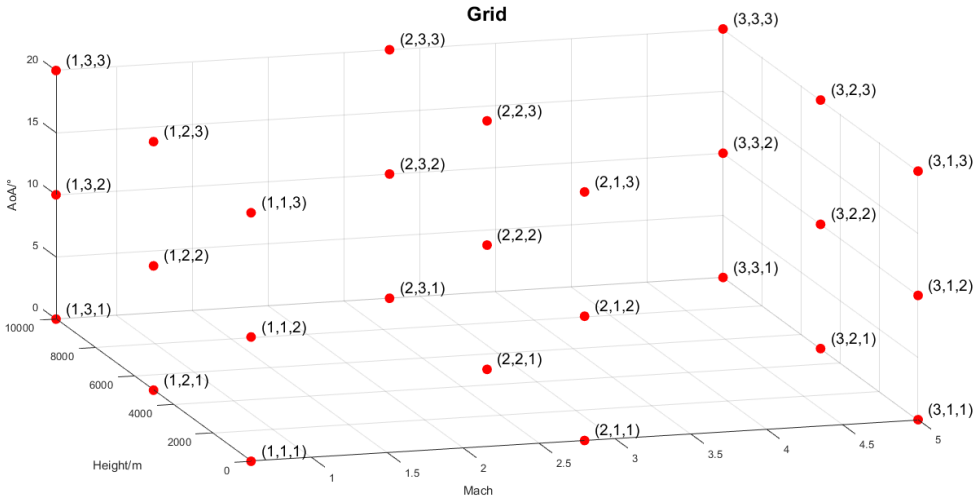

Figure 3.8: Very coarse three-dimensional grid of a missile's flight envelope

is like trying a shot in the dark.
Therefore, it is recommended to define only the bounds like the example illustrated in table 3.4.

3.4.2 Algorithm

In this section, the basic idea behind the grid generation will be explained.
In the beginning, the user input which has been explained in the previous section 3.4.1 will be processed since various options are possible due to the pretension of generic design and the automated adaption of the grid and hence continuously changing parameters. Afterwards, the points where the missile should be linearized and trimmed and controllers should be designed (section 3.3) is defined. After the controller design, their validity area will be verified (section 3.4.3) with the help of LMIs. When a problem occurs, all further steps will be skipped and the grid will get denser along the grid parameter where the scope of the validity area has been tested. The procedure starts again (see figure 3.1 again).

3.4.3 Verification of Validity Area

The verification of the validity area via LMIs is a good option for an automated design. Consider figure 3.9 for the verification algorithm.
The aim is to prove stability of the missile in the whole validity area of a controller like shown in figure 3.10.
First, the adjacent Mach point will be tested for stability, afterwards the second dimension, height, and the last verification step is the OP where both Mach number and height will be changed by either increasing or decreasing the value. This is done by checking for the existence of a positive definite matrix $\boldsymbol{P}$ when controller at (1,2) is applied to either (2,2), (1,1) or (2,1). When this is shown, the controller stabilizes the missile in the whole rectangle defined by these adjacent four OPs. The validity

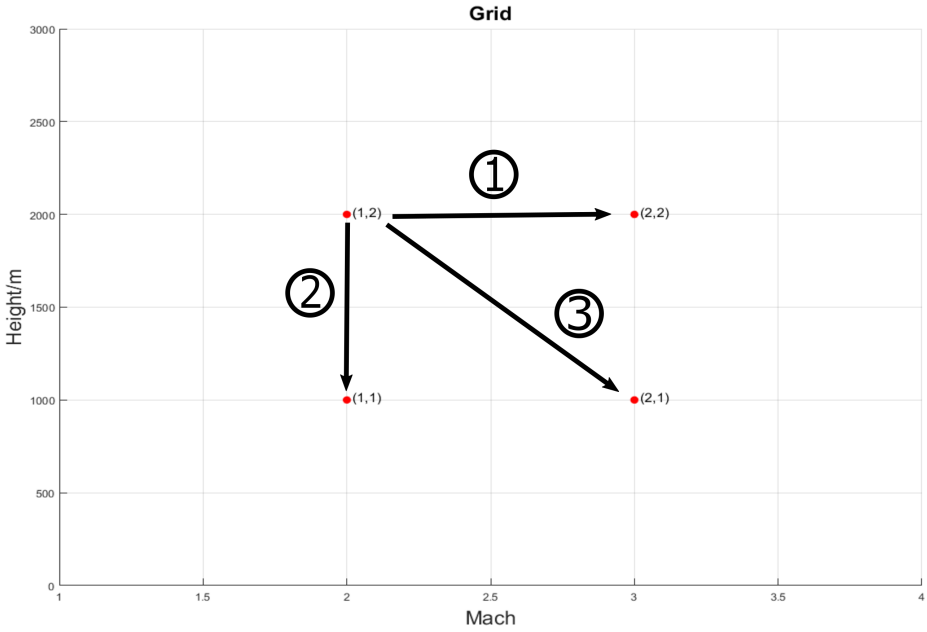

Figure 3.9: Verification steps for the vertex (1,2) in a two-dimensional grid

area is actually as large as illustrated in figure 3.10 for a two-dimensional grid when the operating range is extended like illustrated in figure 3.7. Basic data of the verification is summarized in table 3.5 to get a clue of the number of verification cases and to understand the nomenclature of the grid points. The first dimension of the subscripts is always the Mach number, the second one the height and the third is the angle of attack.

Additionally, figure 3.11 illustrates the increase of grid points and hence the number of verifications according to the grid dimension and the number of grid points along a grid parameter dimension. One can clearly see the enormous increase of verifications of a three-dimensional grid compared to a two-dimensional grid. Hence, the grid points along the third dimension, i.e. the angle of attack, should be as small as possible over the whole operating range.

The verification of the ability to stabilize neighboring OPs is subsequent described in more detail.

As previously mentioned, this is done via LMIs by the prove of an existing positive definite matrix $\boldsymbol{P}$. The algorithm goes through every grid point and determines its adjacent points. Remembering figure 3.9, first adjacent points along a single dimension, i.e. the easy case, will be verified for stability before the combined case will be tested. The reason is that the combined case leads to a denser grid for both dimensions not only a single one. Consider the case where OP (1,2) does not stabilize (2,2) but (1,1). When the verification starts with step 3, i.e. (1,2) $\Rightarrow$ (2,1), the grid gets denser along the Mach number and height. When it starts as shown in figure 3.9, it gets denser only along the Mach number.

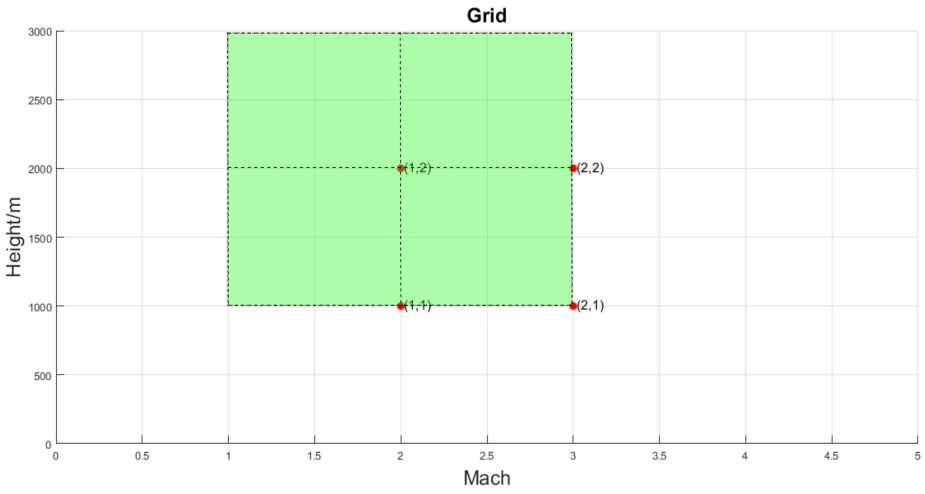

Figure 3.10: Validity area of operating point (1,2) illustrated by the green rectangle

3.4.3.1 LMI Constraints

When the neighboring OPs have been determined, the verification starts for the elevator, rudder and aileron. Hereby is the current controller of the OP which validity area should be verified ($\boldsymbol{A}_{r,c}$, $\boldsymbol{B}_{r,c}$, $\boldsymbol{C}_{r,c}$, $\boldsymbol{D}_{r,c}$) applied to the neighboring plant ($\boldsymbol{A}_n$, $\boldsymbol{B}_n$, $\boldsymbol{B}_{w,n}$, $\boldsymbol{C}_n$, $\boldsymbol{C}_{z,n}$, $\boldsymbol{D}_{w,n}$, $\boldsymbol{D}_{z,n}$, $\boldsymbol{D}_{zw,n}$). Index c is here defined for "current" and index n for "neighbor". The closed loop system for the verification is then according to equation (3.11) now

$$\boldsymbol{A}_{cl} = \begin{bmatrix} \boldsymbol{A}_n + \boldsymbol{B}_n \boldsymbol{D}_{r,c} \boldsymbol{C}_n & \boldsymbol{B}_n \boldsymbol{C}_{r,c} \\ \boldsymbol{B}_{r,c} \boldsymbol{C}_n & \boldsymbol{A}_{r,c} \end{bmatrix}, \tag{3.67}$$

$$\boldsymbol{B}_{cl} = \begin{bmatrix} \boldsymbol{B}_n \boldsymbol{D}_{r,c} \boldsymbol{D}_{w,n} + \boldsymbol{B}_{w,n} \\ \boldsymbol{B}_{r,c} \boldsymbol{D}_{w,n} \end{bmatrix}, \tag{3.68}$$

$$\boldsymbol{C}_{cl} = \begin{bmatrix} \boldsymbol{C}_{z,n} + \boldsymbol{D}_{z,n} \boldsymbol{D}_{r,c} \boldsymbol{C}_n & \boldsymbol{D}_{z,n} \boldsymbol{C}_{r,c} \end{bmatrix}, \tag{3.69}$$

$$\boldsymbol{D}_{cl} = \begin{bmatrix} \boldsymbol{D}_{zw,n} + \boldsymbol{D}_{z,n} \boldsymbol{D}_{r,c} \boldsymbol{D}_{w,n} \end{bmatrix}. \tag{3.70}$$

The verification can be performed by ensuring stability with the LMIs

$$\begin{bmatrix} \boldsymbol{P}\boldsymbol{A}_{cl} + \boldsymbol{A}_{cl}'\boldsymbol{P} & \boldsymbol{P}\boldsymbol{B}_{cl} & \boldsymbol{C}_j' \\ \boldsymbol{B}_{cl}'\boldsymbol{P} & -\gamma_j \mathbb{I} & \boldsymbol{D}_j' \\ \boldsymbol{C}_j & \boldsymbol{D}_j & -\gamma_j \mathbb{I} \end{bmatrix}, \quad \boldsymbol{P} > 0, \quad \gamma_j > 0 \tag{3.71}$$

Table 3.5: Verification cases for 2D and 3D grids according to figures 3.7 and 3.8 respectively

Grid Points	Subscripts	Verifications per OP
Two-Dimensional		
4 Vertices	(1,1) / (1,3) / (3,3) / (3,1)	3
4 Edges	(1,2) / (2,3) / (3,2) / (2,1)	5
1 Middle	(2,2)	8
Total Number of Verification Cases ⇒ 40		
Three-Dimensional		
8 Vertices	(1,1,1) / (1,3,1) / (3,3,1) / (3,1,1) (1,1,3) / (1,3,3) / (3,3,3) / (3,1,3)	7
12 Edges	(1,2,1) / (2,3,1) / (3,2,1) / (2,1,1) (1,2,3) / (2,3,3) / (3,2,3) / (2,1,3) (1,1,2) / (1,3,2) / (3,3,2) / (3,1,2)	11
6 Surfaces	(1,2,2) / (2,3,2) / (3,2,2) / (2,1,2) (2,2,1) / (2,2,3)	17
1 Middle	(2,2,2)	26
Total Number of Verification Cases ⇒ 316		

where index j defines the row of $\boldsymbol{C}_{cl}$ and $\boldsymbol{D}_{cl}$. Additionally one can verify the PPA as well via

$$\boldsymbol{\sigma}' + \boldsymbol{\sigma} < -2\underline{\lambda}\boldsymbol{P}, \tag{3.72}$$

$$\boldsymbol{\sigma}' + \boldsymbol{\sigma} > -2\overline{\lambda}\boldsymbol{P}, \tag{3.73}$$

$$\begin{bmatrix} \delta(\boldsymbol{\sigma}' + \boldsymbol{\sigma}) & \boldsymbol{\sigma}' - \boldsymbol{\sigma} \\ -\boldsymbol{\sigma}' + \boldsymbol{\sigma} & \delta(\boldsymbol{\sigma}' + \boldsymbol{\sigma}) \end{bmatrix} < 0 \tag{3.74}$$

where $\boldsymbol{\sigma} = \boldsymbol{P}\boldsymbol{A}_{cl}$. These additional constraints enhance performance but increase the grid density. It is recommended to apply a tolerance Δ to the limits for the pole placement area. The upper and lower limits are then defined by

$$\underline{\lambda} = \underline{\lambda} \cdot (1 - \Delta), \tag{3.75}$$

$$\overline{\lambda} = \overline{\lambda} \cdot (1 + \Delta), \tag{3.76}$$

where $0 \leq \Delta < 1$. The current controller is now applied to all neighboring OPs. When an error occurs due to the inability of finding a positive definite matrix $\boldsymbol{P}$ to satisfy the Lyapunov function, i.e. stability, or a bad performance $\gamma_n > \Delta\gamma_c$ when this controller tries to stabilize the missile in the surrounding area of the OP by computing the control surface deflection, the grid density along the tested Grid Parameter (GP) increases.
Even if the verification fails for a big tolerance, you should consider to omit LMIs (3.72), (3.73), (3.74).

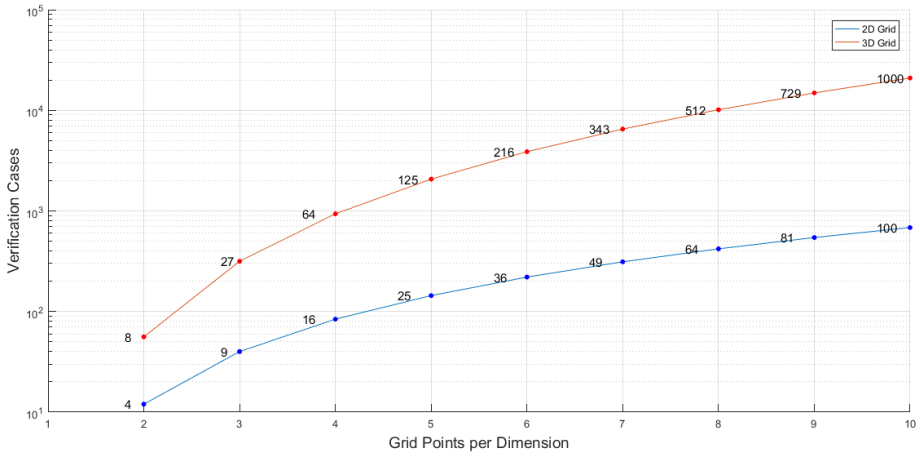

Figure 3.11: Number of verification cases over the grid points per dimension for a two-dimensional(—) and three-dimensional grid (—). The overall growth of the grid points is also illustrated along the two lines

3.5 Gain Scheduling

As we are using the method of linearization as already described in section 2.2, the nonlinear missile has to be trimmed and linearized at various OPs. The controllers which are designed afterwards at these points have to keep the missile stable in the local area of the OP like illustrated in figure 3.10. Consider again the linearized models of the form

$$\delta\dot{\boldsymbol{x}}_i = \boldsymbol{A}_i \cdot \delta\boldsymbol{x}_i + \boldsymbol{B}_i \cdot \delta\boldsymbol{u}_i, \tag{3.77}$$
$$\delta\boldsymbol{y}_i = \boldsymbol{C}_i \cdot \delta\boldsymbol{x}_i + \boldsymbol{D}_i \cdot \delta\boldsymbol{u}_i \tag{3.78}$$

where $\delta\boldsymbol{x}_i = \boldsymbol{x} - \overline{\boldsymbol{x}}_i$, $\delta\boldsymbol{u}_i = \boldsymbol{u} - \overline{\boldsymbol{u}}_i$ and $\delta\boldsymbol{y}_i = \boldsymbol{y} - \overline{\boldsymbol{y}}_i$. There are various possibilities to get the controllers work like a nonlinear controller. Consider figure 3.12 to understand by which controllers the missile will be stabilized. This is exemplary illustrated for the endgame of the missile, i.e. the last seconds until interception.

It is either possible to interpolate between the controller matrices or between the manipulated variables to command the correct control inputs to the elevator, rudder and aileron. Due to the fact that the nonlinearities during the controller design of the output feedback controllers were eliminated by the transformation (3.18), the coordinate systems differ among the controllers. The consequence is that a linear interpolation of the controller matrices is not possible [3]. Hence, the manipulated variable will be interpolated.

The equations for the calculation of such an interpolation over multiple dimensions have been already introduced in section 3.3.3. These equations can be rewritten

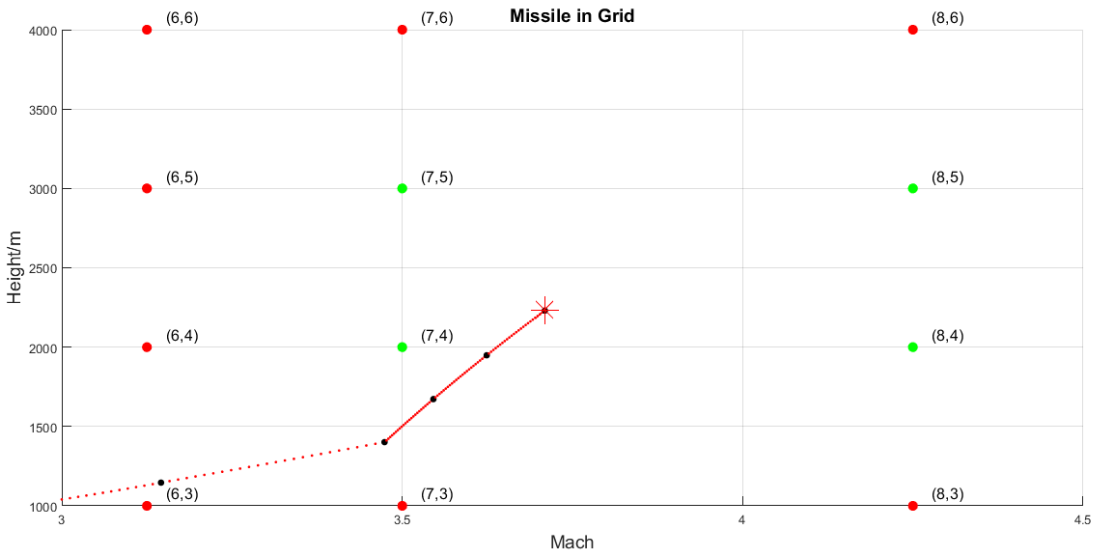

Figure 3.12: The green marked operating points [(7,4); (8,4); (7,5); (8,5)] stabilize the missile (marked by the dotted red line) during the last $\approx 1\,\mathrm{t}$ (1 black dot per t) until strike (marked by the star in red)

to apply them to the interpolation of the manipulated variable [1]

$$\boldsymbol{u} = \frac{\sum_{i=1}^{p} d_M \cdot \boldsymbol{u}_i}{\sum_{i=1}^{p} d_M}, \tag{3.79}$$

$$d_M = 1 - \frac{M_c - (M_1 + \Delta M)}{M_2 - M_1}, \tag{3.80}$$

$$\boldsymbol{u}_M = d_M \boldsymbol{u}_1 + (1 - d_M)\boldsymbol{u}_2. \tag{3.81}$$

Consider figure 3.13 where M represents the Mach number and H the height. The blue illustrated areas around controller one and two are defined by the Δ-values where only this controller provides the control input. We can see a pink area between them where the input will be interpolated by controller one and two. The affiliation parameter d_M weights then how much of the control input is provided by controller one and how much by controller 2. First, the input will be interpolated along M by keeping H constant and afterwards the just determined values will be interpolated along H. For clarification see figure 3.14 where $a = M$ and $b = H$.

This interpolation formulas can be implemented in Simulink for the simulation of the missile during its mission. In order to apply always the appropriate commands, the neighboring OPs will be determined according to the current position of the

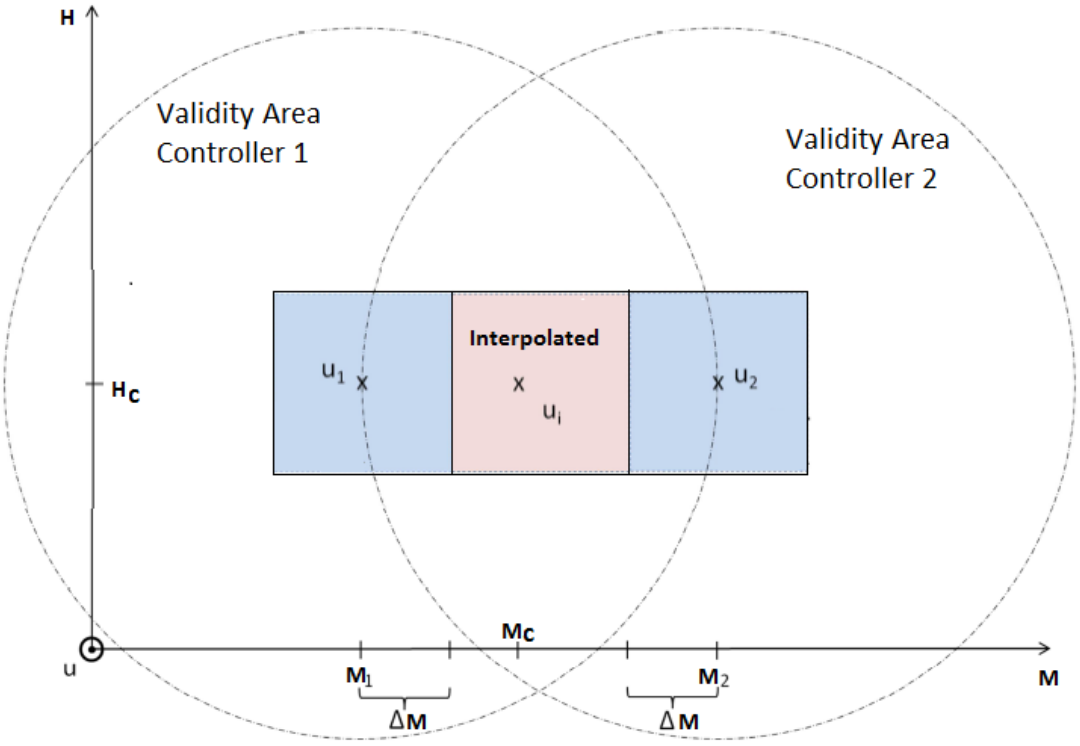

Figure 3.13: Example of the meaning of the validity areas of the controllers

missile in the grid, i.e. flight conditions with respect to Mach number, height and angle of attack. Furthermore, the algorithm in Simulink detects the affiliation to the neighboring OPs in order to weight the influences according to their distance to the missile. Consider figure 3.15 to understand how the algorithm works. The missile starts at $M = 0.6$ and a height of 500 m. Since the missile is accelerating, you can see how the influence of the controllers changes. Until $t \approx 2\,\mathrm{t}$ only the controllers at a height of 500 m [(1,2) and (2,2)] provide the control input since $\Delta H = 50\,\mathrm{m}$ is not exceeded until now and therefore $d_H = 1$ (remember equation (3.80)). At this time 2 t, the influence of controller (1,2) is about 55 % and hence the influence of controller (2,2) is 45 % considering the Mach number. The same algorithm is employed when the missile was trimmed at specific angles of attack but of course with an extended algorithm for three dimensions.
When the missile crosses the imaginary line connecting points (2,2) and (2,3), controllers (1,2) and (1,3) change to (3,2) and (3,3). These controllers will be initialized, i.e. increasing controller states, while the missile is controlled only by (2,2) and (2,3). This section can be seen in the top plot of figure 3.15 between $\approx 4.2\,\mathrm{t}$ and $\approx 4.45\,\mathrm{t}$. Since the missile is in the beginning of this section still controlled by [(1,2); (2,2); (1,3); (2,3)], it has to be divided by ≈ 2 which leads to an initialization phase in this case of $\approx 125\,\mathrm{mt}$ for controllers (3,2) and (3,3).

When the controllers are not asymptotically stable, windup problems occur during blending. Only the closed loops are stable and the controllers need the feedback

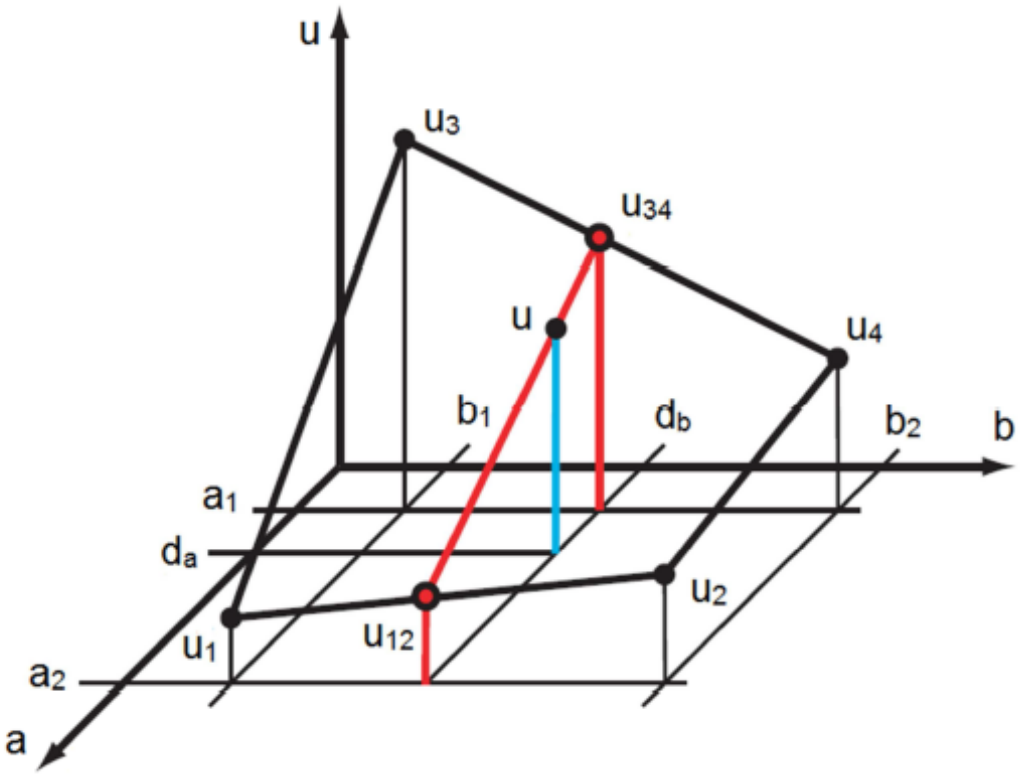

Figure 3.14: Interpolation over multiple dimensions [24]

from the plant. In case of blending where $0 \leq d \leq 1$ the command will be partially or will not received by the plant at all. In worst case only one control loop is closed and the other one without feedback which leads to a divergence of the controller states and hence destabilization of the system [26]. This is the main reason why anti-windup networks have to be implemented for our controller design method. Figure 3.16 illustrates exemplary the anti-windup network in combination with the blending method.

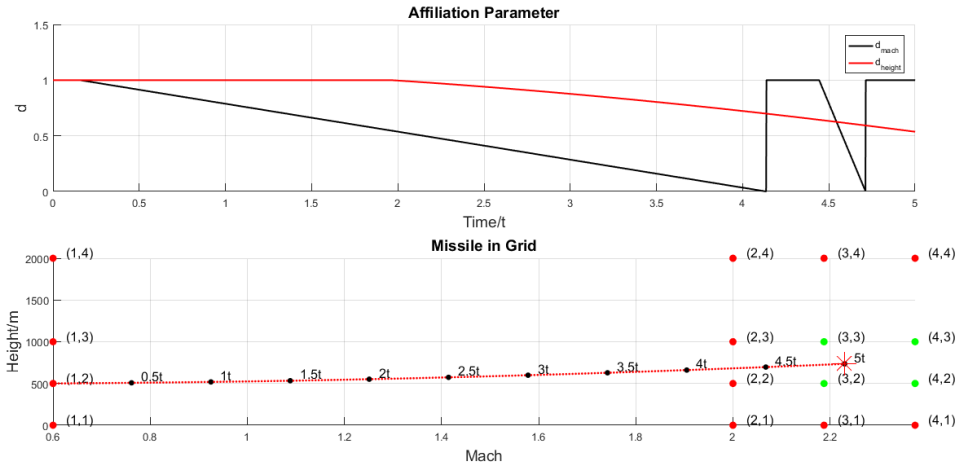

Figure 3.15: Change of the affiliation parameters (top) of the Mach number (—) and height (—) during its flight through the operating point grid (bottom)

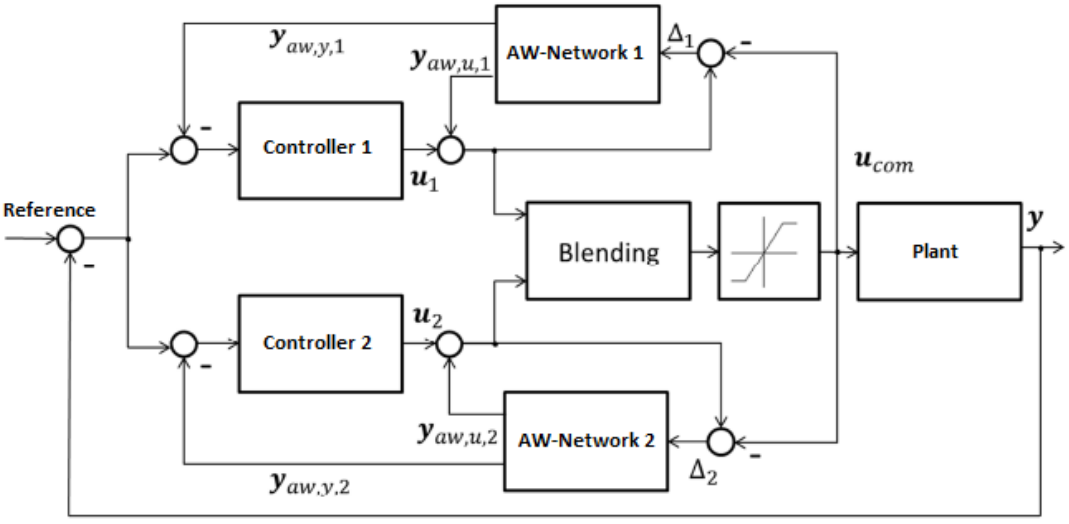

Figure 3.16: Model recovery anti-windup network of two blended controllers

3.6 Anti-Windup

This section is intended to present the anti-windup method according to [13] and [15]. Since the manipulated variable will be determined by four or eight controllers during flight, windup can occur during blending of the controllers.
This method was shortly described in section 2.4.2.1. The idea is a decoupling architecture or rather a separation of the linear unlimited and nonlinear saturated system by using the so-called mismatch-representation. Such an architecture allows the optimization of a logical and intuitive performance criterion by means of convex optimization using LMI constraints.

3.6.1 Model Recovery Anti-Windup

The decoupled anti-windup network introduces an anti-windup compensator which would not replace existing controllers but extend them by an anti-windup network without a new design. This method ensures local stability and robust control even if the system contains poles in the open-right-half complex plane. Therefore, only local anti-windup compensation is covered.
Consider the system shown in figure 2.8 with input saturation and anti-windup compensation. This system is a Finite-Dimensional Linear Time Invariant (FDLTI) plant $\boldsymbol{S}$ which is to be considered of the following form

$$\boldsymbol{S}: \begin{cases} \dot{\boldsymbol{x}}_s &= \boldsymbol{A}_s\boldsymbol{x}_s + \boldsymbol{B}_{s,u}\boldsymbol{u}_s + \boldsymbol{B}_{s,z}\boldsymbol{z}, \\ \boldsymbol{y}_s &= \boldsymbol{C}_s\boldsymbol{x}_s + \boldsymbol{D}_{s,u}\boldsymbol{u}_s + \boldsymbol{D}_{s,z}\boldsymbol{z} \end{cases} \tag{3.82}$$

where $\boldsymbol{x}_s \in \mathbb{R}^{n_s}$ is the plant's state, $\boldsymbol{u}_s \in \mathbb{R}^m$ the control input, disturbance $\boldsymbol{z} \in \mathbb{R}^l$ and output $\boldsymbol{y}_s \in \mathbb{R}^m$. The FDLTI controller $\boldsymbol{R}$ has the following state-space realization

$$\boldsymbol{R}: \begin{cases} \dot{\boldsymbol{x}}_r &= \boldsymbol{A}_r\boldsymbol{x}_r + \boldsymbol{B}_{r,u}\boldsymbol{u}_r + \boldsymbol{B}_{r,z}\boldsymbol{w}, \\ \boldsymbol{y}_r &= \boldsymbol{C}_r\boldsymbol{x}_r + \boldsymbol{D}_{r,u}\boldsymbol{u}_r + \boldsymbol{D}_{r,z}\boldsymbol{w} \end{cases} \tag{3.83}$$

where $\boldsymbol{x}_r \in \mathbb{R}^{n_r}$ is the controller's state, $\boldsymbol{u}_r \in \mathbb{R}^m$ is the measurable input for the output signals of the plant, $w \in \mathbb{R}^m$ is the reference and $\boldsymbol{y}_r \in \mathbb{R}^m$ is the linear controller output. The anti-windup network is defined by equations (2.81) - (2.83)

$$\boldsymbol{AWNW}: \begin{cases} \dot{\boldsymbol{x}}_{aw} &= \boldsymbol{A}_s\boldsymbol{x}_{aw} + \boldsymbol{B}_{s,u}\boldsymbol{k}_{aw}(\boldsymbol{x}_{aw}) + \boldsymbol{B}_{s,u}\delta, \\ \boldsymbol{y}_{aw,y} &= -\boldsymbol{C}_s\boldsymbol{x}_{aw}, \\ \boldsymbol{y}_{aw,u} &= -\boldsymbol{K}_{aw}\boldsymbol{x}_{aw}. \end{cases} \tag{3.84}$$

With the mismatch-transformation (2.92) one gets the linear and nonlinear systems (3.85) and (3.86), respectively. The realization of the model-recovery anti-windup

network in Simulink can be seen in figure 3.17.

$$\boldsymbol{L}: \begin{cases} \dot{\boldsymbol{\xi}}_s & = \boldsymbol{A}_s\boldsymbol{\xi}_s + \boldsymbol{B}_{s,u}\boldsymbol{y}_r + \boldsymbol{B}_{s,z}z \\ \dot{\boldsymbol{\xi}}_r & = \boldsymbol{A}_r\boldsymbol{\xi}_r + \boldsymbol{B}_{r,u}\boldsymbol{C}_s\boldsymbol{\xi}_s + \boldsymbol{B}_{r,w}w \\ \boldsymbol{y}_r & = \boldsymbol{C}_r\boldsymbol{\xi}_r + \boldsymbol{D}_{r,u}\boldsymbol{C}_s\boldsymbol{\xi}_s + \boldsymbol{D}_{r,w}w \\ \boldsymbol{y}_{s,ub} & = \boldsymbol{C}_s\boldsymbol{\xi}_s \end{cases} \tag{3.85}$$

$$\boldsymbol{AWNW}: \begin{cases} \dot{\boldsymbol{\xi}}_{aw} & = \boldsymbol{A}_s\boldsymbol{\xi}_{aw} + \boldsymbol{B}_{s,u}(\mathrm{sat}_u(\boldsymbol{y}_r + \boldsymbol{y}_{aw,u}) - \boldsymbol{y}_r) \\ \boldsymbol{y}_{aw,y} & = \boldsymbol{C}_s\boldsymbol{\xi}_{aw} \\ \boldsymbol{y}_{aw,u} & = \boldsymbol{K}_{aw}\boldsymbol{\xi}_{aw} \end{cases} \tag{3.86}$$

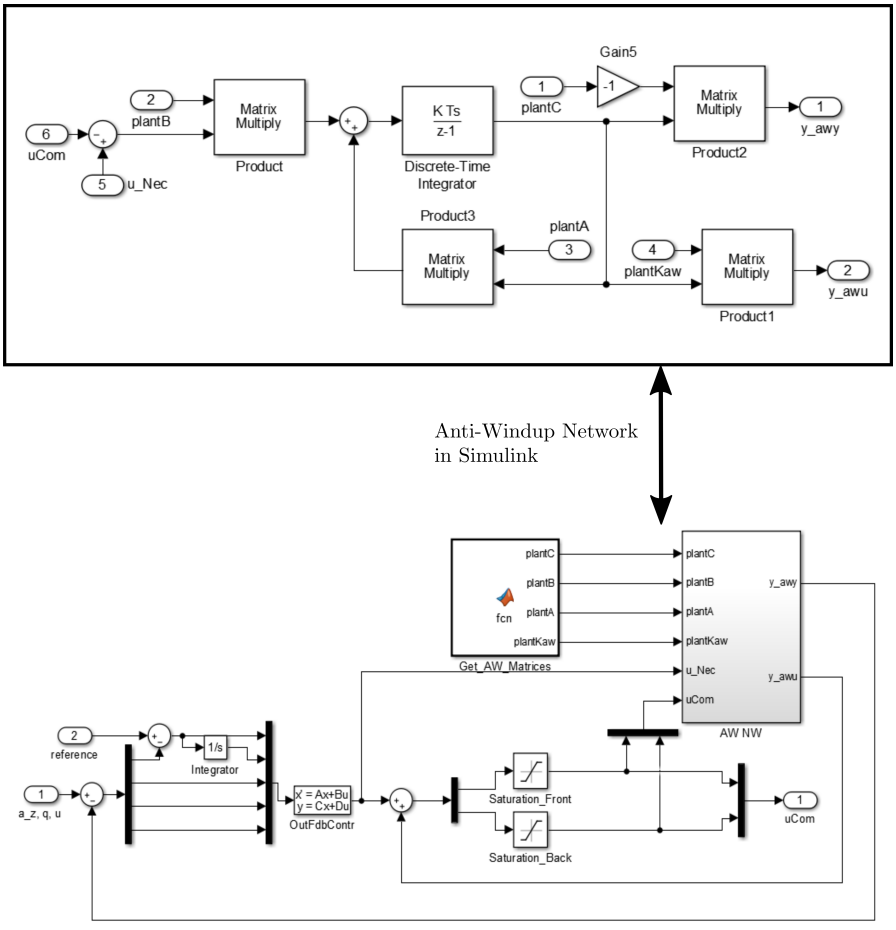

Figure 3.17: Realization of a model recovery anti-windup network in Simulink

Since the system matrix $\boldsymbol{A}_s$ consists of eigenvalues in the open-right-half complex plane, global stability can not be reached. The eigenvalues of $\boldsymbol{A}_s$ or rather the matrix of the elevator of a missile using two aerodynamic actuating systems Aaq is illustrated in figure 3.18.

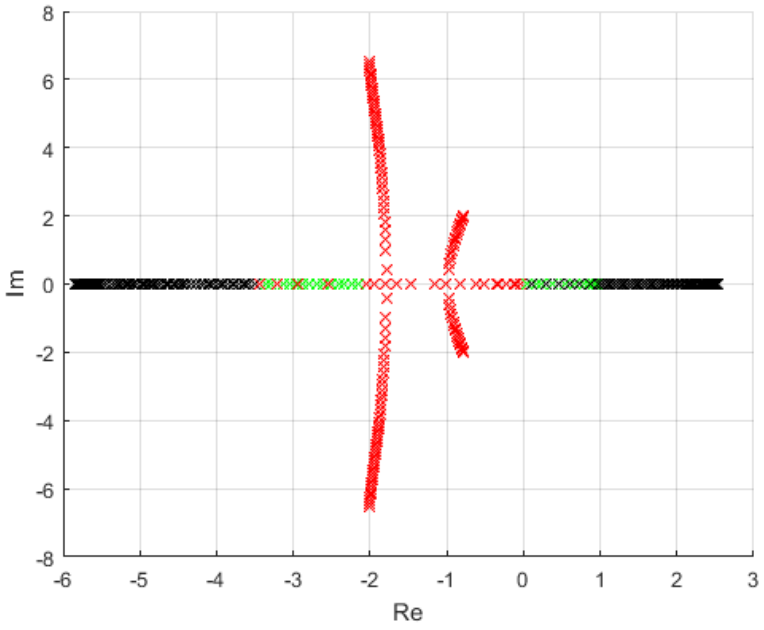

Figure 3.18: Eigenvalues of a missile's elevator open loop system which was linearized at specific Mach numbers and heights

In order to guaranty local stability of the anti-windup compensator, some constraints in the form of linear matrix inequalities have to be defined. In a first step, one has to pay attention of the deadzone nonlinearity of the anti-windup network. Considering the deadzone and hence the maximum possible deflection of a control surface $\overline{\boldsymbol{u}}$ directly in the design of $\boldsymbol{K}_{aw}$ to achieve local stability, one has to sharpen the sector bounds (consider figure 3.19). The maximum possible deflection of a control surface $\overline{\boldsymbol{u}}$ is here defined for the worst case by the consideration of the trim state. Hence, the mentioned $30\,^\circ$ in section 2.1.2 will be either way decreased by the actuators deflection needed to ensure a horizontal flight (section 2.2.1). For example when a missile with $M = 0.6$ and $H = 10\,\text{km}$ needs an elevators deflection of $\eta = 3\,^\circ$ to keep the missile in horizontal flight, the maximal deflection of the elevator which is left is $\overline{u} = 27\,^\circ$.
Under the assumption that $u_i \leq \beta_i \overline{u}_i,\ \forall i$, where $\beta_i > 1,\ \forall i$, one can locally see that $Dz_i(u_i) = u_i - \text{sat}(u_i)$, where $\text{sat}(u_i)$ is defined by equation 2.79, remains below the gradient $\alpha_i := \frac{\beta_i - 1}{\beta_i} < 1$. Index i is the control variable for the number of actuators e.g. the front and back actuation system and parameter β defines the gradient and should not be confused with the sideslip angle of the missile.

Therefore, one can conclude that for all $\boldsymbol{u} \preceq \overline{\boldsymbol{u}}$ the sharper sector $\text{Sector}[0, \mathbb{A}]$ is inherited by the deadzone. The diagonal matrix $\mathbb{A}$ is defined by $\mathbb{A} := \text{diag}(\alpha_1, ..., \alpha_m)$ where m is set by the number of actuators, $\alpha_i \in (0, 1)$ and $\mathbb{A} < \mathbb{I}$. Due to this conclusion, tighter bounds on the small-signal L_2 gain can be obtained and hence local stability is proven.
To guaranty local stability and small-signal L_2 gain, i.e. robustness, the following

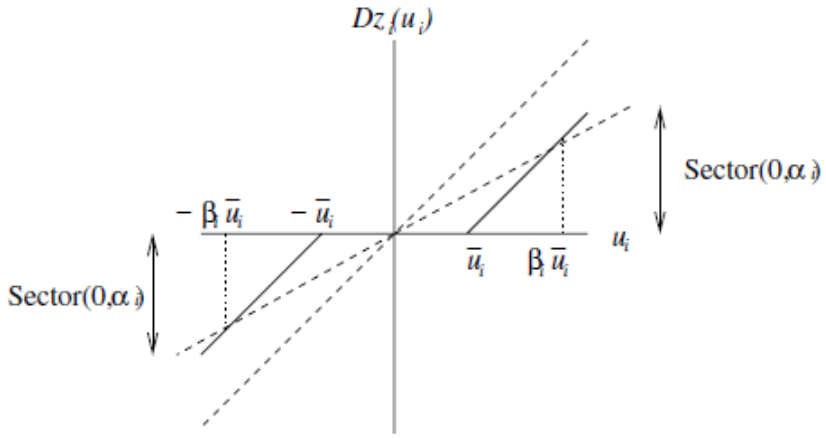

Figure 3.19: Illustration of the deadzone nonlinearity and sector bounds [13]

constraints have to be defined:

- Existence of a positive definite symmetric matrix $\boldsymbol{Q}$

$$\boldsymbol{Q} > 0 \tag{3.87}$$

- Existence of positive definite diagonal matrix $\boldsymbol{U}$ which contains optimization parameters computed by the LMI solver

$$\boldsymbol{U} = \text{diag}(\mu_1, ..., \mu_m) > 0 \tag{3.88}$$

- Existence of a matrix $\boldsymbol{L} \in \mathbb{R}^{m \times n_s}$
- A positive real scalar performance parameter γ such that the following LMI is satisfied

$$\begin{bmatrix} \boldsymbol{Q}\boldsymbol{A}_s' + \boldsymbol{A}_s\boldsymbol{Q} + \boldsymbol{L}'\boldsymbol{B}_s' + \boldsymbol{B}_s L & \boldsymbol{B}_s\boldsymbol{U} - \boldsymbol{L}'\mathbb{A} & \boldsymbol{0} & \boldsymbol{Q}\boldsymbol{C}_s' + \boldsymbol{L}'\boldsymbol{D}_s' \\ (\boldsymbol{B}_s\boldsymbol{U} - \boldsymbol{L}'\mathbb{A})' & -2\boldsymbol{U} & \mathbb{A} & \boldsymbol{U}\boldsymbol{D}_s' \\ \boldsymbol{0} & \mathbb{A}' & -\gamma\mathbb{I} & \boldsymbol{0} \\ (\boldsymbol{Q}\boldsymbol{C}_s' + \boldsymbol{L}'\boldsymbol{D}_s')' & (\boldsymbol{U}\boldsymbol{D}_s')' & \boldsymbol{0} & -\gamma\mathbb{I} \end{bmatrix} < 0 \tag{3.89}$$

If the LMIs (3.87), (3.88) and (3.105) are satisfied, a suitable anti-windup gain which achieves $\boldsymbol{\xi} < \gamma$ can be obtained by:

$$\boldsymbol{K}_{aw} = \boldsymbol{L}\boldsymbol{Q}^{-1} \tag{3.90}$$

The mismatch parameter $\boldsymbol{\xi}$ defines the difference between nominal linear behavior and a saturation event. By minimizing the positive performance parameter γ, the mismatch will be minimized as well. Figure 3.20 shows the influence of gradient α_i on the performance parameter γ.
In order to enhance the performance, further LMIs can be defined. The maximization of the Region Of Attraction (ROA) is a good option to enhance the

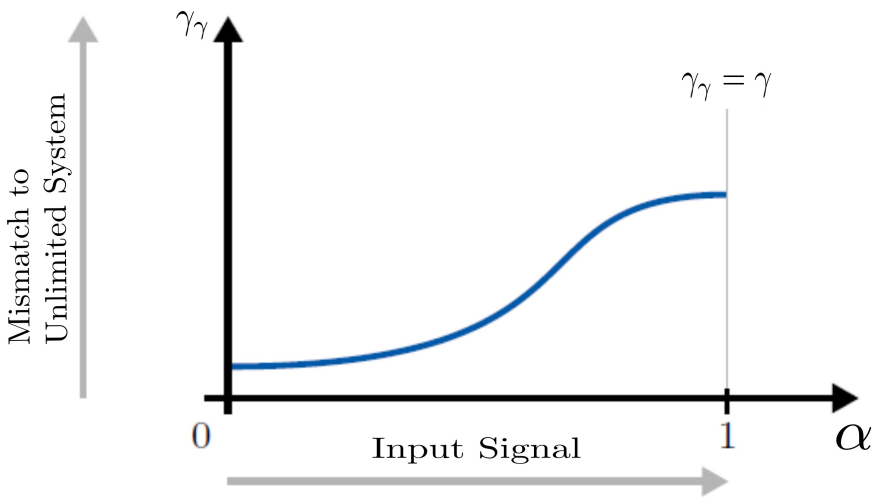

Figure 3.20: Deviation between mismatch-system and unlimited system as a function of the input signal α. A measure of the mismatch performance is the upper bound of the small-signal L_2 gain γ_γ of the anti-windup system [15].

performance. Such a region can be defined by a Lyapunov function [28]. Whenever the initial state is in the region defined by Lyapunov, stability is guaranteed. For the design of the anti-windup gain, the following inequality defines the ROA as an ellipsoid:

$$\mathcal{E}(\boldsymbol{P}) = \{\boldsymbol{\xi}_{aw} \in \mathbb{R}^{n_s} : \boldsymbol{\xi}'_{aw}\boldsymbol{P}\boldsymbol{\xi}_{aw} \leq 1\} \tag{3.91}$$

where $\boldsymbol{\xi}_{aw}(0) = 0$ and the positive definite matrix $\boldsymbol{P}$ defines the ROA. To hold the assumption of $u_i \leq \beta_i \overline{u}_i$:

$$|\boldsymbol{k}'_{aw,i}\boldsymbol{\xi}_{aw}| \leq \beta_i \overline{u}_i, \forall i, \quad \text{where } \boldsymbol{K}_{aw} = \begin{bmatrix} \boldsymbol{k}'_{aw,i} \\ \vdots \\ \boldsymbol{k}'_{aw,m} \end{bmatrix}. \tag{3.92}$$

When $\boldsymbol{\xi}_{aw} \in \mathcal{E}$ and using the substitution method presented in section 2.3.1.4, $\boldsymbol{P} = \boldsymbol{Q}^{-1}$, one gets the following LMIs:

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{L}' \\ \boldsymbol{L} & \beta_i^2 u_i^2 \mathbb{I} \end{bmatrix} \geq 0, \forall i \tag{3.93}$$

Since the ellipse (3.91) should be maximized, the inverse matrix $\boldsymbol{Q}$ has to be minimized [13]. Therefore, the following objective should be minimized when solving the LMIs to get $\boldsymbol{K}_{aw}$

$$\Gamma = \eta\gamma + (1-\eta)\det(\boldsymbol{Q}) \tag{3.94}$$

where $\eta \in [0,1]$ is the trade of the importance between the local performance and the minimization of the ROA and should not be confused with the deflection of the elevator.
Additionally, the exact region of states of which the system should be captured, can be enforced by defining a rectangle in terms of setting its vertices. These vertices are the maximum values of the states where stability should be guaranteed. When this predefined region too large, the problem gets infeasible. Considering the following Lyapunov function

$$a^2 \boldsymbol{\xi}_s' \boldsymbol{P} \boldsymbol{\xi}_s \leq 1 \tag{3.95}$$

$$\Leftrightarrow \frac{1}{a^2} - \boldsymbol{\xi}_s' \boldsymbol{P} \boldsymbol{\xi}_s \geq 0. \tag{3.96}$$

Using substitution and setting $\gamma_\alpha = \frac{1}{a^2}$ one gets the LMI

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{\xi}_{s_i} \\ \boldsymbol{\xi}_{s_i}' & \gamma_{\alpha_i} \end{bmatrix} \geq 0, \forall i \tag{3.97}$$

where $\boldsymbol{\xi}_{s,i}$ is a common point of a predefined region. The aim is now to maximize γ_α and hence minimize a^2 which minimizes $\boldsymbol{Q}$ and maximizes the ROA $\mathcal{E}(\boldsymbol{P})$. The objective function which should be then minimized can be stated as

$$\Gamma = \eta\gamma + (1-\eta)\frac{1}{\sum \gamma_{\alpha_i}}. \tag{3.98}$$

3.6.2 Affine Linear Parameter Dependency

During the flight of the missile, the current conditions like Mach number or height vary and hence the missile moves through the operating point grid. Consider the extraction of a two-dimensional grid illustrated in figure 3.21 and the rectangle defined by the four operating points [(1,1); (2,1); (1,2); (2,2)].
In order to apply always the appropriate control input to the system, this value has to be interpolated by considering all these surrounding operating points according to the current conditions. This leads to a validity area where stability is provided by a specific controller (e.g. at OP (1,2) in figure 3.10 or 3.21).

The control input will be interpolated either by four OPs (two-dimensional grid) or by eight OPs (three-dimensional grid). The same applies for $\boldsymbol{y}_{aw,u_p}$ computed by the anti-windup gains $\boldsymbol{K}_{aw_p}$ where index p is a counter which denotes the current OP by running through the grid dimensions, i.e. first rows, afterwards columns. Hence, $p = 2$ denotes OP (2,1), $p = 3$ denotes OP (1,2) and so on.
How the control input will be interpolated and how the guaranty of stabilizing the missile in a specific validity area of a controller is given was presented in sections 3.3 and 3.5.

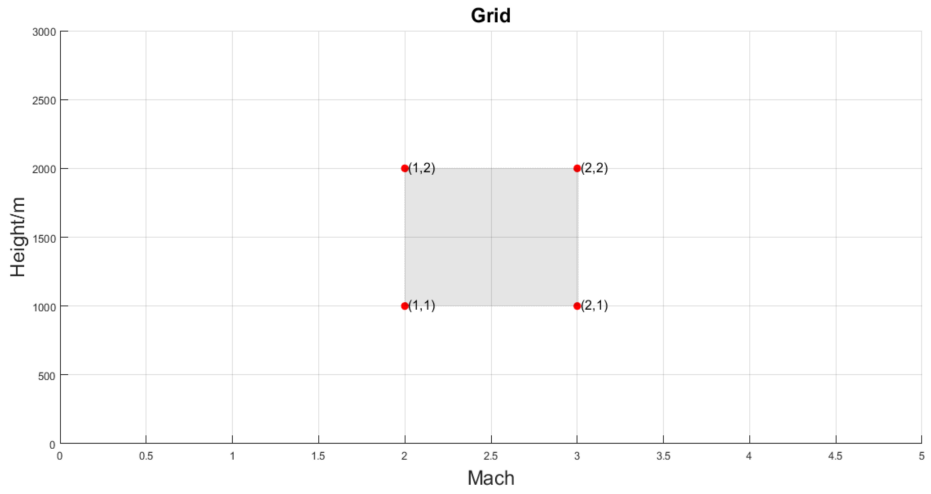

Figure 3.21: Extraction of a two-dimensional operating point grid

Ensuring that every OP stabilizes its neighboring OP, leads to an inclusion of these adjacent OPs in the anti-windup design which is possible because the anti-windup network relies on the design of a state feedback controller. This can be realized due to the assumption of an affine linear parameter dependency of the linearized models from the grid parameters. The system of the anti-windup network is assumed to be

$$\Delta\dot{\boldsymbol{x}} = (\boldsymbol{A}_s + d_{gd}\Delta\boldsymbol{A}_s)\Delta\boldsymbol{x} + (\boldsymbol{B}_s + d_{gd}\Delta\boldsymbol{B}_s)\Delta\boldsymbol{u}, \tag{3.99}$$

$$\Delta\boldsymbol{y} = (\boldsymbol{C}_s + d_{gd}\Delta\boldsymbol{C}_s)\Delta\boldsymbol{x}. \tag{3.100}$$

Before the verification whether this assumption can hold, the plant matrices of the anti-windup network $\boldsymbol{A}_s$, $\boldsymbol{B}_s$, $\boldsymbol{C}_s$ and $\boldsymbol{D}_s$ will be defined as following shown

$$\boldsymbol{A}_s = \begin{bmatrix} \boldsymbol{A}_{aq} & \boldsymbol{B}_{aq} \\ \boldsymbol{0} & -\mathrm{diag}(\boldsymbol{T}_{Act}) \end{bmatrix}, \tag{3.101}$$

$$\boldsymbol{B}_s = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{T}_{Act_1} & 0 \\ 0 & \boldsymbol{T}_{Act_2} \end{bmatrix}, \tag{3.102}$$

$$\boldsymbol{C}_s = \begin{bmatrix} \boldsymbol{C}_{aq} & \boldsymbol{D}_{aq} \\ 0\ 1 & 0\ 0 \\ 0\ 0 & 1\ 0 \\ 0\ 0 & 0\ 1 \end{bmatrix}, \tag{3.103}$$

$$\boldsymbol{D}_s = \begin{bmatrix} \boldsymbol{D}_{aq} \\ 0\ 0 \\ 1\ 0 \\ 0\ 1 \end{bmatrix}, \tag{3.104}$$

where

$\boldsymbol{A}_{aq}$ = System matrix of the elevator axis
$\boldsymbol{B}_{aq}$ = Input matrix of the elevator axis
$\boldsymbol{C}_{aq}$ = Output matrix of the elevator axis
$\boldsymbol{D}_{aq}$ = Feedthrough matrix of the elevator axis
$\boldsymbol{T}_{Act}$ = Time constants of the actuators $(1/T_{aero_i}, i = 1, ..., m)$.

The linearized missile has an affine linear parameter dependency according to the grid parameters Mach number, height and angle of attack. This can be seen in figure 3.22 where all elements of the system matrix of the elevator of a missile along the Mach number from $M = 0.6 - 2$ divided in 200 iteration steps is illustrated. Since no anomaly occurs in the curve characteristic, affine linear parameter dependency is a valid assumption.

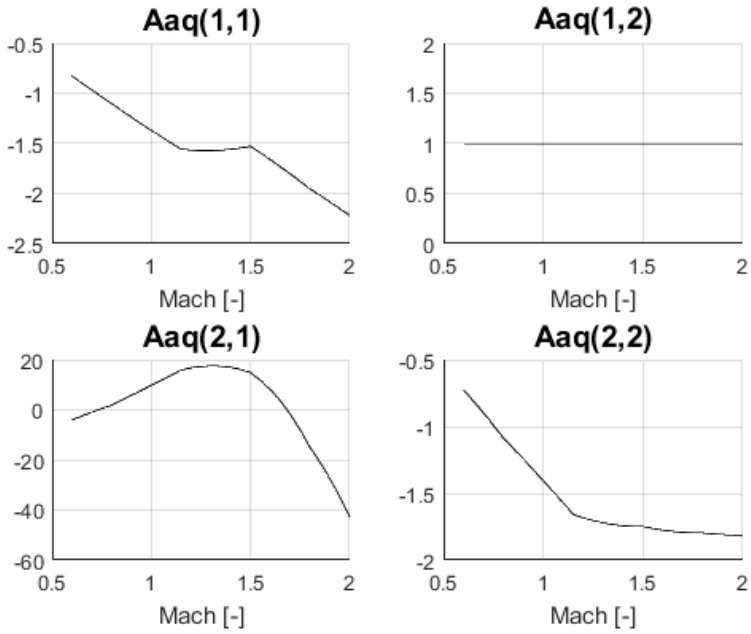

Figure 3.22: Affine linear parameter dependency of the elements of the elevator's system's matrix as a function of the Mach number

For the consideration of the neighboring plants, one can extend LMI (3.105). This constraint will be applied to all surrounding plants as subsequent stated

$$\begin{bmatrix} \boldsymbol{Q}\boldsymbol{A}'_{s_p} + \boldsymbol{A}_{s_p}\boldsymbol{Q} + \boldsymbol{L}'\boldsymbol{B}'_{s_p} + \boldsymbol{B}_{s_p}\boldsymbol{L} & \boldsymbol{B}_{s_p}\boldsymbol{U} - \boldsymbol{L}'\mathbb{A} & \boldsymbol{0} & \boldsymbol{Q}\boldsymbol{C}'_{s_p} + \boldsymbol{L}'\boldsymbol{D}'_{s_p} \\ (\boldsymbol{B}_{s_p}\boldsymbol{U} - \boldsymbol{L}'\mathbb{A})' & -2\boldsymbol{U} & \mathbb{A} & \boldsymbol{U}\boldsymbol{D}'_{s_p} \\ \boldsymbol{0} & \mathbb{A}' & -\gamma\mathbb{I} & \boldsymbol{0} \\ (\boldsymbol{Q}\boldsymbol{C}'_{s_p} + \boldsymbol{L}'\boldsymbol{D}'_{s_p})' & (\boldsymbol{U}\boldsymbol{D}'_{s_p})' & \boldsymbol{0} & -\gamma\mathbb{I} \end{bmatrix} < 0, p = 1, ..., n_{OP} \tag{3.105}$$

where n_{OP} is the number of surrounding plants. For our exemplary grid in figure 3.21 stability during saturation is now guaranteed in the gray rectangle since $\boldsymbol{K}_{aw}$

is able to stabilize the system at the four vertices simultaneously. This figure shows only an extraction due to clarity. For a whole flight envelope, the consideration of surrounding gridpoints in the design depends on the grid dimension or where the point is situated in the grid (vertex, edge (,surface) or middle).

In order to provide a short overview of the influence of the design parameters on the anti-windup network, these parameters will be varied for a missile using two aerodynamic actuating systems. The results are summarized in table 3.6 and illustrated in the following figures. The missile had a Mach number of $M = 3.5$ and a height of $H = 3\,\text{km}$. The anti-windup gain was designed such that the rectangular area with the following vertices

$$Vertices = \begin{bmatrix} \alpha \\ q \\ \eta_f \\ \eta_b \end{bmatrix} = \begin{bmatrix} \pi/9 \\ \pi \\ \pi/4 \\ \pi/4 \end{bmatrix}. \tag{3.106}$$

are included in the ROA of the AW NW.

Table 3.6: Different parametrization of the anti-windup design parameters and its effect on the behavior during saturation. The anti-windup gain was designed including also the rectangular area to enhance performance.

Case	Design Parameter	Value	Figures
1	β	1.01	3.23
	η	0	
	$\Rightarrow \gamma$	625543.0	
2	β	10	3.23
	η	1	
	$\Rightarrow \gamma$	9559.9	
3	β	4	3.23 and 3.24
	η	0.5	
	$\Rightarrow \gamma$	5366.8	
4	β	1.01	3.24
	η	1	
	$\Rightarrow \gamma$	39.2	

As we can see by comparing the results illustrated in figures 3.23 & 3.24, the third case with mean values of β and η has the best performance during saturation. The rise time is quite short and the amplitude of the anti-windup network is the smallest compared to the other cases. Furthermore, the overshoot is zero. Only case four provides almost the same behavior like case three. The amplitude of the anti-windup states is larger but the performance parameter γ is much smaller compared to γ of the third case (see table 3.6). Even though the performance parameter is much smaller, case three is still better in its application. The reason gives the definition of the parameter itself. Since this parameter decreases the mismatch between nominal linear behavior and saturated nonlinear behavior, the activation of the anti-windup network has to be much bigger in order to keep the mismatch

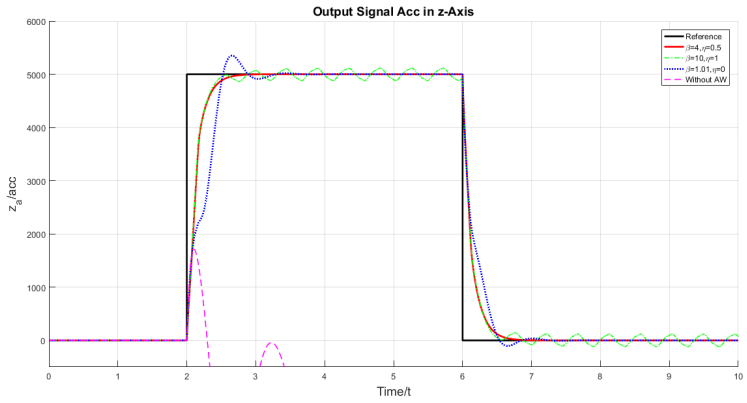

Figure 3.23: Output signals during saturation of a missile's elevator dynamics. Reference signal illustrated in (—), and the design parameters $\beta = 4$, $\eta = 0.5$ illustrated in (—), $\beta = 10$, $\eta = 1$ illustrated in (—), $\beta = 1.01$, $\eta = 0$ illustrated in (—) and without AW illustrated in (—)

small. The performance is of course also influenced by the saturation limit. The result is that for higher saturation cases, the anti-windup network will not be able to guaranty good performance anymore since the states' amplitude is too big and a fast recovery is not possible during saturation (see figure 3.24). This can be also explained by having a closer look to the sector bounds. Remember figure 3.19 and the calculation of gradient $\alpha = \frac{\beta-1}{\beta}$. In case of $\beta = 1.01$, $\alpha = 0.99\,\%$. Hence, the sector$[0, \mathbb{A}]$ is only $0.99\,\%$ of sector$[0, \mathbb{I}]$, i.e. unity gradient line. The sector is too sharp which leads to a decreased performance compared to a larger gradient when the system is highly saturated.
Concluding, it can be said that a balance between performance and robustness has to be achieved. Having a $50\,\%$ weighting between the minimization of the mismatch to the maximization of the ROA and a gradient which is $75\,\%$ of the unity gradient line sector to guarantee a good robustness is a good solution.

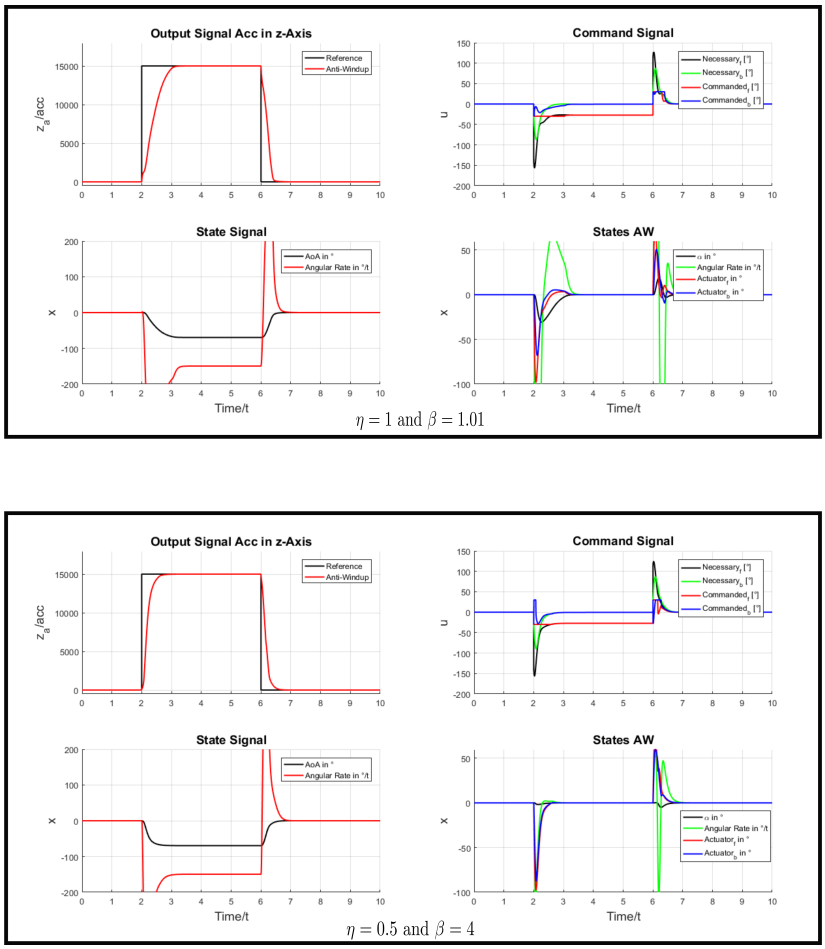

Figure 3.24: Comparison for an exaggerated lateral acceleration step. Illustration of output signals, commands and states of the missile and the anti-windup during saturation of a missile's elevator dynamics with different design parameters ($[\eta = 1,\ \beta = 1.01]$ top picture and $[\eta = 0.5,\ \beta = 4]$ bottom picture)

4 Results

This chapter is intended to remind you briefly of the tasks of this master thesis, to discuss problems which occurred during the design and afterwards the implementation of the automated controller design as well as the results of the grid generation, the controller performance and robustness and the anti-windup performance will be presented.
The task of an automated controller design using convex optimization was separated in four main tasks. First, the operating point grid should have been automatically adapted for two-dimensional or three-dimensional regular grids according to the validity areas in order to get a grid of controllers which is able to stabilize the missile in the whole operating range. The controllers at the OPs should have been designed by considering the conditions of the current OP in order to get higher dynamic controllers at high velocity, low altitude and low angle of attack and achieve therefore a higher agility of the missile compared to small Mach number, high altitude and large angle of attack. Afterwards a method had to be implemented which counteracts the problem of windup over the whole operating range especially during blending which is part of the last task, the implementation. A missile should be simulated by applying the designed controllers and anti-windup networks and by using gain-scheduling to blend the manipulated variable in order to stabilize the missile on its way to the target. It was also required to hold specific performance criteria [3], i.e. the magnitude of the overshoot ($\pm 15\,\%$), settling time (where the deviation from the set value is smaller or equal $5\,\%$) and rise time (maximal 520 mt) according to figure 4.1. The performance and robustness of the controllers will be presented in section 4.2.

During the controller design and automated grid adaption, problems occurred at Mach numbers in a range of $0.6 \leq M \leq 2$. As the EoMs have been described in section 2.1.1, the derivatives for the representation of the missile's moments and forces have been presented. These derivatives differ a lot in this range which is due to the huge differences between the subsonic and supersonic physics of the missile. These changes are shown in figure 4.2 for some derivatives to provide an idea of the problem. You can clearly see that the characteristic curve has large changes in the critical region (marked in light red) compared to the remaining smooth curve. The result of this irregular curve is a high grid density since the controllers are not able to stabilize neighboring OPs in a local area of satisfying size. A normal or satisfying number of grid points along the Mach number or height is maximal 10 to 15 and along the angle of attack 4 to 6. The changes of the derivatives lead to 20 grid points only in the region between 0.6 and 1.2 Mach, hence a distance along the Mach number of 0.032. In case of using regular grids, this interval would lead to nearly 140 grid points only along the Mach number, about 1400 grid points when considering a two-dimensional grid and approximately 5600 grid

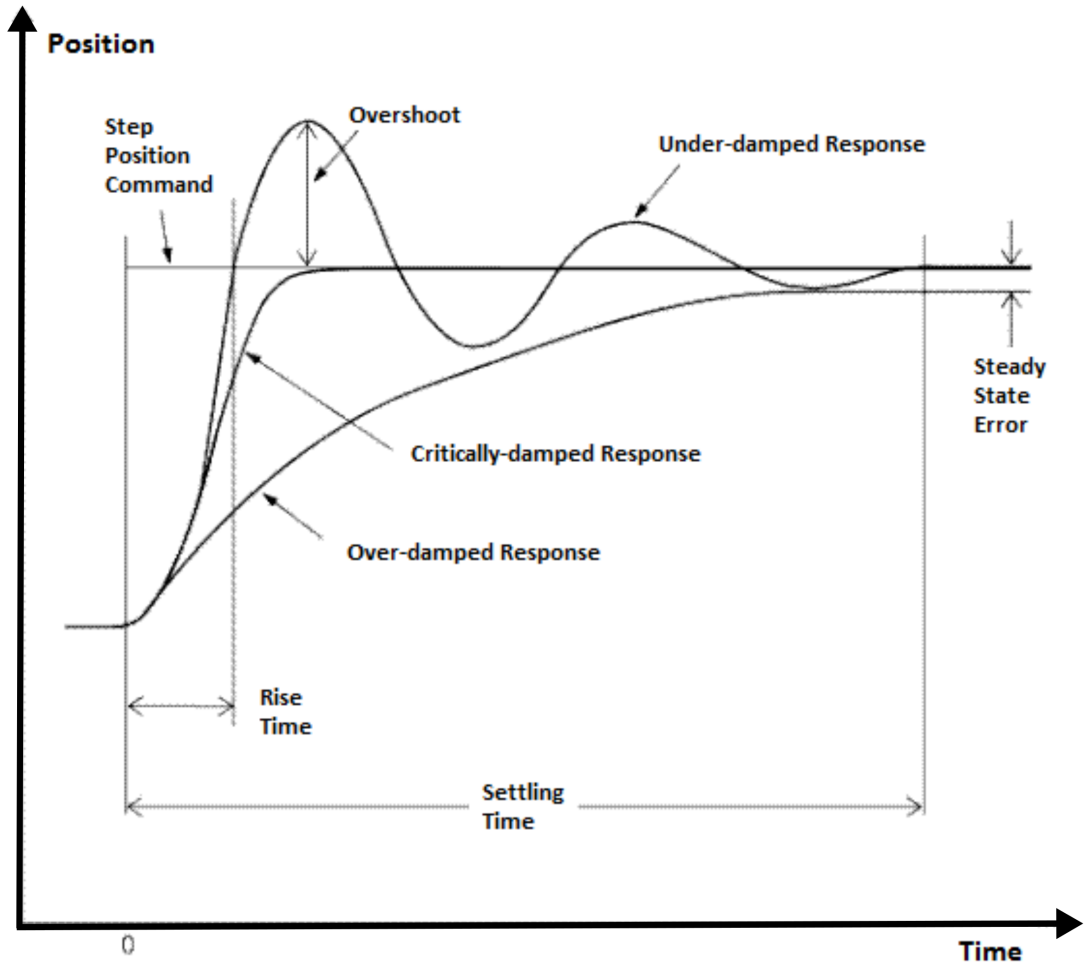

Figure 4.1: Step responses of an arbitrary system [23]

points when a three-dimensional grid should be generated. The computing time would reach exorbitant amounts. The solution to overcome this problem is the usage of another kind of controller for the critical region. Since the air defense missile will be accelerated to velocities above $M = 4$ very quickly, the critical region will be crossed very fast. Therefore, it is not necessarily required that the missile performs lateral accelerations in this short phase. Our solution to overcome this problem is the usage of state feedback controllers to control the rotation rates of the z- and y-axes to zero or another reference rotation rate to stabilize the missile in this region. The result will be shown as part of a whole flight of a missile in section 4.1.
Another additional solution to reduce the number of grid points and hence increase performance of the automated design is a first step in generating irregular grids. Therefore, the distance between OPs will not be decreased among all grid points but only in the region where the validity area of the controller is not as big as in other regions.
One can decrease the maximal bound of the critical region to $M = 1.5$ but due

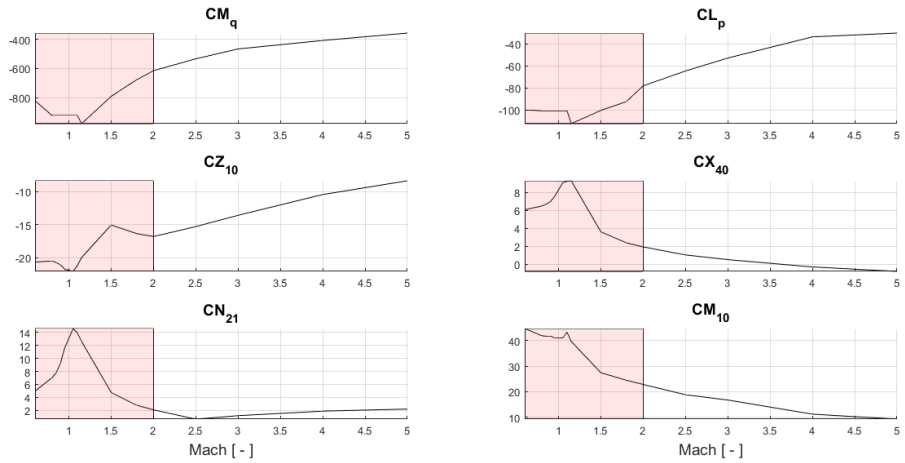

Figure 4.2: Derivatives as a function of Mach number. The light red marked patch illustrated the critical region

to the discontinuities of e.g. derivatives CZ_{10} and CL_p between $1.5 \leq M \leq 2.2$ the grid density would exceed the satisfying number of grid points by additional 15 points as shown in figure 4.3.
This leads to over 150 additional points for a two-dimensional grid and approximately 600 points for a three-dimensional grid. Remember figure 3.11 for the number of verification cases. The computation time of the controller design would increase a lot. To emphasize the advantage of irregular grids over regular grids imagine the increase of the distance between adjacent grid points along the Mach number as stated before. The smallest interval among the 15 additional points is between points [(5,1); (6,1); (7,1)] and [(8,1); (9,1); (10,1)]. Remember figure 3.22 where the elements of the system's matrix had been plotted. You can see this curve enlarged in figure 4.4. This figure illustrates also the real parts of the Eigenvalues of the system's matrix. You can clearly see that the transition from positive unstable eigenvalues to negative stable Eigenvalues is exactly in this region which leads to such a dense grid. The interval is in this region $M = 0.011$. In case of regular grids, the overall number of grid points from $M = 1.5$ to $M = 5$ would increase from 25 to about 310 grid points. Extrapolating this to three dimensions would lead to about 12400 grid points for a regular grid compared to 1000 grid points for an irregular grid.

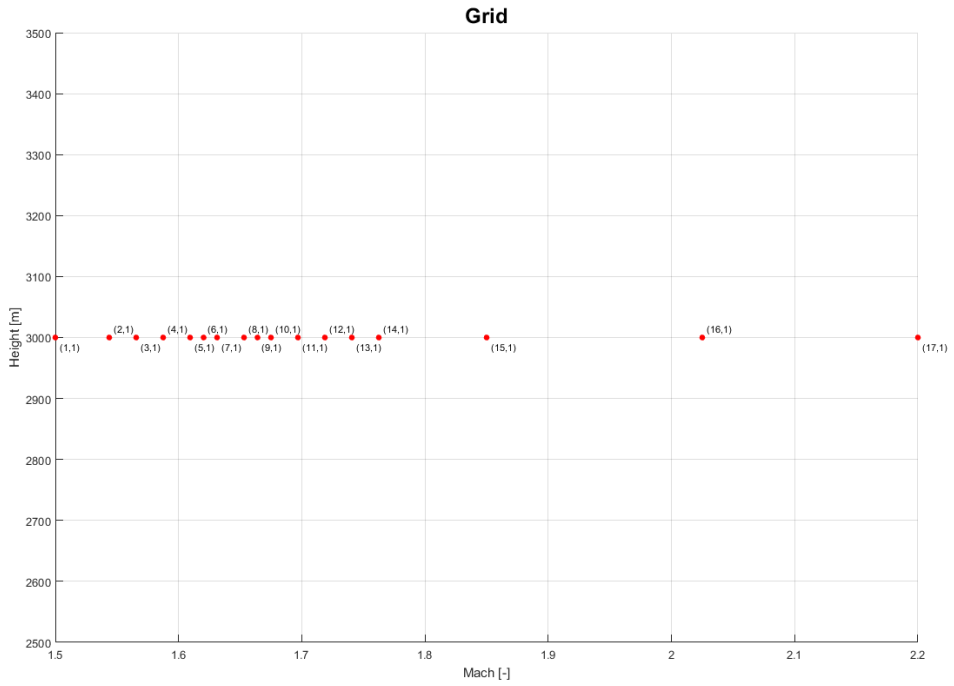

Figure 4.3: Grid density along the Mach number at an arbitrary height for the last part of the critical region when designing dynamic output feedback controllers

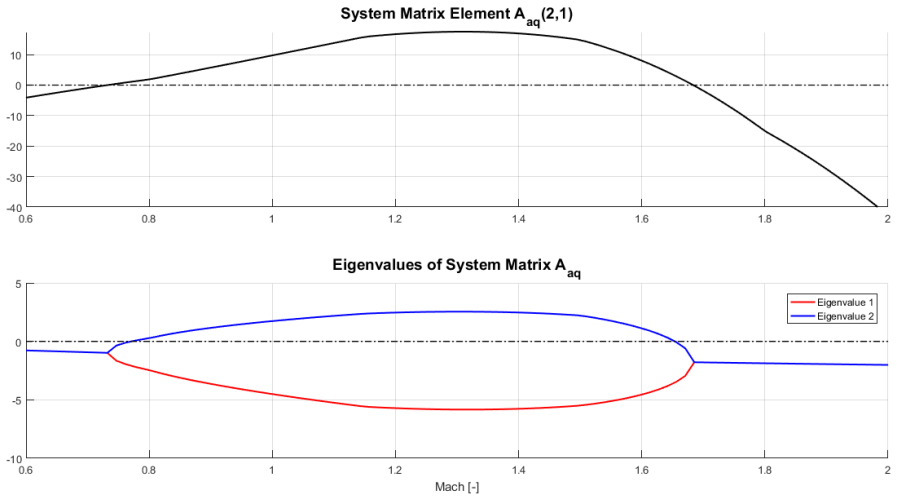

Figure 4.4: System matrix element $A_{aq}(2,1)$ (top) and real parts of the eigenvalues of the system's matrix (bottom) plotted over the Mach number

4.1 Implementation

This section is intended to present the implementation on the basis of an automated controller design of a missile using two aerodynamic actuating systems. The missile was linearized over a two-dimensional grid, i.e. Mach number and height. The input which was necessary to execute the automated controller design is summarized in table 4.1.

Table 4.1: Input data for the automated controller design for the whole operating range of a missile using two aerodynamic actuating systems

Data	Value	Description
Mach number [–]	$M = 0.6 - 5$	Operating range Mach number
Height [m]	$H = 0 - 10\,\mathrm{km}$	Operating range height
Angle of Attack [°]	Not considered	Missile was only linearized over Mach and height
Distortion	0	No distortion of the derivatives to prove robustness
Missile	Missile 2	Controller design applied to missile 2
Critical Region	$0.6 \leq M \leq 2$	Definition of the critical region boundaries
Trim Variables	α and η_b	Trim variables for the straight and level flight set to the angle of attack and the rear elevator
AW Gain Method	2	AW gain design considers rectangular region of states of which the system should be captured

After approximately three hours, the design of the controllers for the critical and non-critical region as well as anti-windup networks was finished. Figure 4.5 illustrates the grid at the beginning of the automated controller design and the final grid which was generated according to the criteria presented in section 3.4.3.
Afterwards, the missile had been simulated for various scenarios in Simulink. As you might have noticed in table 4.1 you can choose between different missiles. In the course of this thesis the design was insomuch extended that it can be applied to other missiles. Furthermore, the generic design was also applied to the simulation where only one Simulink model is needed to simulate missiles with one or two aerodynamic actuating systems, missiles which have been linearized over two or three dimensions and it is able to simulate the missile performing predefined lateral accelerations or performing an air defense maneuver using guidance laws to eliminate a predefined target.
The result of an exemplary air defense maneuver will be subsequently presented. The simulation input was defined as shown in table 4.2.
The results of the simulation are shown in figures 4.6, 4.7 and 4.8. As you can see in figure 4.6, the missile is controlled very quickly to reach the desired accelerations. In the beginning, the missile starts at $M = 0.6$ and $H = 0\,\mathrm{km}$ and accelerates through the critical region until $\approx 4.3\,\mathrm{t}$. Velocities below $M = 0.6$ are not considered since no lateral acceleration steps or guidance commands are controlled in this initial boost phase. A pitch rate of $5\,^\circ/\mathrm{s}$ was set to increase the angle of attack and hence the elevation angle and therefore height until the missile reached almost supersonic velocity. During the change from subsonic to supersonic velocities to the end of the critical region the rotation rates are controlled to zero.

Table 4.2: Simulation input data for an air defense maneuver with a missile equipped with two aerodynamic actuating systems

Data	Value	Description
Grid point	(1,1)	Initial Mach number and height set to $M = 0.6$ and $H = 0\,\mathrm{km}$
Thrustphase	1	Start of simulation with boost phase
Elevation angle	$10\,^\circ$	Angle of ignition (elevation)
Azimuth angle	$40\,^\circ$	Angle of ignition (azimuth)
Target position	$[6000; 1000; -10000]\,\mathrm{m}$	Initial position of the target in NED frame
Offset	$[0; -200; 1000]\,\mathrm{m}$	Step in NED frame the target makes when the missile approached to a defined distance
Approach	$3000\,\mathrm{m}$	When the missile reached this distance to the target, the step is executed
Pitch rate	$[5\ \ 0]\,^\circ/\mathrm{s}$	Pitch rate before and after $M = 0.9$ until the end of the critical region
Yaw rate	$[0\ \ 0]\,^\circ/\mathrm{s}$	Yaw rate before and after $M = 0.9$ until the end of the critical region

Afterwards, the missile receives the commands to reach its desired path to strike the target at the initial target position $[6000; 1000; -10000]\,\mathrm{m}$ in the NED frame. Figure 4.8 shows the transition to the sustain phase at 9 t. The missile will be still accelerated but much slower. Burn out begins at about 17.5 t. After about 20 t the distance between the missile and the target is smaller or equal than 3 km. At this point,i.e. begin of the endgame, the target makes a step in order to show the ability of lateral acceleration. The new target position is now $[6000; 800; -9000]\,\mathrm{m}$ in the NED frame as we can see in figure 4.7. The fast tracking of the necessary high lateral acceleration can be seen in figure 4.6. Until this point the guidance is executed by transmitting via ground tracking a predicted intercept point based on radar measurements to the missile. During endgame, the seeker head adopts the guidance and provides the more precise measurements with a higher update frequency which leads to such acceleration steps in the end.

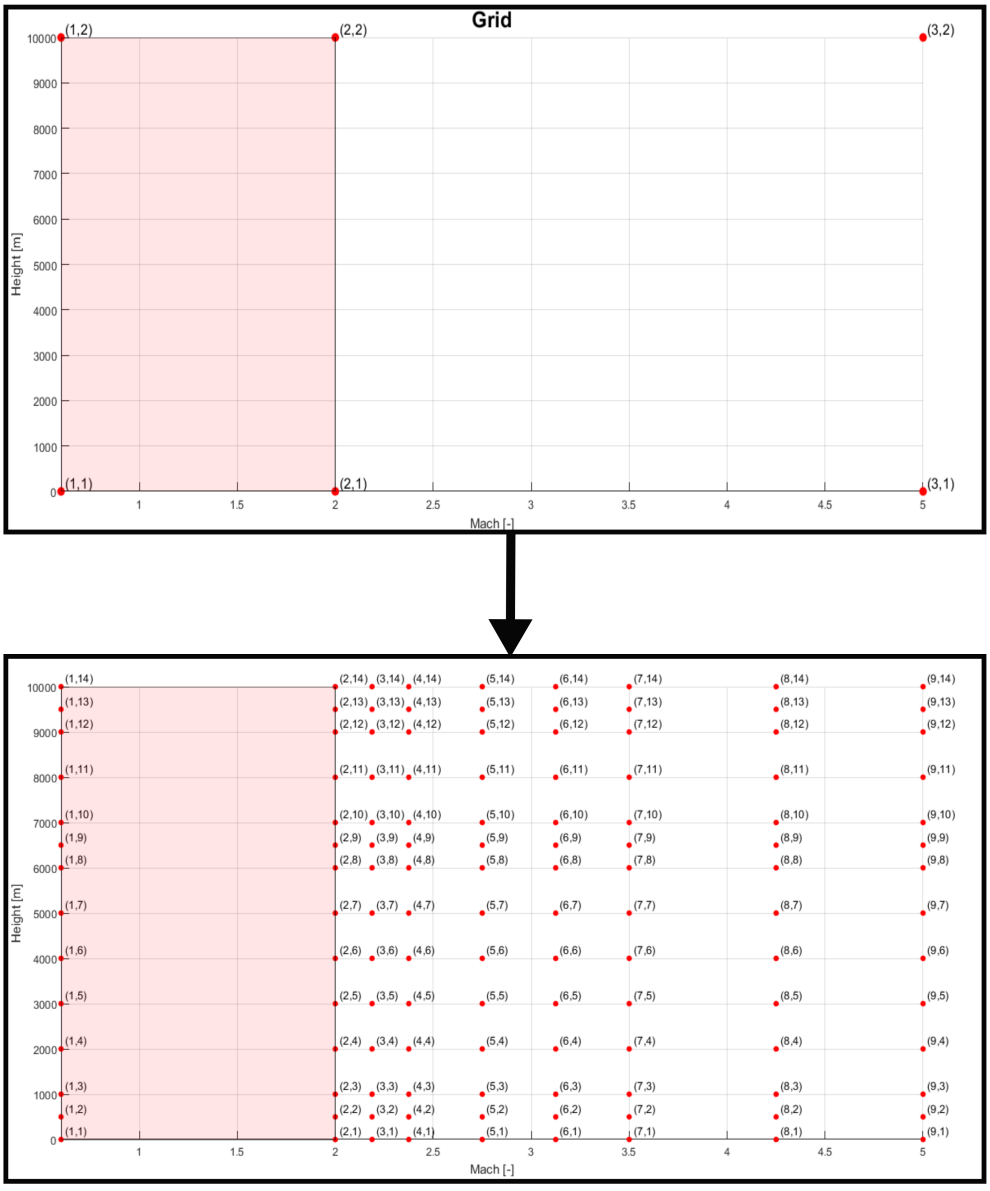

Figure 4.5: Grid at the beginning (top) and final distribution of the operating points after the automated controller design (bottom) for a missile with two aerodynamic actuating systems

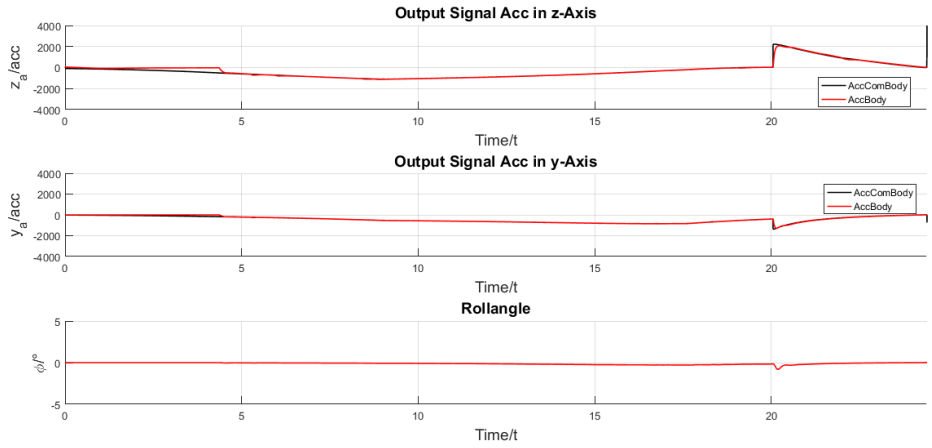

Figure 4.6: Commanded lateral accelerations (—) calculated by guidance laws shown in the top and middle plots and the actual behavior of the missile illustrated in — for all three axis

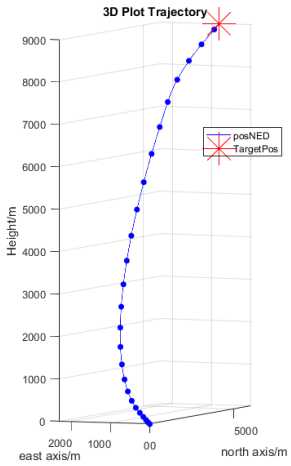

Figure 4.7: Trajectory of the missile in three-dimensional space

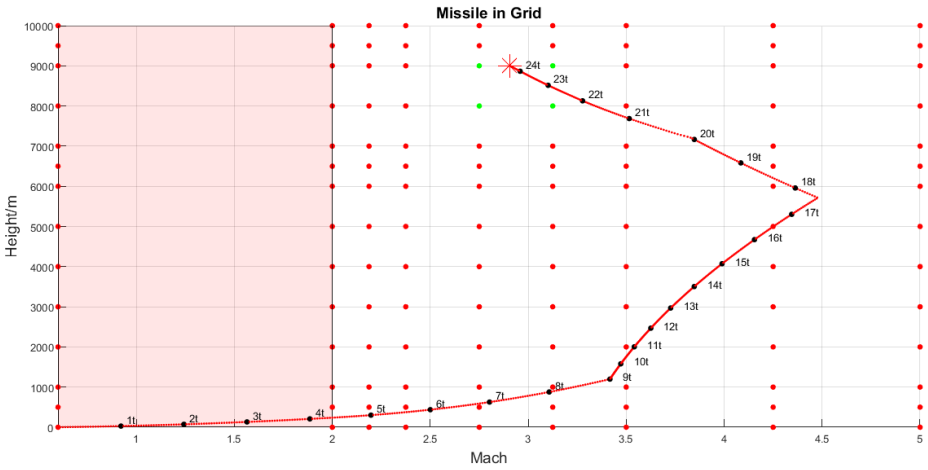

Figure 4.8: The path of the missile to the target through the operating point grid

4.2 Performance and Robustness

4.2.1 Two Aerodynamic Actuating Systems

For the performance and robustness check of the designed controllers, various settings had been applied to the nonlinear model. These are listed in the following table 4.3. The figures can be seen in appendix section A. The performance check was done for a missile traveling with Mach numbers between $2 \leq M \leq 4$. These checks are separated in simulations at low and high altitude for small and high acceleration steps.

Table 4.3: Simulation input data for an air defense maneuver with a missile which uses two aerodynamic actuating systems

Setting	**Characteristics**	**Figures** (Appendix)
	Low Altitude	
1. Small acceleration step of $z_a = 500\,\mathrm{acc}$ and $y_a = 500\,\mathrm{acc}$	Good performance, settling time below $200\,\mathrm{mt}$, no overshoot, only small AW activation	Output signals & states A.1 - Commands A.2
2. High acceleration step of $z_a = 5000\,\mathrm{acc}$ and $y_a = 2500\,\mathrm{acc}$	Good performance, settling time below $400\,\mathrm{mt}$, overshoot below $1\,\%$, large AW activation at the beginning and end of the steps but still stable	Output signals & states A.3 - Commands A.4 - Commands before PT1 of elevator A.5
	High Altitude	
3. Small acceleration step of $z_a = 500\,\mathrm{acc}$ and $y_a = 500\,\mathrm{acc}$	Good performance, settling time below $200\,\mathrm{mt}$, no overshoot, only small AW activation, small bump at the end of z_a step	Output signals & states A.6 - Commands A.7
4. High acceleration step of $z_a = 1500\,\mathrm{acc}$ and $y_a = 1000\,\mathrm{acc}$	Good performance, settling time below $200\,\mathrm{mt}$, no overshoot, medium AW activation at the beginning and end of the steps	Output signals & states A.8 - Commands A.9
5. High acceleration step of $z_a = 2750\,\mathrm{acc}$	Good performance, settling time below $400\,\mathrm{mt}$, small overshoot, large AW activation at the beginning of the step	Output & states of the elevator A.10 - Commands before PT-1 of elevator A.11
	Distorted Missile - Low Altitude	
6. High acceleration step of $z_a = 5000\,\mathrm{acc}$ and $y_a = 2500\,\mathrm{acc}$	Good performance, settling time below $800\,\mathrm{mt}$, Overshoot of about $20\,\%$, large AW activation at the beginning, end and during the steps	Output signals & states A.12 - Commands A.13 - Output comparison A.14

The results imply that the automatically designed controllers have a good performance at low and high altitudes for small and high acceleration steps as well as for controllers which have been designed with a distorted aerodynamic missile model. The small activation of the anti-windup networks in settings (1.) and (3.) arise from the calculation of $\boldsymbol{\delta}_i = \boldsymbol{u}_i - \mathrm{sat}(\boldsymbol{u}), i...p$ which has been stated in section 2.4.2.1. The manipulated variable is of course not bigger than $30\,^\circ$ (see figure A.2 and A.7). The difference $\boldsymbol{\delta} \neq 0$ arises from the interpolation of $\boldsymbol{u}$ during gain-scheduling. Therefore, the single $\boldsymbol{u}_i$ of the neighboring OPs differs slightly

from sat($\boldsymbol{u}$) because $\boldsymbol{u}$ is a function of all currently active controllers.
The small bump in setting (3.) is due to the imaginary part of some eigenvalues of the controllers. These are sometimes quite small and lead to no optimal performance. This can be resolved by increasing the imaginary to real ratio $\delta = \frac{\mathrm{Im}\{\lambda_i\}}{\mathrm{Re}\{\lambda_i\}}$. This yields probably at another OP no optimal performance. One has to set a good working balance.

Setting (2.) illustrates quite well the advantage of using anti-windup networks (figure A.3). The missile remains stable and reaches the desired acceleration step of 5000 acc even though the front elevator is in saturation for about 200 mt as you can see in figure A.5. These commanded signals are taken before the actual dynamics of the elevator are simulated by a PT-1 element (see for comparison figure A.4).

The good performance of the model-recovery anti-windup network by [13] and [15] can be also seen in setting (5.). Consider figures A.10 and A.11. Still at an altitude of about 9 km where the density of the atmosphere is only a third compared to sea level, and a Mach number of $M = 4$ the missile is able to perform acceleration steps of about 2750 acc. The front elevator remains nearly 400 mt saturated but keeps the missile stable.

For the robustness check the aerodynamic model, i.e. derivatives, had been distorted for the controller design. These distortion changes are listed in table 4.4 according to [16].

Table 4.4: Distortions for a missile using two aerodynamic actuating systems

Variables	Deviation
I_{xx}, I_{yy}, I_{zz}	+5 %
m	−1 %
x_{cg}	−50 mm
CZ_{10}, CZ_{20}, CZ_{21}	−10 %
CM_{10}, CM_{20}, CM_{21}, CM_{30}	−10 %
CX_{00}, CX_{40}, $CX_{et2,b}$, $CX_{et2,f}$, $CX_{e11,b}$, $CX_{e11,f}$	−10 %
CY_{21}	−10 %
CZ_{η_b}, CZ_{η_f}, CM_{η_b}, CM_{η_f}, CL_{ξ_b}, CL_{ξ_f}	−10 %
CN_{21}, CL_{21}	−20 %
CL_p, CM_q	−20 %

As setting (6.) shows by illustrations A.12, A.13 and A.14, the missile stays stable even though the aerodynamic model was distorted by the deviations listed in table 4.4. It is still able to perform steps of 2500 acc to 5000 acc. The overshoot is about 20 % percent which is satisfying according to such high acceleration steps and distortions.

For the controller design, the angle of attack can be considered for the linearization. The design was executed for an extraction of the whole operating range. The results are appended in section A.1.4. Figure A.15 shows the grid generation. You can see that we need about 160 controllers only for this extraction of the operating range ($0.6 \leq M \leq 4.5$, $2\,\text{km} \leq H \leq 8\,\text{km}$, $0\,^\circ \leq \alpha \leq 10\,^\circ$). Furthermore, figures A.16, A.17, A.18 illustrate the output signals, commanded signals and trajectory of the missile in 3D space respectively of an air defense scenario. You can see here the good performance of controllers too and their ability to handle high lateral acceleration steps when switching to the seeker head at about 14 t.
The performance for the execution of low lateral acceleration steps ($z_a = 500\,\text{acc}$, $y_a = 500\,\text{acc}$) at a low altitude of $H = 2\,\text{km}$ for an average velocity of $M = 3$ can be seen in figures A.19 and A.20. The performance is unchanged compared to the two-dimensional grid. A reason could be the design limits which are not optimal for the third dimension, i.e. angle of attack. Improving these limits and the settings for the LMI solver and decreasing the tolerances for the performance should lead to an enhanced performance compared to the controllers designed for a two-dimensional operating point grid.

4.2.2 One Aerodynamic Actuating Systems at the Back

As already mentioned, the automated controller design was extended in order to apply the design on other missiles. The whole design was successfully tested on a missile using one aerodynamic actuating system at the back. It was tested at low altitude (2 km) and a medium Mach number with an average value of $M = 3.5$ for low and high acceleration steps with respect to the missile's possible agility. The results are shown in appendix section A.2.1. Figures A.24 and A.23 illustrate the output signals and commanded signals for high acceleration steps of this missile ($z_a = 1750\,\text{acc}$, $y_a = 500\,\text{acc}$). We can see a good performance as well with a settling time of about 400 mt for the elevator and much less for the rudder. The overshoot of 10 % does not exceed the limit of ±15%. Furthermore, the non-minimal-phase behavior of tail-controlled missiles can be seen.
When comparing the missile using two aerodynamic actuating systems to the missile using one aerodynamic actuating system at the back, it is clear that the first one is much more agile with possible acceleration steps almost three times larger. The second one still needs control surface deflections bigger than 30 ° when performing acceleration steps of about 1750 acc. By optimizing the controller performance e.g. by improving the pole placement area, the agility can be enhanced but will not be as good as when using two aerodynamic actuating systems which is also a reason of the missile's mass and size. The mass of the one employing one aerodynamic actuating system is three times compared to the other one and its length is about twice as much. Therefore, the larger and heavier missile is of course much less agile.

5 Conclusion and Outlook

One concept for an automated controller design was worked out and together with the background theory for the control of missiles in the course of the present master thesis presented. Applying this design to a missile which uses aerodynamic actuating systems, a good working irregular operating point grid will be determined. Either two-dimensional of Mach number and height or three-dimensional of Mach number, height and angle of attack. The missile will be trimmed and linearized over a local area at the operating points. Afterwards, dynamic output feedback controllers for the execution of lateral acceleration steps as well as state feedback controllers to keep the roll angle zero are designed. The controllers stabilize the missile in a validity area which is big enough to stabilize the missile at adjacent grid points. Since the aerodynamics differ a lot between subsonic and supersonic, controllers to keep a defined roll rate in this "critical region" are automatically designed as well. For the stabilization of the missile in the presence of saturation of the actuators, a model-recovery anti-windup network was used and due to affine linear parameter dependency among the grid parameters automatically designed at the grid points. The whole design can then be simulated in Simulink which is linked via S-functions to the missile simulation program to get almost real conditions and feedback. The method of gain-scheduling allows the interpolation of the manipulated variable computed by neighboring operating points to keep the missile stable on its mission.

As you have seen in section 4, the controllers have a good working performance but are not optimized. To reach an optimal controller performance was not part of this master thesis but can be achieved by improving the determination of the design limits for the pole placement area. Additionally, the weighting factors on the control error and actuator dynamics for the output energy optimization can be enhanced. To improve computing time you should consider recursive grid generation and using sparse grids (see figure 5.1 [6]).
This method is much more complex but much less time consuming and allows to design output feedback controllers in regions below $M = 2$. Furthermore, the grid points and hence computing time when linearizing the missile over three dimensions can be drastically reduced. Furthermore, the possibility of applying the automated controller design to missiles using a divert and attitude control system should be considered.
The developed automated controller design is able to evaluate different missiles using aerodynamic actuating systems in the sense of flight control and can be extended easily by additional missile concepts.

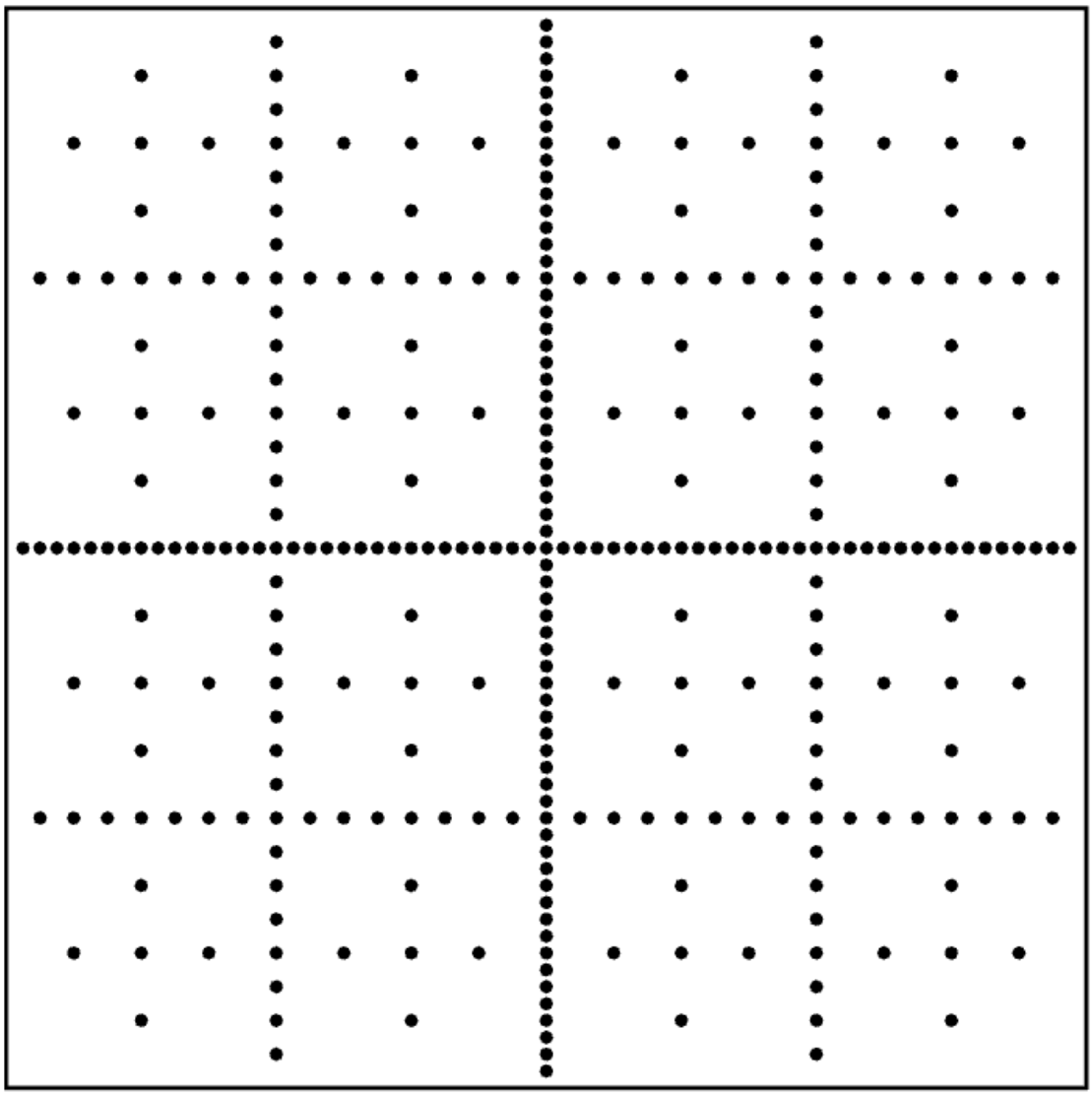

Figure 5.1: Example of a sparse grid [6]

Bibliography

[1] Jürgen Adamy. *Nichtlineare Regelungen*. Springer, 2009.

[2] Tore Bakka. *Multiobjective Optimization and Multivariable Control of Offshore Wind Turbine System*. University of Agder, 2013.

[3] Benedikt Bartenschlager. "Regelkonzept für einen Flugkörper mit zwei Stellsystemen". unpublished. Diploma Thesis. Technische Universität München, 2014.

[4] Frank L. Lewis Brian L. Stevens. *Aircraft Control and Simulation*. John Wiley and Sons, Inc., New York, 1992.

[5] Mahmoud Chilali Carsten Scherer Pascal Gahinet. *Multiobjective Output-Feedback Control via LMI Optimization*. IEEE Transactions on Automatic Control. 1997.

[6] Michael Gabriel Hans-Joachim Bungartz. *Sparse Grids*. Acta Numerica. 2004.

[7] Florian Holzapfel. *Flugsystemdynamik 2*. Technische Universität München.

[8] Richard D. Braatz Jeremy G. VanAntwerp. *A tutorial on linear and bilinear matrix inequalities*. Journal of Process Control. 2000.

[9] Klaus Kefferpütz. "Regelungen für Systeme unter Stellgrößen- und Stellratenbeschränkungen". PhD thesis. 2012.

[10] Hendrik Lens. "Schnelle Regelung mit Ausgangsrückführung für Systeme mit Stellgrößenbeschränkungen". PhD thesis. 2009.

[11] Johan Löfberg. *YALMIP A toolbox for modeling and optimization in MATLAB*. 2004.

[12] Jan Lunze. *Regelungstechnik 1*. Springer, 2007.

[13] Guido Herrmann Matthew C. Turner and Ian Postlethwaite. *Anti-windup Compensation using a Decoupling Architecture*. University of Leicester. 2007.

[14] Arkadi Nemirovski. *INTERIOR POINT POLYNOMIAL TIME METHODS IN CONVEX PROGRAMMING*. Lecture Notes. GEORGIA INSTITUTE OF TECHNOLOGY.

[15] Andreas Ortseifen. "Entwurf von modellbasierten Anti-Windup-Methoden für Systeme mit Stellbegrenzungen". PhD thesis. 2012.

[16] Florian Peter et al. *L1 Adaptive Augmentation of a Missile Autopilot*. AIAA Guidance, Navigation and Control Conference. 2012.

[17] M. J. Todd R. H. Tütüncü K. C. Toh. *SDPT3 - a Matlab software package for semidefinite-quadratic-linear programming, version 3.0*. IEEE International Symposium on Computer Aided Control Systems Design. 2001.

[18] Wolfgang Alles Rudolf Brockhaus and Robert Luckner. *Flugregelung*. Springer, 2011.

[19] George M. Siouris. *Missile Guidance and Control Systems*. Springer, 2004.

[20] Eric Feron Stephen Boyd Laurent El Ghaoui and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

[21] Lieven Vandenberghe Stephen Boyd. *Convex Optimization*. Lecture Slides. Cambridge University Press.

[22] Lieven Vandenberghe Stephen Boyd. *Convex Optimization*. Cambridge University Press, 2004.

[23] *Tuning Analysis Tools: Step and Frequency Response*. `http://www1.adept.com/main/KE/DATA/Controller/SmartMotion_Developer/TH_Tuning.html`. Accessed: 19.09.2016.

[24] Tobias Martin Ulmer. "Entwurf eines robusten Kennfeldreglers für die Längsbewegung eines generischen Jet-Trainers mittels H - Loop-Shaping". PhD thesis. 2007.

[25] Leif Walter et al. *Real-Time Optimal Gain Scheduling for Nonlinear Dynamic Inversion*. AIAA Guidance, Navigation, and Control Conference. 2014.

[26] Roan Westerhof. *A survey of literature on controller scheduling*. Technische Universität Eindhoven. 2000.

[27] Peter H. Zipfel. *Modeling and Simulation of Aerospace Vehicle Dynamics*. American Institute of Aeronautics and Astronautics, 2007.

[28] Javed Qureshi Zoran Gajic Muhammad Tahir. *Lyapunov Matrix Equation in System Stability and Control*. Springer, 1995.

Appendices

A Performance and Robustness Checks

A.1 Two Aerodynamic Actuating Systems

A.1.1 Low Altitude

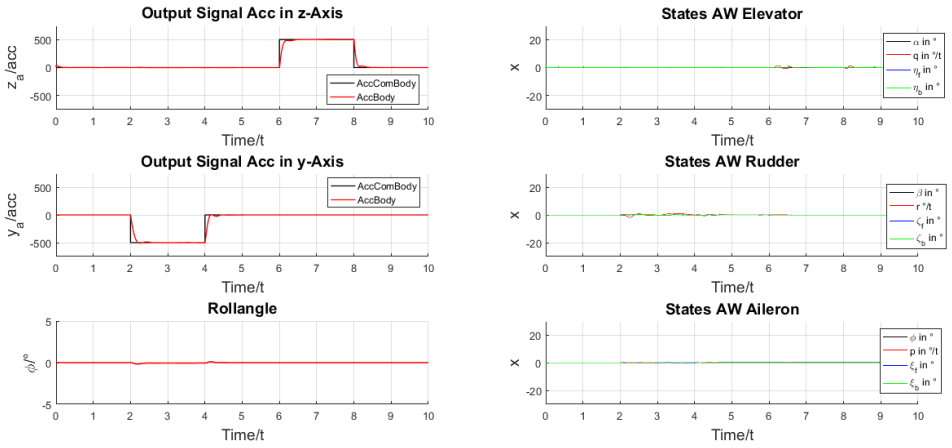

Figure A.1: Output signals and anti-windup states for a missile executing small lateral acceleration steps ($z_a = 500\,\text{acc}$, $y_a = 500\,\text{acc}$) at low altitudes ($H = 2\,\text{km}$) for average velocities ($M = 3$)

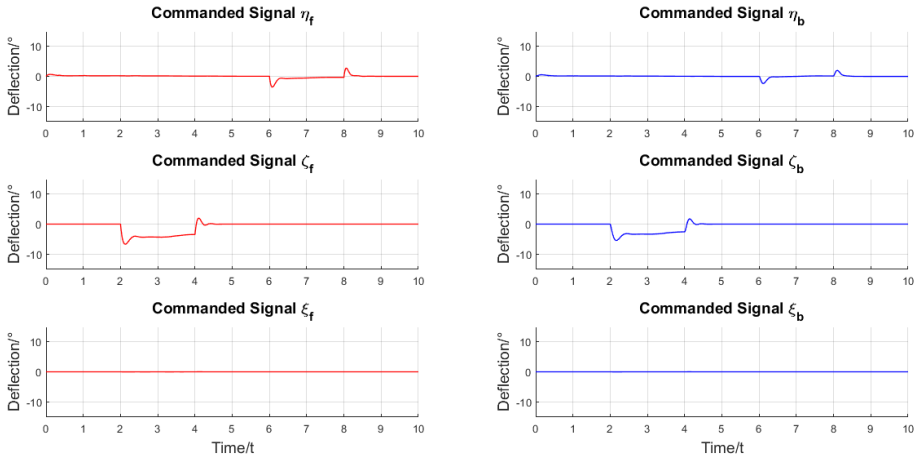

Figure A.2: Commanded signals and anti-windup states for a missile executing small lateral acceleration steps ($z_a = 500\,\text{acc}$, $y_a = 500\,\text{acc}$) at low altitudes ($H = 2\,\text{km}$) for average velocities ($M = 3$)

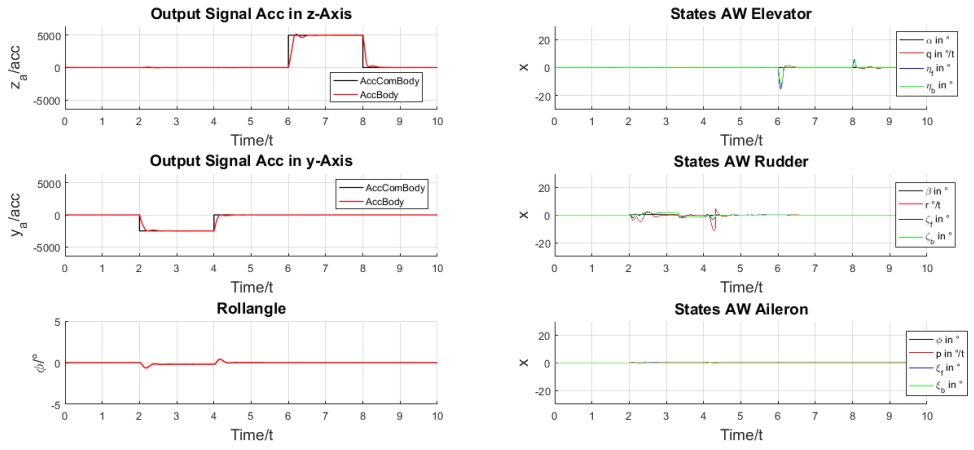

Figure A.3: Output signals and anti-windup states for a missile executing high lateral acceleration steps ($z_a = 5000\,\text{acc}$, $y_a = 2500\,\text{acc}$) at low altitudes ($H = 2\,\text{km}$) for average velocities ($M = 3$)

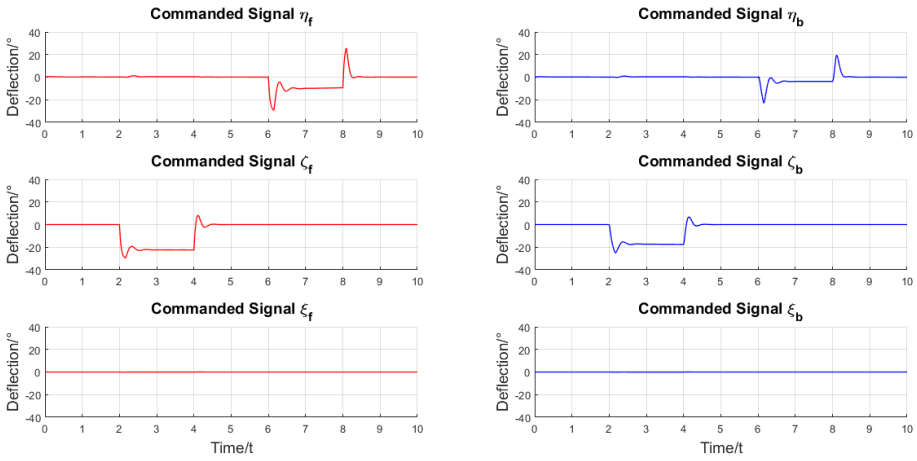

Figure A.4: Commanded signals and anti-windup states for a missile executing high lateral acceleration steps ($z_a = 5000\,\text{acc}$, $y_a = 2500\,\text{acc}$) at low altitudes ($H = 2\,\text{km}$) for average velocities ($M = 3$)

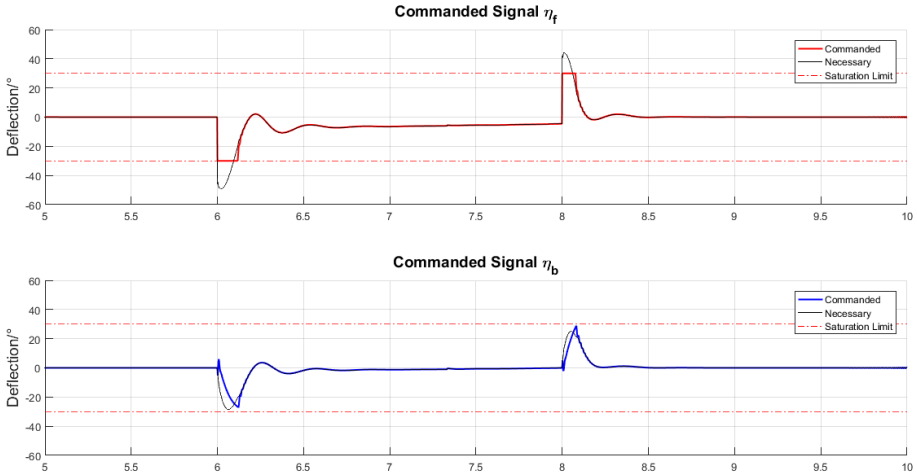

Figure A.5: Commanded signals and anti-windup states of the elevator of a missile executing a high lateral acceleration step ($z_a = 5000\,\text{acc}$) at low altitude ($H = 2\,\text{km}$) with a velocity of about $M = 4$. The signal is taken before the delayed dynamics of the actuator

A.1.2 High Altitude

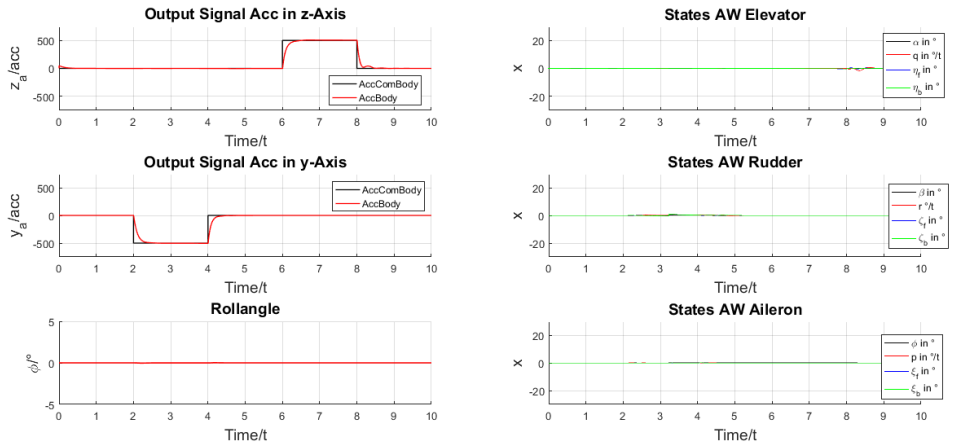

Figure A.6: Output signals and anti-windup states for a missile executing small lateral acceleration steps ($z_a = 500\,\text{acc}$, $y_a = 500\,\text{acc}$) at high altitudes ($H = 9\,\text{km}$) for average velocities ($M = 3$)

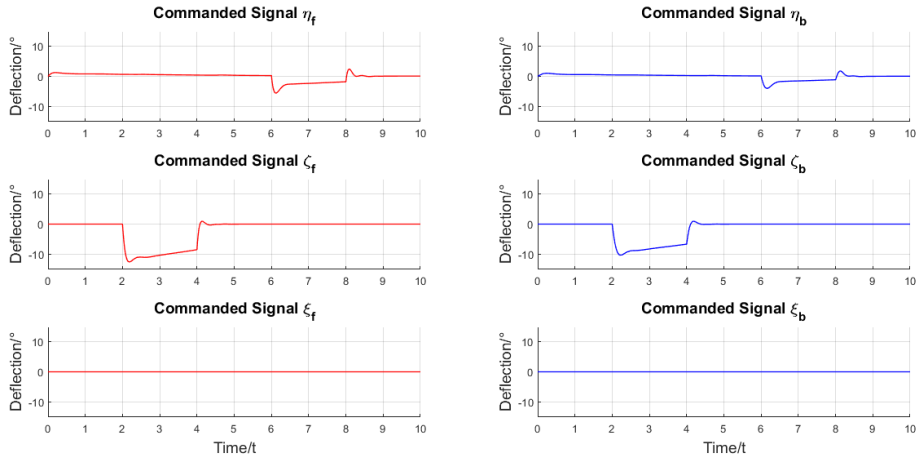

Figure A.7: Commanded signals for a missile executing small lateral acceleration steps ($z_a = 500\,\text{acc}$, $y_a = 500\,\text{acc}$) at high altitudes ($H = 9\,\text{km}$) for average velocities ($M = 3$)

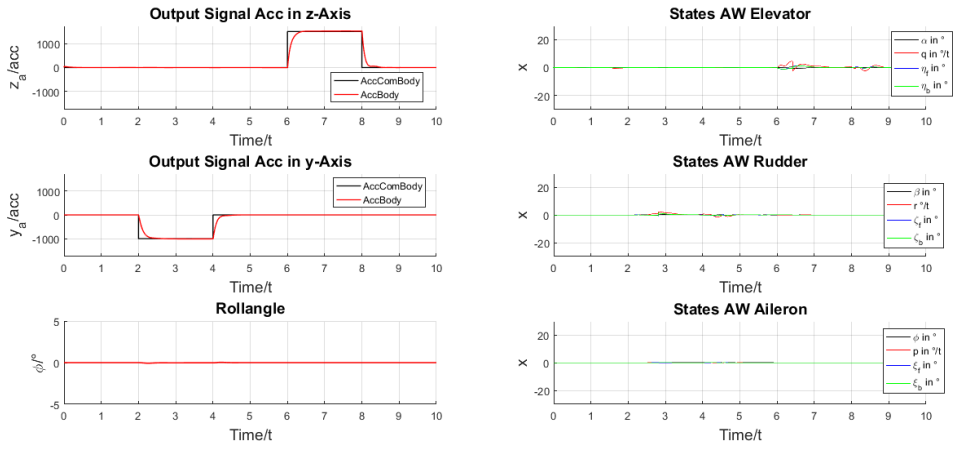

Figure A.8: Output signals and anti-windup states for a missile executing high lateral acceleration steps ($z_a = 1500\,\text{acc}$, $y_a = 1000\,\text{acc}$) at high altitudes ($H = 9\,\text{km}$) for average velocities ($M = 3$)

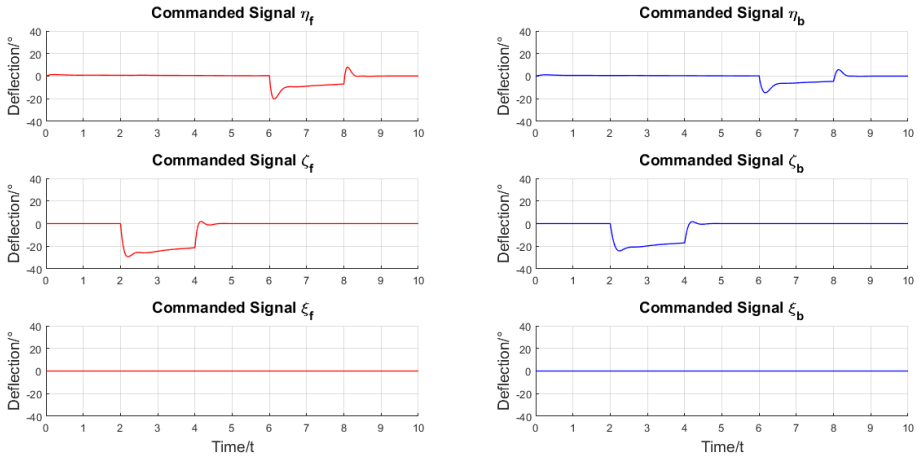

Figure A.9: Commanded signals for a missile executing high lateral acceleration steps ($z_a = 1500\,\text{acc}$, $y_a = 1000\,\text{acc}$) at high altitudes ($H = 9\,\text{km}$) for average velocities ($M = 3$)

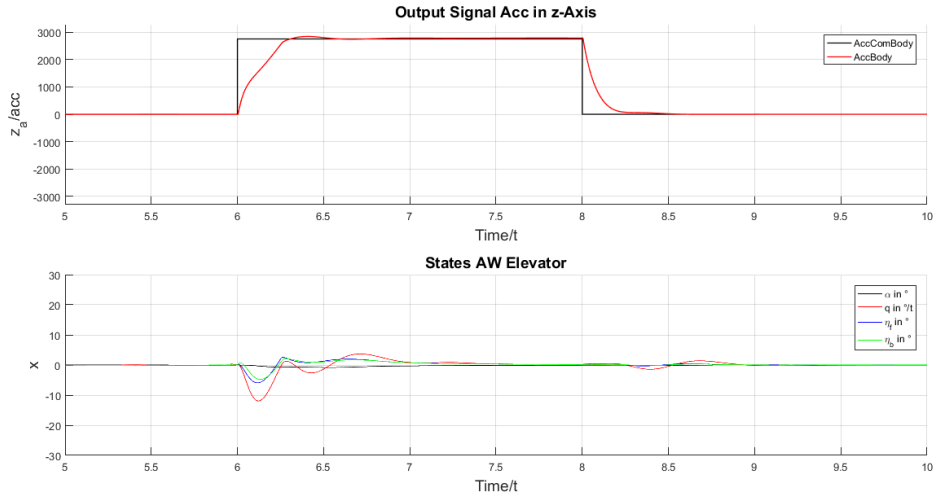

Figure A.10: Output signal and anti-windup states for a missile executing high lateral acceleration step of $z_a = 2750\,\text{acc}$ at high altitude of $H = 9\,\text{km}$ at a velocity of about $M = 4$

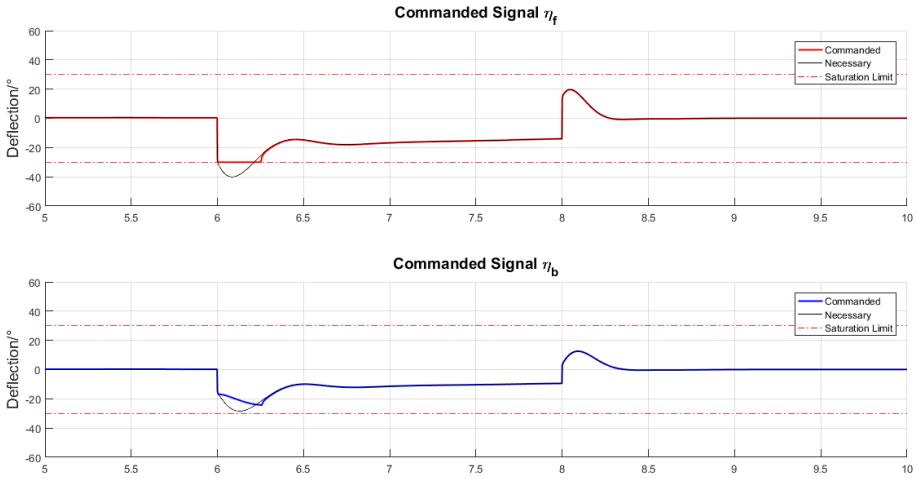

Figure A.11: Commanded signals for a missile executing high lateral acceleration step of $z_a = 2750\,\text{acc}$ at high altitude of $H = 9\,\text{km}$ at a velocity of about $M = 4$

A.1.3 Robustness

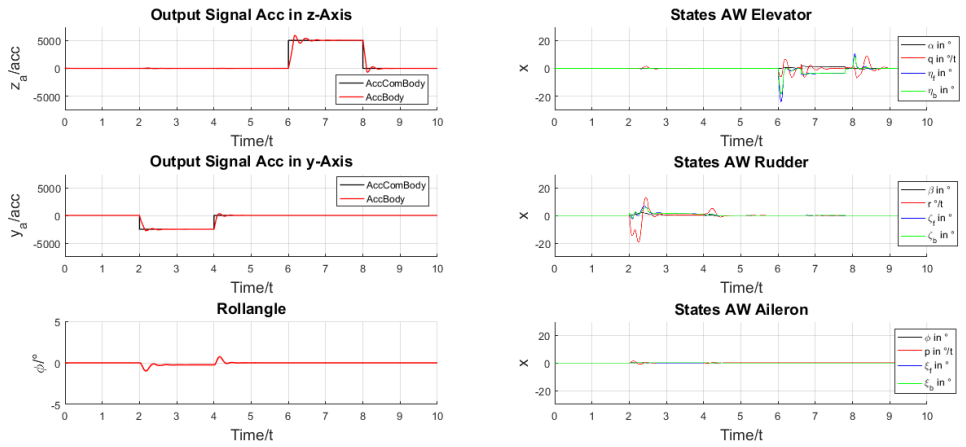

Figure A.12: Output signal and anti-windup states for a missile which was designed with a distorted aerodynamic model executing high lateral acceleration steps ($z_a = 5000\,\text{acc}$, $y_a = 2500\,\text{acc}$) at high altitude ($H = 2.5\,\text{km}$) for an average velocity ($M = 3.5$)

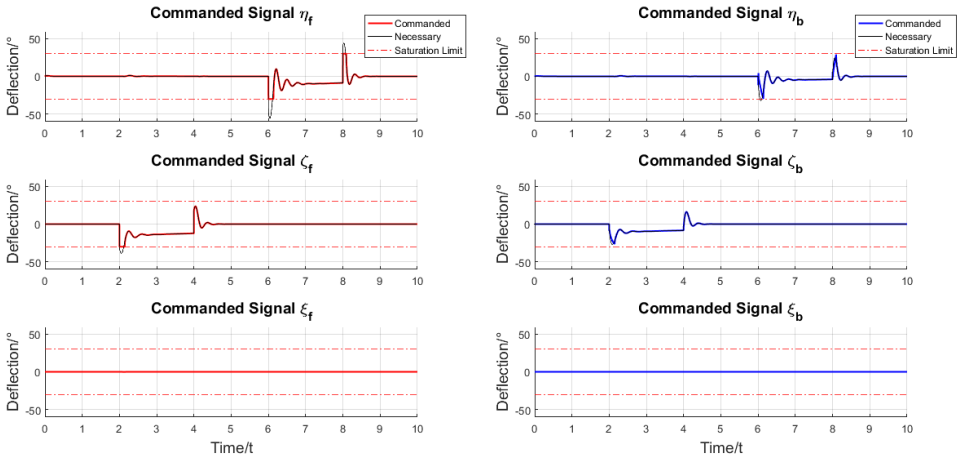

Figure A.13: Commanded signals for a missile which was designed with a distorted aerodynamic model executing high lateral acceleration steps ($z_a = 5000\,\text{acc}$, $y_a = 2500\,\text{acc}$) at high altitude ($H = 2.5\,\text{km}$) for an average velocity ($M = 3.5$)

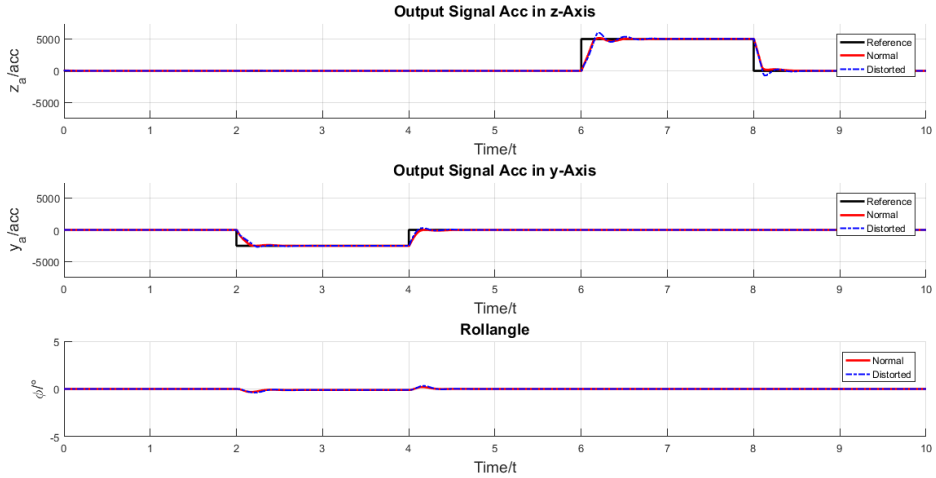

Figure A.14: Output signals for a missile which was normally designed (—) and which was designed with a distorted aerodynamic model (—) executing high lateral acceleration steps ($z_a = 5000\,\text{acc}$, $y_a = 2500\,\text{acc}$) at high altitude ($H = 2.5\,\text{km}$) for an average velocity ($M = 3.5$)

A.1.4 Three-Dimensional Grid

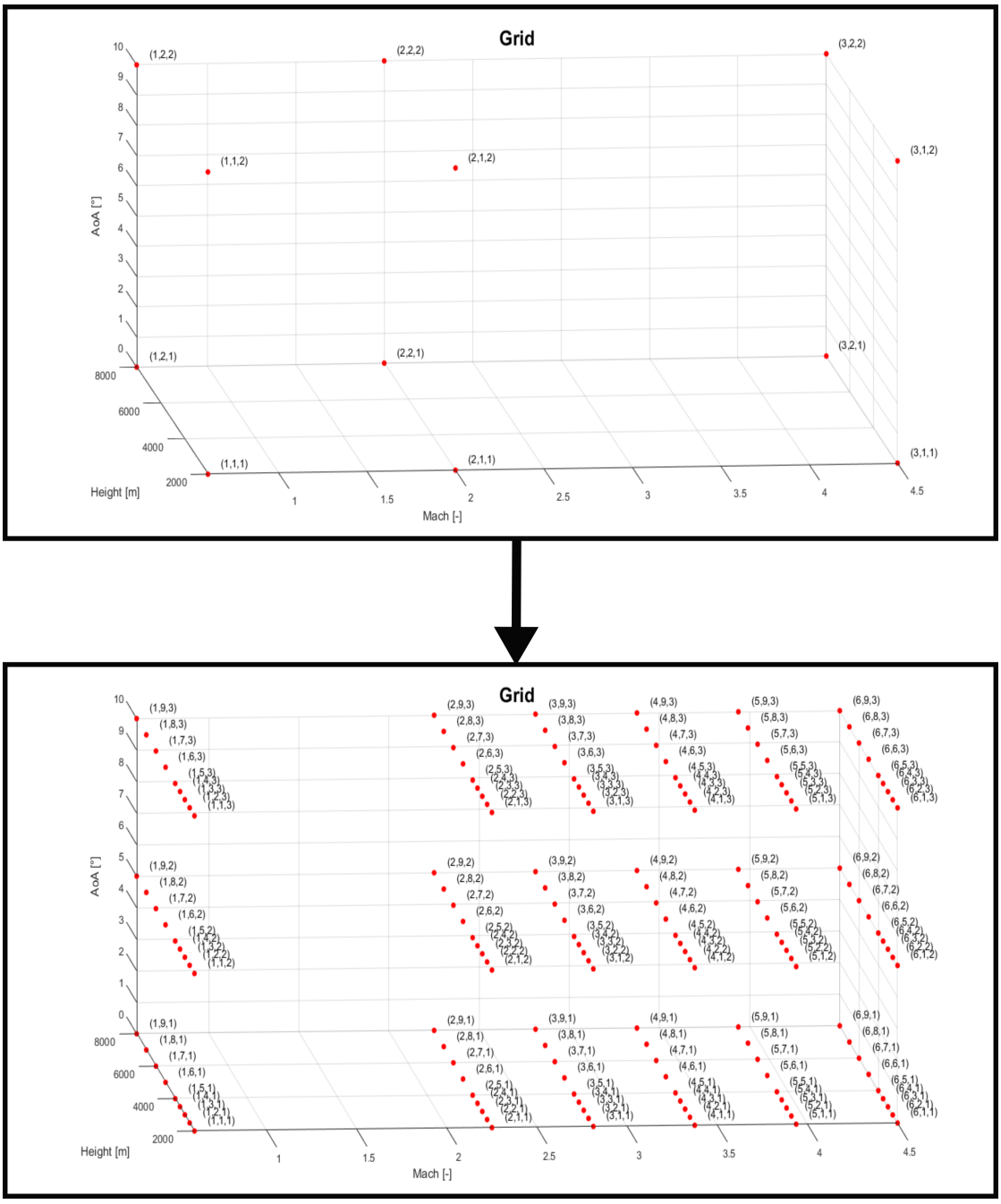

Figure A.15: Three-dimensional grid at the beginning of the design (top) and the final grid (bottom) for an operating range of $M = 0.6 - 4.5$, $H = 2 - 8\,\mathrm{km}$ and $\alpha = 0 - 10\,^\circ$

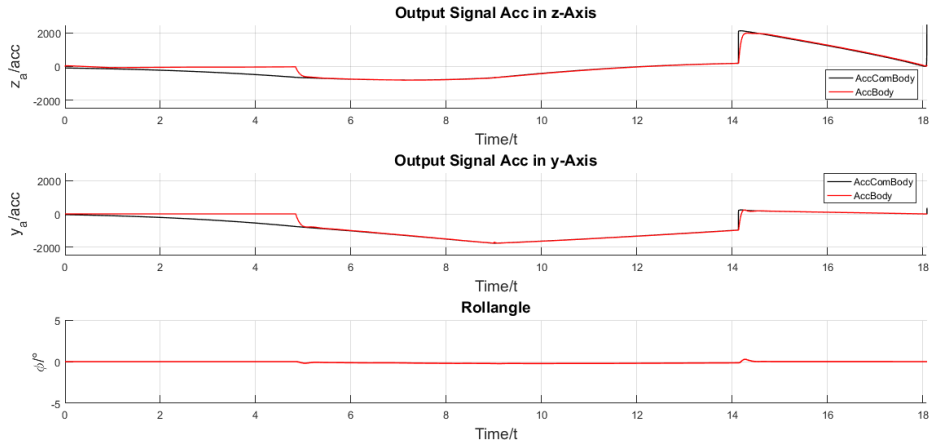

Figure A.16: Output signals for a missile using guidance laws to eliminate target

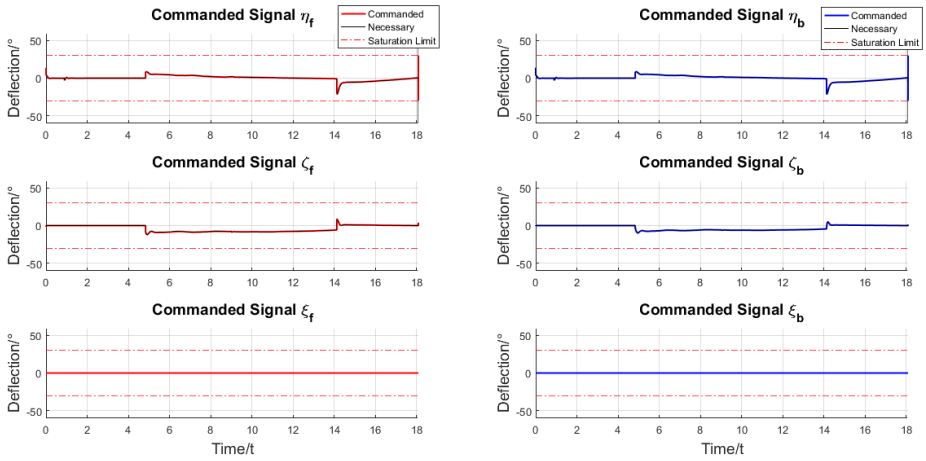

Figure A.17: Commanded signals for a missile using guidance laws to eliminate target

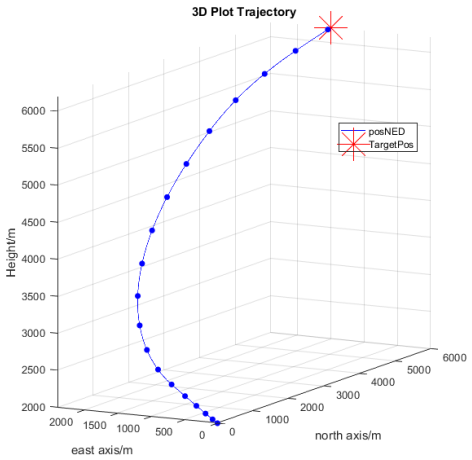

Figure A.18: Trajectory in 3D space for a missile using guidance laws to eliminate the target

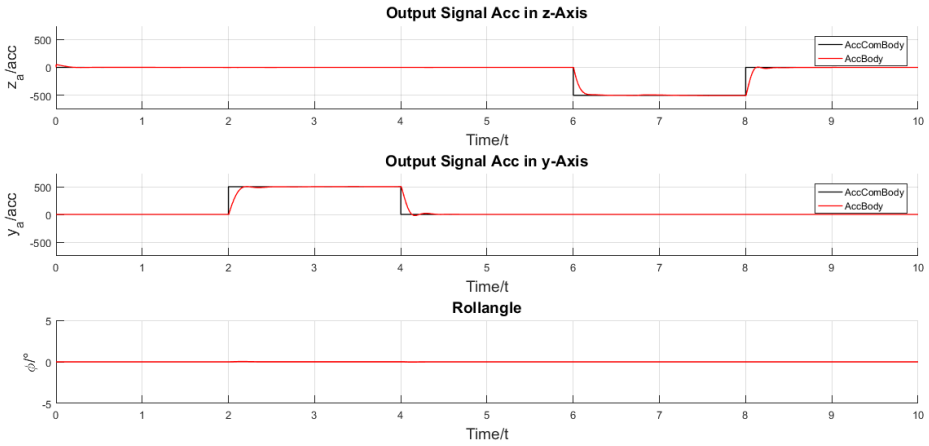

Figure A.19: Output signals for a missile executing low lateral acceleration steps ($z_a = 500\,\text{acc}$, $y_a = 500\,\text{acc}$) at low altitudes ($H = 2\,\text{km}$) for average velocities ($M = 3$)

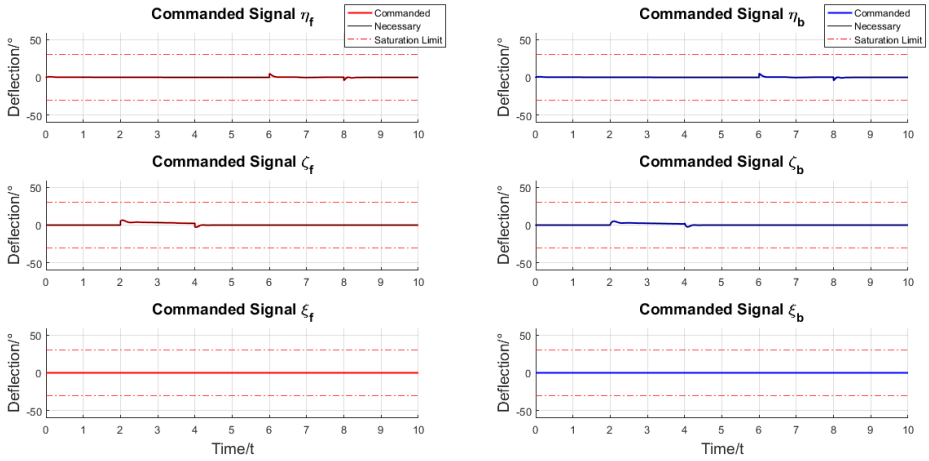

Figure A.20: Commanded signals for a missile executing low lateral acceleration steps ($z_a = 500\,\text{acc}$, $y_a = 500\,\text{acc}$) at low altitudes ($H = 2\,\text{km}$) for average velocities ($M = 3$)

A.2 One Aerodynamic Actuating System at the Back

A.2.1 Low Altitude and Medium Mach Number

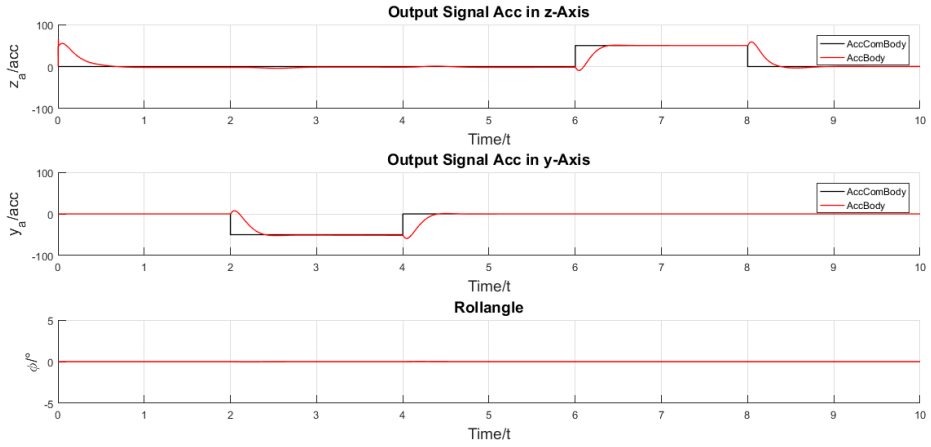

Figure A.21: Output signal for a missile executing small lateral accelerating steps ($z_a = 50\,\text{acc}$, $y_a = 50\,\text{acc}$) at high altitude ($H = 2\,\text{km}$) for an average velocity ($M = 3.5$)

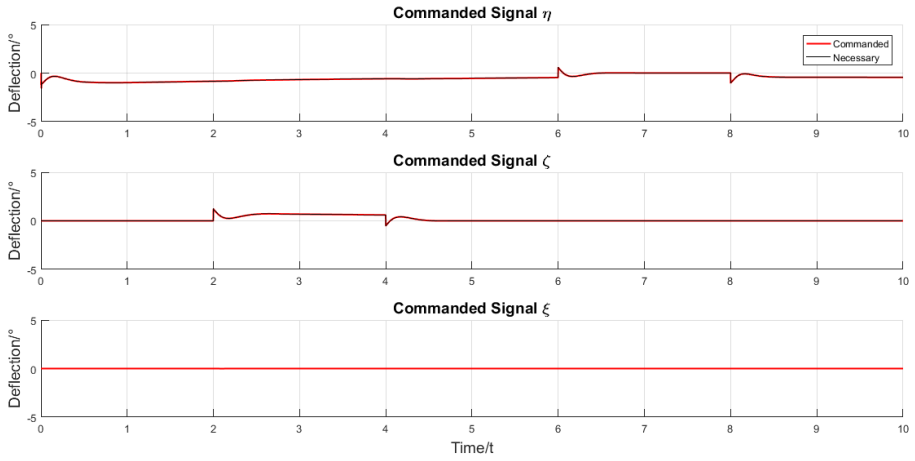

Figure A.22: Commanded signals for a missile executing small lateral accelerating steps ($z_a = 50\,\text{acc}$, $y_a = 50\,\text{acc}$) at high altitude ($H = 2\,\text{km}$) for an average velocity ($M = 3.5$)

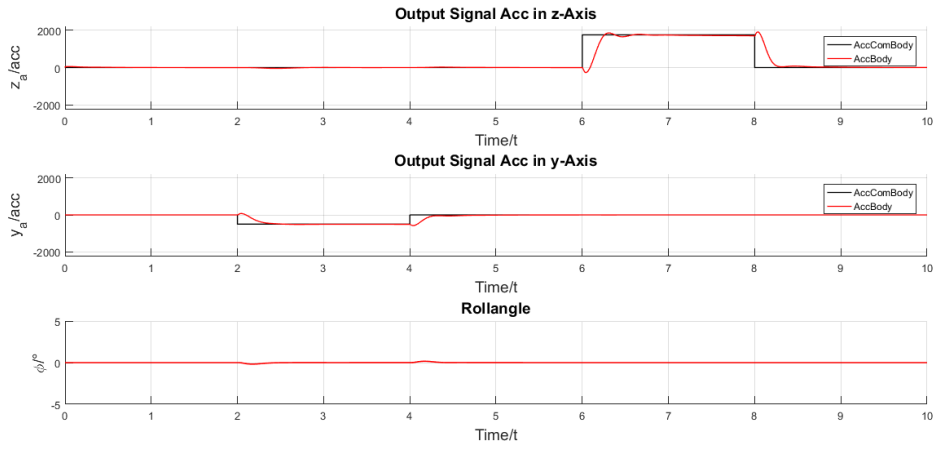

Figure A.23: Output signal for a missile executing high lateral acceleration steps ($z_a = 1750\,\text{acc}$, $y_a = 500\,\text{acc}$) at high altitude ($H = 2\,\text{km}$) for an average velocity ($M = 3.5$)

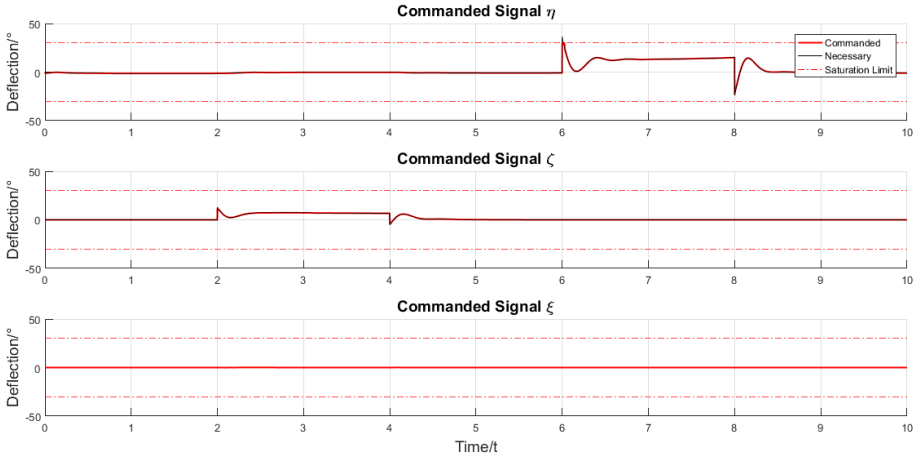

Figure A.24: Commanded signals for a missile executing high lateral acceleration steps ($z_a = 1750\,\text{acc}$, $y_a = 500\,\text{acc}$) at high altitude ($H = 2\,\text{km}$) for an average velocity ($M = 3.5$)